
Open edX Enterprise Service Documentation

Release 4.17.0

Axim Collaborative, Inc

Apr 30, 2024

CONTENTS

1	enterprise	3
1.1	Documentation	3
1.2	License	3
1.3	How To Contribute	3
1.4	Reporting Security Issues	4
1.5	Getting Help	4
2	Getting Started	5
2.1	Production setup	5
2.2	Local development	5
3	Configuration and usage	7
3.1	Getting around the administrator interface	7
3.2	Manage Learners View	8
3.3	Enrollment notification email templates	10
3.4	Integrated Channels	11
4	Development documentation	15
4.1	High-level Architecture Overview	15
4.2	Local setup	15
5	Testing	19
5.1	Running all of the python tests	19
5.2	Code coverage	19
5.3	Running subsets of tests	20
5.4	Quality	20
6	Internationalization	21
6.1	Updating Translations	21
6.2	Fake Translations	21
7	Segment events	23
7.1	edx.bi.user.enterprise.onboarding	23
7.2	edx.bi.user.enterprise.enrollment.course	23
7.3	edx.bi.user.consent_form.shown	24
7.4	edx.bi.user.consent_form.accepted	24
7.5	edx.bi.user.consent_form.denied	24
8	enterprise	25
8.1	enterprise package	25

9	Change Log	275
9.1	Unreleased	275
9.2	[4.17.0]	275
9.3	[4.16.5]	275
9.4	[4.16.4]	275
9.5	[4.16.3]	275
9.6	[4.16.2]	275
9.7	[4.16.1]	276
9.8	[4.16.0]	276
9.9	[4.15.12]	276
9.10	[4.15.11]	276
9.11	[4.15.10]	276
9.12	[4.15.9]	276
9.13	[4.15.8]	276
9.14	[4.15.7]	276
9.15	[4.15.6]	277
9.16	[4.15.5]	277
9.17	[4.15.4]	277
9.18	[4.15.3]	277
9.19	[4.15.2]	277
9.20	[4.15.1]	277
9.21	[4.15.0]	277
9.22	[4.14.0]	277
9.23	[4.13.14]	278
9.24	[4.13.13]	278
9.25	[4.13.12]	278
9.26	[4.13.11]	278
9.27	[4.13.10]	278
9.28	[4.13.9]	278
9.29	[4.13.8]	278
9.30	[4.13.7]	278
9.31	[4.13.6]	279
9.32	[4.13.5]	279
9.33	[4.13.4]	279
9.34	[4.13.3]	279
9.35	[4.13.2]	279
9.36	[4.13.1]	279
9.37	[4.13.0]	279
9.38	[4.12.6]	279
9.39	[4.12.5]	280
9.40	[4.12.4]	280
9.41	[4.12.3]	280
9.42	[4.12.2]	280
9.43	[4.12.1]	280
9.44	[4.12.0]	280
9.45	[4.11.15]	280
9.46	[4.11.14]	280
9.47	[4.11.13]	281
9.48	[4.11.12]	281
9.49	[4.11.11]	281
9.50	[4.11.10]	281
9.51	[4.11.9]	281
9.52	[4.11.8]	281
9.53	[4.11.7]	281

9.54	[4.11.6]	281
9.55	[4.11.5]	282
9.56	[4.11.4]	282
9.57	[4.11.3]	282
9.58	[4.11.2]	282
9.59	[4.11.1]	282
9.60	[4.11.0]	282
9.61	[4.10.11]	282
9.62	[4.10.10]	282
9.63	[4.10.9]	283
9.64	[4.10.8]	283
9.65	[4.10.7]	283
9.66	[4.10.6]	283
9.67	[4.10.5]	283
9.68	[4.10.4]	283
9.69	[4.10.3]	283
9.70	[4.10.2]	283
9.71	[4.10.1]	284
9.72	[4.10.0]	284
9.73	[4.9.5]	284
9.74	[4.9.4]	284
9.75	[4.9.3]	284
9.76	[4.9.2]	284
9.77	[4.9.1]	284
9.78	[4.9.0]	285
9.79	[4.8.18]	285
9.80	[4.8.17]	285
9.81	[4.8.16]	285
9.82	[4.8.15]	285
9.83	[4.8.14]	285
9.84	[4.8.13]	285
9.85	[4.8.12]	285
9.86	[4.8.11]	286
9.87	[4.8.10]	286
9.88	[4.8.9]	286
9.89	[4.8.8]	286
9.90	[4.8.7]	286
9.91	[4.8.6]	286
9.92	[4.8.5]	286
9.93	[4.8.4]	286
9.94	[4.8.3]	287
9.95	[4.8.1]	287
9.96	[4.7.6]	287
9.97	[4.7.5]	287
9.98	[4.7.4]	287
9.99	[4.7.3]	287
9.100	[4.7.2]	287
9.101	[4.7.1]	287
9.102	[4.7.0]	288
9.103	[4.6.12]	288
9.104	[4.6.11]	288
9.105	[4.6.10]	288
9.106	[4.6.9]	288
9.107	[4.6.8]	288

9.108 [4.6.7]	288
9.109 [4.6.6]	288
9.110 [4.6.5]	289
9.111 [4.6.4]	289
9.112 [4.6.3]	289
9.113 [4.6.2]	289
9.114 [4.6.1]	289
9.115 [4.6.0]	289
9.116 [4.5.7]	289
9.117 [4.5.6]	289
9.118 [4.5.5]	290
9.119 [4.5.4]	290
9.120 [4.5.3]	290
9.121 [4.5.2]	290
9.122 [4.5.1]	290
9.123 [4.5.0]	290
9.124 [4.4.0]	290
9.125 [4.3.4]	290
9.126 [4.3.3]	291
9.127 [4.3.2]	291
9.128 [4.3.1]	291
9.129 [4.3.0]	291
9.130 [4.2.0]	291
9.131 [4.1.15]	291
9.132 [4.1.14]	291
9.133 [4.1.13]	291
9.134 [4.1.12]	292
9.135 [4.1.11]	292
9.136 [4.1.10]	292
9.137 [4.1.9]	292
9.138 [4.1.8]	292
9.139 [4.1.7]	292
9.140 [4.1.6]	292
9.141 [4.1.5]	292
9.142 [4.1.4]	293
9.143 [4.1.3]	293
9.144 [4.1.2]	293
9.145 [4.1.1]	293
9.146 [4.1.0]	293
9.147 [4.0.18]	293
9.148 [4.0.17]	293
9.149 [4.0.16]	293
9.150 [4.0.15]	294
9.151 [4.0.14]	294
9.152 [4.0.13]	294
9.153 [4.0.12]	294
9.154 [4.0.11]	294
9.155 [4.0.10]	294
9.156 [4.0.9]	294
9.157 [4.0.8]	294
9.158 [4.0.7]	295
9.159 [4.0.6]	295
9.160 [4.0.5]	295
9.161 [4.0.4]	295

9.162 [4.0.3]	295
9.163 [4.0.2]	295
9.164 [4.0.1]	295
9.165 [4.0.0]	295
9.166 [3.69.1]	296
9.167 [3.69.0]	296
9.168 [3.68.1]	296
9.169 [3.68.0]	296
9.170 [3.67.7]	296
9.171 [3.67.6]	296
9.172 [3.67.5]	296
9.173 [3.67.4]	296
9.174 [3.67.3]	297
9.175 [3.67.2]	297
9.176 [3.67.1]	297
9.177 [4.0.0]	297
9.178 [3.67.0]	297
9.179 [3.66.8]	297
9.180 [3.66.7]	297
9.181 [3.66.6]	297
9.182 [3.66.5]	298
9.183 [3.66.4]	298
9.184 [3.66.3]	298
9.185 [3.66.2]	298
9.186 [3.66.1]	298
9.187 [3.66.0]	298
9.188 [3.65.4]	298
9.189 [3.65.2]	298
9.190 [3.65.1]	299
9.191 [3.65.0]	299
9.192 [3.64.1]	299
9.193 [3.64.0]	299
9.194 [3.63.0]	299
9.195 [3.62.7]	299
9.196 [3.62.6]	299
9.197 [3.62.5]	299
9.198 [3.62.4]	300
9.199 [3.62.3]	300
9.200 [3.62.2]	300
9.201 [3.62.1]	300
9.202 [3.62.0]	300
9.203 [3.61.16]	300
9.204 [3.61.15]	300
9.205 [3.61.14]	300
9.206 [3.61.13]	301
9.207 [3.61.12]	301
9.208 [3.61.11]	301
9.209 [3.61.10]	301
9.210 [3.61.9]	301
9.211 [3.61.8]	301
9.212 [3.61.7]	301
9.213 [3.61.6]	301
9.214 [3.61.5]	302
9.215 [3.61.4]	302

9.216 [3.61.3]	302
9.217 [3.61.2]	302
9.218 [3.61.1]	302
9.219 [3.61.0]	302
9.220 [3.60.21]	302
9.221 [3.60.20]	302
9.222 [3.60.19]	303
9.223 [3.60.18]	303
9.224 [3.60.17]	303
9.225 [3.60.16]	303
9.226 [3.60.15]	303
9.227 [3.60.14]	303
9.228 [3.60.13]	303
9.229 [3.60.12]	303
9.230 [3.60.11]	304
9.231 [3.60.10]	304
9.232 [3.60.9]	304
9.233 [3.60.8]	304
9.234 [3.60.7]	304
9.235 [3.60.6]	304
9.236 [3.60.5]	304
9.237 [3.60.4]	304
9.238 [3.60.3]	305
9.239 [3.60.2]	305
9.240 [3.60.1]	305
9.241 [3.60.0]	305
9.242 [3.59.2]	305
9.243 [3.59.1]	305
9.244 [3.59.0]	305
9.245 [3.58.18]	305
9.246 [3.58.17]	306
9.247 [3.58.16]	306
9.248 [3.58.15]	306
9.249 [3.58.14]	306
9.250 [3.58.13]	306
9.251 [3.58.12]	306
9.252 [3.58.11]	306
9.253 [3.58.10]	306
9.254 [3.58.9]	307
9.255 [3.58.8]	307
9.256 [3.58.7]	307
9.257 [3.58.6]	307
9.258 [3.58.5]	307
9.259 [3.58.4]	307
9.260 [3.58.3]	307
9.261 [3.58.2]	307
9.262 [3.58.1]	308
9.263 [3.58.0]	308
9.264 [3.57.3]	308
9.265 [3.57.2]	308
9.266 [3.57.1]	308
9.267 [3.57.0]	308
9.268 [3.56.16]	308
9.269 [3.56.15]	308

9.270 [3.56.14]	309
9.271 [3.56.13]	309
9.272 [3.56.12]	309
9.273 [3.56.11]	309
9.274 [3.56.10]	309
9.275 [3.56.9]	309
9.276 [3.56.8]	309
9.277 [3.56.7]	309
9.278 [3.56.6]	310
9.279 [3.56.5]	310
9.280 [3.56.4]	310
9.281 [3.56.3]	310
9.282 [3.56.2]	310
9.283 [3.56.1]	310
9.284 [3.56.0]	310
9.285 [3.55.3]	310
9.286 [3.55.2]	311
9.287 [3.55.1]	311
9.288 [3.55.0]	311
9.289 [3.54.2]	311
9.290 [3.54.1]	311
9.291 [3.54.0]	311
9.292 [3.53.4]	311
9.293 [3.53.3]	311
9.294 [3.53.2]	312
9.295 [3.53.1]	312
9.296 [3.53.0]	312
9.297 [3.52.0]	312
9.298 [3.51.1]	312
9.299 [3.51.0]	312
9.300 [3.50.0]	312
9.301 [3.49.10]	312
9.302 [3.49.9]	313
9.303 [3.49.8]	313
9.304 [3.49.7]	313
9.305 [3.49.6]	313
9.306 [3.49.5]	313
9.307 [3.49.4]	313
9.308 [3.49.3]	313
9.309 [3.49.2]	313
9.310 [3.49.1]	314
9.311 [3.49.0]	314
9.312 [3.48.0]	314
9.313 [3.47.2]	314
9.314 [3.47.1]	314
9.315 [3.47.0]	314
9.316 [3.46.6]	314
9.317 [3.46.5]	314
9.318 [3.46.4]	315
9.319 [3.46.3]	315
9.320 [3.46.2]	315
9.321 [3.46.1]	315
9.322 [3.46.0]	315
9.323 [3.44.4]	315

9.324 [3.44.3]	315
9.325 [3.44.2]	315
9.326 [3.44.1]	316
9.327 [3.44.0]	316
9.328 [3.43.1]	316
9.329 [3.43.0]	316
9.330 [3.42.5]	316
9.331 [3.42.4]	316
9.332 [3.42.3]	316
9.333 [3.42.2]	316
9.334 [3.42.1]	317
9.335 [3.42.0]	317
9.336 [3.41.13]	317
9.337 [3.41.12]	317
9.338 [3.41.11]	317
9.339 [3.41.10]	317
9.340 [3.41.9]	317
9.341 [3.41.8]	317
9.342 [3.41.7]	318
9.343 [3.41.6]	318
9.344 [3.41.5]	318
9.345 [3.41.4]	318
9.346 [3.41.3]	318
9.347 [3.41.2]	318
9.348 [3.41.0]	318
9.349 [3.40.16]	318
9.350 [3.40.15]	319
9.351 [3.40.14]	319
9.352 [3.40.13]	319
9.353 [3.40.12]	319
9.354 [3.40.11]	319
9.355 [3.40.10]	319
9.356 [3.40.9]	319
9.357 [3.40.8]	319
9.358 [3.40.7]	320
9.359 [3.40.6]	320
9.360 [3.40.5]	320
9.361 [3.40.4]	320
9.362 [3.40.3]	320
9.363 [3.40.2]	320
9.364 [3.40.1]	320
9.365 [3.40.0]	320
9.366 [3.39.1]	321
9.367 [3.39.0]	321
9.368 [3.38.7]	321
9.369 [3.38.6]	321
9.370 [3.38.5]	321
9.371 [3.38.4]	321
9.372 [3.38.3]	321
9.373 [3.38.2]	321
9.374 [3.38.1]	322
9.375 [3.38.0]	322
9.376 [3.37.0]	322
9.377 [3.36.13]	322

9.378 [3.36.12]	322
9.379 [3.36.11]	322
9.380 [3.36.10]	322
9.381 [3.36.9]	322
9.382 [3.36.8]	323
9.383 [3.36.7]	323
9.384 [3.36.6]	323
9.385 [3.36.5]	323
9.386 [3.36.4]	323
9.387 [3.36.3]	323
9.388 [3.36.2]	323
9.389 [3.36.1]	323
9.390 [3.36.0]	324
9.391 [3.35.3]	324
9.392 [3.35.2]	324
9.393 [3.35.1]	324
9.394 [3.35.0]	324
9.395 [3.34.2]	324
9.396 [3.34.1]	324
9.397 [3.34.0]	324
9.398 [3.33.12]	325
9.399 [3.33.11]	325
9.400 [3.33.10]	325
9.401 [3.33.9]	325
9.402 [3.33.8]	325
9.403 [3.33.7]	325
9.404 [3.33.6]	325
9.405 [3.33.5]	325
9.406 [3.33.4]	326
9.407 [3.33.3]	326
9.408 [3.33.2]	326
9.409 [3.33.1]	326
9.410 [3.33.0]	326
9.411 [3.32.0]	326
9.412 [3.31.1]	326
9.413 [3.31.0]	326
9.414 [3.30.14]	327
9.415 [3.30.13]	327
9.416 [3.30.12]	327
9.417 [3.30.11]	327
9.418 [3.30.10]	327
9.419 [3.30.9]	327
9.420 [3.30.8]	327
9.421 [3.30.7]	327
9.422 [3.30.6]	328
9.423 [3.30.5]	328
9.424 [3.30.4]	328
9.425 [3.30.3]	328
9.426 [3.30.2]	328
9.427 [3.30.1]	328
9.428 [3.30.0]	328
9.429 [3.29.0]	328
9.430 [3.28.24]	329
9.431 [3.28.23]	329

9.432 [3.28.22]	329
9.433 [3.28.21]	329
9.434 [3.28.20]	329
9.435 [3.28.19]	329
9.436 [3.28.18]	329
9.437 [3.28.17]	329
9.438 [3.28.16]	330
9.439 [3.28.15]	330
9.440 [3.28.14]	330
9.441 [3.28.13]	330
9.442 [3.28.12]	330
9.443 [3.28.11]	330
9.444 [3.28.10]	330
9.445 [3.28.9]	330
9.446 [3.28.8]	331
9.447 [3.28.7]	331
9.448 [3.28.6]	331
9.449 [3.28.5]	331
9.450 [3.28.4]	331
9.451 [3.28.3]	331
9.452 [3.28.2]	331
9.453 [3.28.1]	331
9.454 [3.28.0]	332
9.455 [3.27.27]	332
9.456 [3.27.26]	332
9.457 [3.27.25]	332
9.458 [3.27.24]	332
9.459 [3.27.23]	332
9.460 [3.27.22]	332
9.461 [3.27.21]	332
9.462 [3.27.20]	333
9.463 [3.27.19]	333
9.464 [3.27.18]	333
9.465 [3.27.17]	333
9.466 [3.27.16]	333
9.467 [3.27.15]	333
9.468 [3.27.14]	333
9.469 [3.27.13]	333
9.470 [3.27.12]	334
9.471 [3.27.11]	334
9.472 [3.27.10]	334
9.473 [3.27.9]	334
9.474 [3.27.8]	334
9.475 [3.27.7]	334
9.476 [3.27.6]	334
9.477 [3.27.5]	334
9.478 [3.27.4]	335
9.479 [3.27.3]	335
9.480 [3.27.2]	335
9.481 [3.27.1]	335
9.482 [3.27.0]	335
9.483 [3.26.23]	335
9.484 [3.26.22]	335
9.485 [3.26.21]	336

9.486 [3.26.20]	336
9.487 [3.26.19]	336
9.488 [3.26.18]	336
9.489 [3.26.17]	336
9.490 [3.26.16]	336
9.491 [3.26.15]	336
9.492 [3.26.14]	336
9.493 [3.26.13]	337
9.494 [3.26.12]	337
9.495 [3.26.11]	337
9.496 [3.26.10]	337
9.497 [3.26.9]	337
9.498 [3.26.8]	337
9.499 [3.26.7]	337
9.500 [3.26.6]	337
9.501 [3.26.5]	338
9.502 [3.26.4]	338
9.503 [3.26.3]	338
9.504 [3.26.2]	338
9.505 [3.26.1]	338
9.506 [3.26.0]	338
9.507 [3.25.2]	338
9.508 [3.25.1]	338
9.509 [3.25.0]	339
9.510 [3.24.0]	339
9.511 [3.23.12]	339
9.512 [3.23.11]	339
9.513 [3.23.10]	339
9.514 [3.23.9]	339
9.515 [3.23.8]	339
9.516 [3.23.7]	339
9.517 [3.23.6]	340
9.518 [3.23.5]	340
9.519 [3.23.4]	340
9.520 [3.23.3]	340
9.521 [3.23.2]	340
9.522 [3.23.1]	340
9.523 [3.23.0]	340
9.524 [3.22.16]	340
9.525 [3.22.15]	341
9.526 [3.22.14]	341
9.527 [3.22.13]	341
9.528 [3.22.12]	341
9.529 [3.22.11]	341
9.530 [3.22.10]	341
9.531 [3.22.9]	341
9.532 [3.22.8]	341
9.533 [3.22.7]	342
9.534 [3.22.6]	342
9.535 [3.22.5]	342
9.536 [3.22.4]	342
9.537 [3.22.3]	342
9.538 [3.22.2]	342
9.539 [3.22.1]	342

9.540 [3.22.0]	343
9.541 [3.21.4]	343
9.542 [3.21.3]	343
9.543 [3.21.2]	343
9.544 [3.21.1]	343
9.545 [3.21.0]	343
9.546 [3.20.5]	343
9.547 [3.20.4]	343
9.548 [3.20.3]	344
9.549 [3.20.2]	344
9.550 [3.20.1]	344
9.551 [3.20.0]	344
9.552 [3.19.0]	344
9.553 [3.18.7]	344
9.554 [3.18.6]	344
9.555 [3.18.5]	344
9.556 [3.18.4]	345
9.557 [3.18.3]	345
9.558 [3.18.2]	345
9.559 [3.18.1]	345
9.560 [3.18.0]	345
9.561 [3.17.47]	345
9.562 [3.17.46]	345
9.563 [3.17.45]	346
9.564 [3.17.44]	346
9.565 [3.17.43]	346
9.566 [3.17.42]	346
9.567 [3.17.41]	346
9.568 [3.17.40]	346
9.569 [3.17.39]	346
9.570 [3.17.37]	347
9.571 [3.17.36]	347
9.572 [3.17.35]	347
9.573 [3.17.34]	347
9.574 [3.17.33]	347
9.575 [3.17.32]	347
9.576 [3.17.31]	347
9.577 [3.17.30]	347
9.578 [3.17.29]	348
9.579 [3.17.28]	348
9.580 [3.17.27]	348
9.581 [3.17.26]	348
9.582 [3.17.25]	348
9.583 [3.17.24]	348
9.584 [3.17.23]	348
9.585 [3.17.22]	348
9.586 [3.17.21]	349
9.587 [3.17.20]	349
9.588 [3.17.19]	349
9.589 [3.17.18]	349
9.590 [3.17.17]	349
9.591 [3.17.16]	349
9.592 [3.17.15]	350
9.593 [3.17.14]	350

9.594 [3.17.13]	350
9.595 [3.17.12]	350
9.596 [3.17.11]	350
9.597 [3.17.10]	350
9.598 [3.17.9]	351
9.599 [3.17.8]	351
9.600 [3.17.7]	351
9.601 [3.17.6]	351
9.602 [3.17.5]	351
9.603 [3.17.4]	351
9.604 [3.17.3]	351
9.605 [3.17.2]	351
9.606 [3.17.1]	352
9.607 [3.17.0]	352
9.608 [3.16.11]	352
9.609 [3.16.10]	352
9.610 [3.16.9]	352
9.611 [3.16.8]	352
9.612 [3.16.7]	352
9.613 [3.16.6]	353
9.614 [3.16.5]	353
9.615 [3.16.4]	353
9.616 [3.16.3]	353
9.617 [3.16.2]	353
9.618 [3.16.1]	353
9.619 [3.16.0]	353
9.620 [3.15.0]	353
9.621 [3.14.1]	354
9.622 [3.14.0]	354
9.623 [3.13.12]	354
9.624 [3.13.11]	354
9.625 [3.13.10]	354
9.626 [3.13.9]	354
9.627 [3.13.8]	354
9.628 [3.13.7]	354
9.629 [3.13.6]	355
9.630 [3.13.5]	355
9.631 [3.13.4]	355
9.632 [3.13.3]	355
9.633 [3.13.2]	355
9.634 [3.13.1]	355
9.635 [3.13.0]	355
9.636 [3.12.4]	355
9.637 [3.12.3]	356
9.638 [3.12.2]	356
9.639 [3.12.1]	356
9.640 [3.12.0]	356
9.641 [3.11.1]	356
9.642 [3.11.0]	356
9.643 [3.10.5]	356
9.644 [3.10.4]	356
9.645 [3.10.3]	357
9.646 [3.10.2]	357
9.647 [3.10.1]	357

9.648 [3.10.0]	357
9.649 [3.9.13]	357
9.650 [3.9.12]	357
9.651 [3.9.11]	357
9.652 [3.9.10]	357
9.653 [3.9.9]	358
9.654 [3.9.8]	358
9.655 [3.9.7]	358
9.656 [3.9.6]	358
9.657 [3.9.5]	358
9.658 [3.9.4]	358
9.659 [3.9.3]	358
9.660 [3.9.2]	358
9.661 [3.9.1]	359
9.662 [3.9.0]	359
9.663 [3.8.43]	359
9.664 [3.8.42]	359
9.665 [3.8.41]	359
9.666 [3.8.40]	359
9.667 [3.8.39]	359
9.668 [3.8.38]	359
9.669 [3.8.37]	360
9.670 [3.8.36]	360
9.671 [3.8.35]	360
9.672 [3.8.34]	360
9.673 [3.8.33]	360
9.674 [3.8.32]	360
9.675 [3.8.31]	360
9.676 [3.8.30]	360
9.677 [3.8.29]	361
9.678 [3.8.28]	361
9.679 [3.8.27]	361
9.680 [3.8.26]	361
9.681 [3.8.25]	361
9.682 [3.8.24] 2020-10-02	361
9.683 [3.8.23] 2020-10-01	361
9.684 [3.8.22]	361
9.685 [3.8.21] 2020-09-28	362
9.686 [3.8.20] 2020-09-24	362
9.687 [3.8.19] 2020-09-24	362
9.688 [3.8.18] 2020-09-23	362
9.689 [3.8.17] 2020-09-23	362
9.690 [3.8.16] 2020-09-22	362
9.691 [3.8.15] 2020-09-22	362
9.692 [3.8.14] 2020-09-21	362
9.693 [3.8.13] 2020-09-20	363
9.694 [3.8.12] 2020-09-21	363
9.695 [3.8.11] 2020-09-18	363
9.696 [3.8.10] 2020-09-17	363
9.697 [3.8.9] 2020-09-16	363
9.698 [3.8.8] 2020-09-15	363
9.699 [3.8.7] 2020-09-15	363
9.700 [3.8.6] 2020-09-15	363
9.701 [3.8.5] 2020-09-14	364

9.702 [3.8.4] 2020-09-14	364
9.703 [3.8.3] 2020-09-14	364
9.704 [3.8.2] 2020-09-11	364
9.705 [3.8.1] 2020-09-10	364
9.706 [3.8.0] 2020-09-09	364
9.707 [3.7.8] 2020-09-09	364
9.708 [3.7.7] 2020-09-04	364
9.709 [3.7.6] 2020-09-09	365
9.710 [3.7.5] 2020-09-08	365
9.711 [3.7.4] 2020-09-04	365
9.712 [3.7.3] 2020-09-01	365
9.713 [3.7.2] 2020-09-01	365
9.714 [3.7.1] 2020-08-28	365
9.715 [3.7.0] 2020-08-27	365
9.716 [3.6.9] 2020-08-26	365
9.717 [3.6.8] 2020-08-26	366
9.718 [3.6.7] 2020-08-26	366
9.719 [3.6.6] 2020-08-24	366
9.720 [3.6.5] 2020-08-24	366
9.721 [3.6.4] 2020-08-18	366
9.722 [3.6.3] 2020-08-19	366
9.723 [3.6.2] 2020-08-17	366
9.724 [3.6.1] 2020-08-17	366
9.725 [3.6.0] 2020-08-12	367
9.726 [3.5.4] 2020-08-12	367
9.727 [3.5.3] 2020-08-11	367
9.728 [3.5.2] 2020-08-11	367
9.729 [3.5.1] 2020-08-11	367
9.730 [3.5.0] 2020-08-10	367
9.731 [3.4.40] 2020-08-05	367
9.732 [3.4.39] 2020-08-04	368
9.733 [3.4.38] 2020-08-04	368
9.734 [3.4.37] 2020-08-04	368
9.735 [3.4.36] 2020-08-04	368
9.736 [3.4.35] 2020-08-04	368
9.737 [3.4.34] 2020-08-03	368
9.738 [3.4.33] 2020-08-03	368
9.739 [3.4.32] 2020-07-31	368
9.740 [3.4.31] 2020-07-30	369
9.741 [3.4.30] 2020-07-29	369
9.742 [3.4.29] 2020-07-29	369
9.743 [3.4.28] 2020-07-24	369
9.744 [3.4.27] 2020-07-23	369
9.745 [3.4.26] 2020-07-20	369
9.746 [3.3.26] 2020-07-17	369
9.747 [3.3.25] 2020-07-16	370
9.748 [3.3.24] 2020-07-15	370
9.749 [3.3.23] 2020-07-13	370
9.750 [3.3.22] 2020-07-13	370
9.751 [3.3.21] - 2020-07-10	370
9.752 [3.3.20] - 2020-07-09	370
9.753 [3.3.19] - 2020-07-08	370
9.754 [3.3.18] - 2020-07-07	370
9.755 [3.3.17] - 2020-07-07	371

9.756 [3.3.16] - 2020-07-02	371
9.757 [3.3.15] - 2020-06-30	371
9.758 [3.3.14] - 2020-06-30	371
9.759 [3.3.13] - 2020-06-29	371
9.760 [3.3.12] - 2020-06-27	371
9.761 [3.3.11] - 2020-06-25	371
9.762 [3.3.10] - 2020-06-24	371
9.763 [3.3.9] - 2020-06-24	372
9.764 [3.3.8] - 2020-06-23	372
9.765 [3.3.7] - 2020-06-19	372
9.766 [3.3.6] - 2020-06-17	372
9.767 [3.3.5] - 2020-06-17	372
9.768 [3.3.4] - 2020-06-15	372
9.769 [3.3.3] - 2020-06-12	372
9.770 [3.3.2] - 2020-06-10	372
9.771 [3.3.1] - 2020-06-10	373
9.772 [3.3.0] - 2020-06-09	373
9.773 [3.2.22] - 2020-06-09	373
9.774 [3.2.21] - 2020-06-03	373
9.775 [3.2.20] - 2020-06-01	373
9.776 [3.2.19] - 2020-06-01	373
9.777 [3.2.18] - 2020-05-28	373
9.778 [3.2.17] - 2020-05-27	374
9.779 [3.2.16] - 2020-05-27	374
9.780 [3.2.15] - 2020-05-26	374
9.781 [3.2.14] - 2020-05-19	374
9.782 [3.2.13] - 2020-05-15	374
9.783 [3.2.12] - 2020-05-13	374
9.784 [3.2.11] - 2020-05-12	374
9.785 [3.2.10] - 2020-05-08	374
9.786 [3.2.9] - 2020-05-08	375
9.787 [3.2.8] - 2020-05-07	375
9.788 [3.2.7] - 2020-05-07	375
9.789 [3.2.6] - 2020-05-06	375
9.790 [3.2.5] - 2020-05-06	375
9.791 [3.2.4] - 2020-05-06	375
9.792 [3.2.3] - 2020-05-06	375
9.793 [3.2.2] - 2020-05-05	376
9.794 [3.2.1] - 2020-05-04	376
9.795 [3.2.0] - 2020-04-23	376
9.796 [3.1.3] - 2020-04-23	376
9.797 [3.1.2] - 2020-04-21	376
9.798 [3.1.1] - 2020-04-20	376
9.799 [3.1.0] - 2020-04-14	376
9.800 [3.0.15] - 2020-04-14	376
9.801 [3.0.14] - 2020-04-10	377
9.802 [3.0.13] - 2020-04-10	377
9.803 [3.0.12] - 2020-04-10	377
9.804 [3.0.11] - 2020-04-10	377
9.805 [3.0.10] - 2020-04-08	377
9.806 [3.0.9] - 2020-04-08	377
9.807 [3.0.8] - 2020-04-08	377
9.808 [3.0.7] - 2020-04-07	378
9.809 [3.0.6] - 2020-04-06	378

9.810 [3.0.5] - 2020-03-31	378
9.811 [3.0.4] - 2020-03-31	378
9.812 [3.0.3] - 2020-03-27	378
9.813 [3.0.2] - 2020-03-26	378
9.814 [3.0.1] - 2020-03-18	378
9.815 [3.0.0] - 2020-03-16	378
9.816 [2.5.5] - 2020-03-13	379
9.817 [2.5.4] - 2020-03-12	379
9.818 [2.5.3] - 2020-03-11	379
9.819 [2.5.2] - 2020-03-10	379
9.820 [2.5.1] - 2020-03-05	379
9.821 [2.5.0] - 2020-03-03	379
9.822 [2.4.2] - 2020-02-27	379
9.823 [2.4.1] - 2020-02-26	379
9.824 [2.4.0] - 2020-02-25	380
9.825 [2.3.9] - 2020-02-17	380
9.826 [2.3.8] - 2020-02-10	380
9.827 [2.3.7] - 2020-02-07	380
9.828 [2.3.6] - 2020-02-06	380
9.829 [2.3.4] - 2020-02-04	380
9.830 [2.3.3] - 2020-02-03	380
9.831 [2.3.2] - 2020-01-31	380
9.832 [2.3.1] - 2020-01-29	381
9.833 [2.3.0] - 2020-01-29	381
9.834 [2.2.0] - 2020-01-28	381
9.835 [2.1.7] - 2020-01-28	381
9.836 [2.1.6] - 2020-01-27	381
9.837 [2.1.5] - 2020-01-27	381
9.838 [2.1.4] - 2020-01-24	381
9.839 [2.1.03] - 2020-01-24	381
9.840 [2.1.01] - 2020-01-21	382
9.841 [2.1.0] - 2020-01-16	382
9.842 [2.0.50] - 2020-01-16	382
9.843 [2.0.49] - 2020-01-15	382
9.844 [2.0.48] - 2020-01-14	382
9.845 [2.0.47] - 2020-01-13	382
9.846 [2.0.46] - 2020-01-10	382
9.847 [2.0.45] - 2020-01-09	382
9.848 [2.0.43] - 2020-01-08	383
9.849 [2.0.42] - 2020-01-07	383
9.850 [2.0.41] - 2020-01-06	383
9.851 [2.0.40] - 2020-01-06	383
9.852 [2.0.39] - 2020-01-06	383
9.853 [2.0.38] - 2020-01-02	383
9.854 [2.0.37] - 2020-01-02	383
9.855 [2.0.36] - 2019-12-30	384
9.856 [2.0.35] - 2019-12-30	384
9.857 [2.0.34] - 2019-12-24	384
9.858 [2.0.33] - 2019-12-23	384
9.859 [2.0.32] - 2019-12-17	384
9.860 [2.0.31] - 2019-12-11	384
9.861 [2.0.30] - 2019-12-04	384
9.862 [2.0.29] - 2019-12-04	384
9.863 [2.0.28] - 2019-12-3	385

9.864 [2.0.27] - 2019-11-26	385
9.865 [2.0.26] - 2019-11-26	385
9.866 [2.0.25] - 2019-11-22	385
9.867 [2.0.24] - 2019-11-21	385
9.868 [2.0.23] - 2019-11-20	385
9.869 [2.0.22] - 2019-11-18	385
9.870 [2.0.21] - 2019-11-14	385
9.871 [2.0.20] - 2019-11-13	386
9.872 [2.0.19] - 2019-11-08	386
9.873 [2.0.19] - 2019-11-13	386
9.874 [2.0.17] - 2019-11-08	386
9.875 [2.0.16] - 2019-11-07	386
9.876 [2.0.15] - 2019-11-07	386
9.877 [2.0.14] - 2019-11-07	386
9.878 [2.0.13] - 2019-11-07	387
9.879 [2.0.12] - 2019-11-06	387
9.880 [2.0.11] - 2019-11-06	387
9.881 [2.0.10] - 2019-10-29	387
9.882 [2.0.9] - 2019-10-28	387
9.883 [2.0.8] - 2019-10-22	387
9.884 [2.0.7] - 2019-10-21	387
9.885 [2.0.6] - 2019-10-18	387
9.886 [2.0.5] - 2019-10-15	388
9.887 [2.0.4] - 2019-10-10	388
9.888 [2.0.3] - 2019-10-09	388
9.889 [2.0.2] - 2019-10-07	388
9.890 [2.0.1] - 2019-10-07	388
9.891 [2.0.0] - 2019-10-02	388
9.892 [1.11.0] - 2019-10-02	388
9.893 [1.10.8] - 2019-10-01	388
9.894 [1.10.7] - 2019-09-25	389
9.895 [1.10.6] - 2019-09-25	389
9.896 [1.10.5] - 2019-09-23	389
9.897 [1.10.4] - 2019-09-05	389
9.898 [1.10.3] - 2019-09-19	389
9.899 [1.10.2] - 2019-09-18	389
9.900 [1.10.1] - 2019-09-16	389
9.901 [1.10.0] - 2019-09-16	389
9.902 [1.9.12] - 2019-09-06	390
9.903 [1.9.11] - 2019-09-05	390
9.904 [1.9.10] - 2019-09-04	390
9.905 [1.9.9] - 2019-08-29	390
9.906 [1.9.8] - 2019-08-29	390
9.907 [1.9.7] - 2019-08-28	390
9.908 [1.9.6] - 2019-08-23	390
9.909 [1.9.5] - 2019-08-21	390
9.910 [1.9.4] - 2019-08-20	391
9.911 [1.9.3] - 2019-08-20	391
9.912 [1.9.2] - 2019-08-20	391
9.913 [1.9.1] - 2019-08-19	391
9.914 [1.9.0] - 2019-08-12	391
9.915 [1.8.9] - 2019-08-15	391
9.916 [1.8.8] - 2019-08-01	391
9.917 [1.8.7] - 2019-07-31	391

9.918 [1.8.6] - 2019-07-25	392
9.919 [1.8.5] - 2019-07-25	392
9.920 [1.8.4] - 2019-07-24	392
9.921 [1.8.3] - 2019-07-24	392
9.922 [1.8.2] - 2019-07-23	392
9.923 [1.8.1] - 2019-07-22	392
9.924 [1.8.0] - 2019-07-22	392
9.925 [1.7.3] - 2019-07-19	392
9.926 [1.7.2] - 2019-07-18	393
9.927 [1.7.1] - 2019-07-15	393
9.928 [1.7.0] - 2019-07-15	393
9.929 [1.6.23] - 2019-07-15	393
9.930 [1.6.22] - 2019-07-11	393
9.931 [1.6.21] - 2019-07-11	393
9.932 [1.6.20] - 2019-07-10	393
9.933 [1.6.19] - 2019-07-09	393
9.934 [1.6.18] - 2019-06-24	394
9.935 [1.6.17] - 2019-06-20	394
9.936 [1.6.16] - 2019-06-20	394
9.937 [1.6.15] - 2019-06-18	394
9.938 [1.6.14] - 2019-06-18	394
9.939 [1.6.13] - 2019-06-17	394
9.940 [1.6.12] - 2019-06-14	394
9.941 [1.6.11] - 2019-06-14	394
9.942 [1.6.10] - 2019-06-13	395
9.943 [1.6.9] - 2019-06-12	395
9.944 [1.6.8] - 2019-06-11	395
9.945 [1.6.7] - 2019-06-11	395
9.946 [1.6.6] - 2019-06-10	395
9.947 [1.6.5] - 2019-06-06	395
9.948 [1.6.4] - 2019-06-05	395
9.949 [1.6.3] - 2019-06-04	395
9.950 [1.6.2] - 2019-05-31	396
9.951 [1.6.1] - 2019-05-30	396
9.952 [1.6.0] - 2019-05-29	396
9.953 [1.5.9] - 2019-05-27	396
9.954 [1.5.8] - 2019-05-24	396
9.955 [1.5.7] - 2019-05-24	396
9.956 [1.5.6] - 2019-05-24	396
9.957 [1.5.5] - 2019-05-21	396
9.958 [1.5.4] - 2019-05-20	397
9.959 [1.5.3] - 2019-05-16	397
9.960 [1.5.2] - 2019-05-15	397
9.961 [1.5.1] - 2019-05-09	397
9.962 [1.5.0] - 2019-05-08	397
9.963 [1.4.10] - 2019-05-08	397
9.964 [1.4.9] - 2019-05-02	397
9.965 [1.4.8] - 2019-04-26	397
9.966 [1.4.7] - 2019-04-17	398
9.967 [1.4.6] - 2019-04-17	398
9.968 [1.4.5] - 2019-04-12	398
9.969 [1.4.4] - 2019-04-11	398
9.970 [1.4.3] - 2019-04-11	398
9.971 [1.4.2] - 2019-04-11	398

9.972 [1.4.1] - 2019-04-10	398
9.973 [1.4.0] - 2019-04-08	398
9.974 [1.3.11] - 2019-04-07	399
9.975 [1.3.10] - 2019-04-03	399
9.976 [1.3.9] - 2019-03-29	399
9.977 [1.3.8] - 2019-03-29	399
9.978 [1.3.7] - 2019-03-28	399
9.979 [1.3.6] - 2019-03-27	399
9.980 [1.3.5] - 2019-03-22	399
9.981 [1.3.4] - 2019-03-21	399
9.982 [1.3.3] - 2019-03-21	400
9.983 [1.3.2] - 2019-03-18	400
9.984 [1.3.1] - 2019-03-13	400
9.985 [1.3.0] - 2019-03-07	400
9.986 [1.2.12] - 2019-02-15	400
9.987 [1.2.11] - 2019-02-07	400
9.988 [1.2.10] - 2019-01-30	400
9.989 [1.2.9] - 2019-01-23	400
9.990 [1.2.8] - 2019-01-22	401
9.991 [1.2.7] - 2019-01-18	401
9.992 [1.2.5] - 2019-01-16	401
9.993 [1.2.4] - 2019-01-16	401
9.994 [1.2.3] - 2019-01-10	401
9.995 [1.2.2] - 2019-01-09	401
9.996 [1.2.1] - 2018-12-21	401
9.997 [1.2.0] - 2018-12-19	401
9.998 [1.1.4] - 2018-12-19	402
9.999 [1.1.3] - 2018-12-18	402
9.1000[1.1.2] - 2018-12-12	402
9.1001[1.1.1] - 2018-12-11	402
9.1002[1.1.0] - 2018-12-06	402
9.1003[1.0.6] - 2018-11-28	402
9.1004[1.0.5] - 2018-11-14	402
9.1005[1.0.4] - 2018-11-07	403
9.1006[1.0.3] - 2018-11-02	403
9.1007[1.0.2] - 2018-10-25	403
9.1008[1.0.1] - 2018-10-24	403
9.1009[1.0.0] - 2018-10-16	403
9.1010[0.73.6] - 2018-10-04	403
9.1011[0.73.5] - 2018-09-21	403
9.1012[0.73.4] - 2018-09-17	403
9.1013[0.73.3] - 2018-09-14	404
9.1014[0.73.2] - 2018-09-11	404
9.1015[0.73.1] - 2018-08-30	404
9.1016[0.73.0] - 2018-08-21	404
9.1017[0.72.7] - 2018-08-20	404
9.1018[0.72.6] - 2018-08-17	404
9.1019[0.72.5] - 2018-08-09	404
9.1020[0.72.4] - 2018-08-08	404
9.1021[0.72.3] - 2018-08-08	405
9.1022[0.72.2] - 2018-07-27	405
9.1023[0.72.1] - 2018-07-26	405
9.1024[0.72.0] - 2018-07-24	405
9.1025[0.71.2] - 2018-07-23	405

9.1026[0.71.1] - 2018-07-23	405
9.1027[0.71.0] - 2018-07-20	405
9.1028[0.70.8] - 2018-07-13	405
9.1029[0.70.7] - 2018-07-12	406
9.1030[0.70.6] - 2018-07-09	406
9.1031[0.70.5] - 2018-07-06	406
9.1032[0.70.4] - 2018-07-03	406
9.1033[0.70.3] - 2018-06-29	406
9.1034[0.70.2] - 2018-06-28	406
9.1035[0.70.1] - 2018-06-27	406
9.1036[0.70.0] - 2018-06-26	406
9.1037[0.69.6] - 2018-06-25	407
9.1038[0.69.5] - 2018-06-25	407
9.1039[0.69.4] - 2018-06-20	407
9.1040[0.69.3] - 2018-06-20	407
9.1041[0.69.2] - 2018-06-18	407
9.1042[0.69.1] - 2018-06-07	407
9.1043[0.69.0] - 2018-05-31	407
9.1044[0.68.9] - 2018-05-31	407
9.1045[0.68.8] - 2018-05-30	408
9.1046[0.68.7] - 2018-05-24	408
9.1047[0.68.6] - 2018-05-22	408
9.1048[0.68.5] - 2018-05-17	408
9.1049[0.68.4] - 2018-05-16	408
9.1050[0.68.3] - 2018-05-11	408
9.1051[0.68.2] - 2018-05-10	408
9.1052[0.68.1] - 2018-05-09	408
9.1053[0.68.0] - 2018-05-09	409
9.1054[0.67.8] - 2018-05-04	409
9.1055[0.67.7] - 2018-04-23	409
9.1056[0.67.6] - 2018-04-20	409
9.1057[0.67.5] - 2018-04-18	409
9.1058[0.67.4] - 2018-04-12	409
9.1059[0.67.3] - 2018-04-05	409
9.1060[0.67.2] - 2018-04-05	409
9.1061[0.67.1] - 2018-04-04	410
9.1062[0.67.0] - 2018-03-26	410
9.1063[0.66.2] - 2018-03-26	410
9.1064[0.66.1] - 2018-03-23	410
9.1065[0.66.0] - 2018-03-05	410
9.1066[0.65.8] - 2018-02-23	410
9.1067[0.65.7] - 2018-02-14	410
9.1068[0.65.6] - 2018-02-13	410
9.1069[0.65.5] - 2018-02-13	411
9.1070[0.65.4] - 2018-02-09	411
9.1071[0.65.3] - 2018-02-06	411
9.1072[0.65.2] - 2018-02-05	411
9.1073[0.65.1] - 2018-02-02	411
9.1074[0.65.0] - 2018-01-30	411
9.1075[0.64.0] - 2018-01-29	411
9.1076[0.63.0] - 2018-01-25	411
9.1077[0.62.0] - 2018-01-18	412
9.1078[0.61.6] - 2018-01-18	412
9.1079[0.61.5] - 2018-01-18	412

9.1080[0.61.4] - 2018-01-12	412
9.1081[0.61.3] - 2018-01-11	412
9.1082[0.61.2] - 2018-01-09	412
9.1083[0.61.1] - 2018-01-09	413
9.1084[0.61.0] - 2018-01-09	413
9.1085[0.60.0] - 2018-01-03	413
9.1086[0.59.0] - 2017-12-28	413
9.1087[0.58.0] - 2017-12-22	413
9.1088[0.57.0] - 2017-12-21	413
9.1089[0.56.5] - 2017-12-20	413
9.1090[0.56.4] - 2017-12-19	413
9.1091[0.56.3] - 2017-12-14	414
9.1092[0.56.2] - 2017-12-13	414
9.1093[0.56.1] - 2017-12-13	414
9.1094[0.56.0] - 2017-12-13	414
9.1095[0.55.7] - 2017-12-13	414
9.1096[0.55.6] - 2017-12-12	414
9.1097[0.55.5] - 2017-12-11	414
9.1098[0.55.4] - 2017-12-06	414
9.1099[0.55.3] - 2017-12-05	415
9.1100[0.55.2] - 2017-12-04	415
9.1101[0.55.1] - 2017-11-30	415
9.1102[0.55.0] - 2017-11-29	415
9.1103[0.54.1] - 2017-11-29	415
9.1104[0.54.0] - 2017-11-28	415
9.1105[0.53.19] - 2017-11-28	415
9.1106[0.53.18] - 2017-11-28	415
9.1107[0.53.17] - 2017-11-27	416
9.1108[0.53.16] - 2017-11-22	416
9.1109[0.53.15] - 2017-11-16	416
9.1110[0.53.14] - 2017-11-14	416
9.1111[0.53.13] - 2017-11-14	416
9.1112[0.53.12] - 2017-11-09	416
9.1113[0.53.11] - 2017-11-06	416
9.1114[0.53.10] - 2017-11-02	416
9.1115[0.53.9] - 2017-11-02	417
9.1116[0.53.8] - 2017-11-02	417
9.1117[0.53.7] - 2017-10-30	417
9.1118[0.53.6] - 2017-10-30	417
9.1119[0.53.5] - 2017-10-26	417
9.1120[0.53.4] - 2017-10-26	417
9.1121[0.53.3] - 2017-10-26	417
9.1122[0.53.2] - 2017-10-24	417
9.1123[0.53.1] - 2017-10-24	418
9.1124[0.53.0] - 2017-10-24	418
9.1125[0.52.10] - 2017-10-23	418
9.1126[0.52.9] - 2017-10-20	418
9.1127[0.52.8] - 2017-10-20	418
9.1128[0.52.7] - 2017-10-20	418
9.1129[0.52.6] - 2017-10-19	418
9.1130[0.52.5] - 2017-10-19	418
9.1131[0.52.4] - 2017-10-18	419
9.1132[0.52.3] - 2017-10-18	419
9.1133[0.52.2] - 2017-10-18	419

9.1134[0.52.1] - 2017-10-15	419
9.1135[0.52.0] - 2017-10-14	419
9.1136[0.51.5] - 2017-10-11	419
9.1137[0.51.4] - 2017-10-11	419
9.1138[0.51.3] - 2017-10-10	419
9.1139[0.51.2] - 2017-10-10	420
9.1140[0.51.1] - 2017-10-06	420
9.1141[0.51.0] - 2017-10-05	420
9.1142[0.50.1] - 2017-10-03	420
9.1143[0.50.0] - 2017-10-03	420
9.1144[0.49.0] - 2017-10-02	420
9.1145[0.48.2] - 2017-09-29	420
9.1146[0.48.1] - 2017-09-25	420
9.1147[0.48.0] - 2017-09-25	421
9.1148[0.47.1] - 2017-09-25	421
9.1149[0.47.0] - 2017-09-21	421
9.1150[0.46.8] - 2017-09-21	421
9.1151[0.46.7] - 2017-09-21	421
9.1152[0.46.6] - 2017-09-21	421
9.1153[0.46.5] - 2017-09-21	421
9.1154[0.46.4] - 2017-09-20	421
9.1155[0.46.3] - 2017-09-19	422
9.1156[0.46.2] - 2017-09-19	422
9.1157[0.46.1] - 2017-09-18	422
9.1158[0.46.0] - 2017-09-15	422
9.1159[0.45.0] - 2017-09-14	422
9.1160[0.44.0] - 2017-09-08	422
9.1161[0.43.5] - 2017-09-08	422
9.1162[0.43.4] - 2017-09-07	422
9.1163[0.43.3] - 2017-09-06	423
9.1164[0.43.2] - 2017-09-06	423
9.1165[0.43.1] - 2017-09-05	423
9.1166[0.43.0] - 2017-08-31	423
9.1167[0.42.0] - 2017-08-24	423
9.1168[0.41.0] - 2017-08-24	423
9.1169[0.40.7] - 2017-08-23	423
9.1170[0.40.6] - 2017-08-23	423
9.1171[0.40.5] - 2017-08-23	424
9.1172[0.40.4] - 2017-08-23	424
9.1173[0.40.3] - 2017-08-21	424
9.1174[0.40.2] - 2017-08-16	424
9.1175[0.40.1] - 2017-08-11	424
9.1176[0.40.0] - 2017-08-08	424
9.1177[0.39.9] - 2017-08-08	424
9.1178[0.39.8] - 2017-08-04	425
9.1179[0.39.7] - 2017-08-04	425
9.1180[0.39.6] - 2017-08-02	425
9.1181[0.39.5] - 2017-08-02	425
9.1182[0.39.4] - 2017-08-01	425
9.1183[0.39.3] - 2017-07-28	425
9.1184[0.39.2] - 2017-07-27	425
9.1185[0.39.1] - 2017-07-27	425
9.1186[0.39.0] - 2017-07-24	426
9.1187[0.38.7] - 2017-07-22	426

9.1188[0.38.6] - 2017-07-21	426
9.1189[0.38.5] - 2017-07-19	426
9.1190[0.38.4] - 2017-07-18	426
9.1191[0.38.3] - 2017-07-14	426
9.1192[0.38.2] - 2017-07-14	426
9.1193[0.38.1] - 2017-07-13	426
9.1194[0.38.0] - 2017-07-11	427
9.1195[0.37.1] - 2017-07-11	427
9.1196[0.37.0] - 2017-07-06	427
9.1197[0.36.11] - 2017-06-29	427
9.1198[0.36.10] - 2017-06-29	427
9.1199[0.36.9] - 2017-06-28	427
9.1200[0.36.8] - 2017-06-28	427
9.1201[0.36.7] - 2017-06-25	427
9.1202[0.36.6] - 2017-06-25	428
9.1203[0.36.5] - 2017-06-23	428
9.1204[0.36.4] - 2017-06-21	428
9.1205[0.36.3] - 2017-06-21	428
9.1206[0.36.2] - 2017-06-21	428
9.1207[0.36.1] - 2017-06-20	428
9.1208[0.36.0] - 2017-06-20	428
9.1209[0.35.2] - 2017-06-20	428
9.1210[0.35.1] - 2017-06-15	429
9.1211[0.35.0] - 2017-06-15	429
9.1212[0.34.7] - 2017-06-14	429
9.1213[0.34.6] - 2017-06-14	429
9.1214[0.34.5] - 2017-06-09	429
9.1215[0.34.4] - 2017-06-09	429
9.1216[0.34.3] - 2017-06-07	429
9.1217[0.34.2] - 2017-06-06	429
9.1218[0.34.1] - 2017-06-06	430
9.1219[0.34.0] - 2017-06-05	430
9.1220[0.33.24] - 2017-06-02	430
9.1221[0.33.23] - 2017-06-02	430
9.1222[0.33.22] - 2017-06-02	430
9.1223[0.33.21] - 2017-06-01	430
9.1224[0.33.20] - 2017-05-23	430
9.1225[0.33.19] - 2017-05-18	430
9.1226[0.33.18] - 2017-05-17	431
9.1227[0.33.17] - 2017-05-16	431
9.1228[0.33.16] - 2017-05-15	431
9.1229[0.33.15] - 2017-05-10	431
9.1230[0.33.14] - 2017-05-10	431
9.1231[0.33.13] - 2017-05-09	431
9.1232[0.33.12] - 2017-05-03	431
9.1233[0.33.11] - 2017-05-02	431
9.1234[0.33.10] - 2017-05-02	432
9.1235[0.33.9] - 2017-04-26	432
9.1236[0.33.8] - 2017-04-26	432
9.1237[0.33.7] - 2017-04-24	432
9.1238[0.33.6] - 2017-04-21	432
9.1239[0.33.5] - 2017-04-20	432
9.1240[0.33.4] - 2017-04-18	432
9.1241[0.33.3] - 2017-04-18	432

9.1242[0.33.2] - 2017-04-17	433
9.1243[0.33.1] - 2017-04-17	433
9.1244[0.33.0] - 2017-04-11	433
9.1245[0.32.1] - 2017-04-06	433
9.1246[0.32.0] - 2017-04-06	433
9.1247[0.31.4] - 2017-04-05	433
9.1248[0.31.3] - 2017-04-04	433
9.1249[0.31.2] - 2017-04-03	433
9.1250[0.31.1] - 2017-03-31	434
9.1251[0.31.0] - 2017-03-30	434
9.1252[0.30.0] - 2017-03-27	434
9.1253[0.29.1] - 2017-03-27	434
9.1254[0.29.0] - 2017-03-23	434
9.1255[0.28.0] - 2017-03-23	434
9.1256[0.27.6] - 2017-03-21	434
9.1257[0.27.5] - 2017-03-17	434
9.1258[0.27.4] - 2017-03-17	435
9.1259[0.27.3] - 2017-03-16	435
9.1260[0.27.2] - 2017-03-10	435
9.1261[0.27.1] - 2017-03-13	435
9.1262[0.27.0] - 2017-03-02	435
9.1263[0.26.3] - 2017-03-02	435
9.1264[0.26.2] - 2017-03-02	435
9.1265[0.26.1] - 2017-02-28	435
9.1266[0.26.0] - 2017-02-28	436
9.1267[0.25.0] - 2017-02-28	436
9.1268[0.24.0] - 2017-02-27	436
9.1269[0.23.2] - 2017-02-22	436
9.1270[0.22.1] - 2017-02-20	436
9.1271[0.22.0] - 2017-02-14	436
9.1272[0.21.2] - 2017-02-07	436
9.1273[0.21.0] - 2017-01-30	436
9.1274[0.20.0] - 2017-01-30	437
9.1275[0.19.1] - 2017-01-30	437
9.1276[0.19.0] - 2017-01-30	437
9.1277[0.18.0] - 2017-01-27	437
9.1278[0.17.0] - 2017-01-25	437
9.1279[0.16.0] - 2017-01-25	437
9.1280[0.15.0] - 2017-01-25	437
9.1281[0.14.0] - 2017-01-20	437
9.1282[0.13.0] - 2017-01-06	438
9.1283[0.12.0] - 2017-01-05	438
9.1284[0.11.0] - 2017-01-05	438
9.1285[0.10.0] - 2017-01-04	438
9.1286[0.9.0] - 2016-12-29	438
9.1287[0.8.0] - 2016-12-08	438
9.1288[0.7.0] - 2016-12-07	438
9.1289[0.6.0] - 2016-12-04	438
9.1290[0.5.0] - 2016-11-28	439
9.1291[0.4.1] - 2016-11-24	439
9.1292[0.4.0] - 2016-11-21	439
9.1293[0.3.1] - 2016-11-21	439
9.1294[0.3.0] - 2016-11-16	439
9.1295[0.2.0] - 2016-11-15	439

9.1296[0.1.2] - 2016-11-04	440
9.1297[0.1.1] - 2016-11-03	440
9.1298[0.1.0] - 2016-10-13	440
10 Multiple Enterprise Support	441
10.1 Status	441
10.2 Context	441
10.3 Decisions	441
10.4 Consequences	442
11 New Enterprise Catalog IDA	443
11.1 Status	443
11.2 Context	443
11.3 Decisions	444
11.4 Consequences	446
12 Canvas Integration via Classic Integration model vs LTI	447
12.1 Status	447
12.2 Context	447
12.3 Decision	447
12.4 Consequences	447
13 Enterprise role assignments will record a customer UUID	449
13.1 Status	449
13.2 Context	449
13.3 Decisions (interspersed with Consequences)	449
14 Bulk Enrollment endpoint for Subscription learners	451
14.1 Status	451
14.2 Context	451
14.3 Decisions	451
14.4 Consequences	452
15 Adding enterprise_uuid in course enrollment event	453
15.1 Status	453
15.2 Context	453
15.3 Decision	453
15.4 Consequences	453
16 Transmitting Only Updates IDA	455
16.1 Status	455
16.2 Context	455
16.3 Decisions	456
16.4 Exceptions	456
16.5 Consequences	456
17 Determining course completion for various course modes	457
17.1 Status	457
17.2 Context	457
17.3 Decisions	457
17.4 Consequences	458
18 Adding course key - uuid mapping for Cornerstone	459
18.1 Status	459
18.2 Context	459

18.3	Decisions	459
18.4	Consequences	460
19	Zero state browsing with universal link	461
19.1	Status	461
19.2	Context	461
19.3	Decision	462
19.4	Phase 1 - Browsing via the learner portal search page	462
19.5	Phase 2 - Browsing anonymously	463
20	Moving the responsibility of catalog diffing to the enterprise-catalog service	465
20.1	Status	465
20.2	Context	465
20.3	Decisions	467
20.4	Consequences	467
20.5	Further Improvements	468
21	Enterprise subsidy enrollments and entitlements	469
21.1	Status	469
21.2	Context	469
21.3	Decisions	469
21.4	A rework of the enterprise subsidy enrollment models and creation of enterprise entitlements	470
21.5	Consequences	471
21.6	Further Improvements	471
21.7	Alternatives Considered	472
22	Waffle-based feature flags for Enterprise	473
22.1	Status	473
22.2	Context	473
22.3	Decisions	473
22.4	Consequences	473
22.5	Further Improvements	474
22.6	Alternatives Considered	474
23	0013 Creating Custom Groups for Enterprise Customers	475
23.1	Status	475
23.2	Terminology	475
23.3	Context	475
23.4	EnterpriseGroup	475
23.5	EnterpriseGroupMembership	478
23.6	Consequences	478
23.7	Further Improvements	478
	Python Module Index	479
	Index	481

Your project description goes here

Contents:

ENTERPRISE

The `Open edX Enterprise Service` app provides enterprise features to the Open edX platform. The majority of these features are structured around the concept of an `Enterprise Customer`, which is an organization or a group of people that “consumes” courses published on the Open edX platform.

1.1 Documentation

Full documentation for the `Open edX Enterprise Service` can be found at <http://open-edx-enterprise-service-documentation.readthedocs.io/en/latest/>.

1.2 License

The code in this repository is licensed under the AGPL 3.0 unless otherwise noted.

Please see `LICENSE.txt` for details.

1.3 How To Contribute

Contributions are very welcome.

Please read [How To Contribute](#) for details.

A Pull Request Description Template can be found at `PULL_REQUEST_TEMPLATE.md`` - this template is automatically applied when you open a pull request from GitHub. Please make sure to include this template if submitting a pull request via other channels.

After submitting a pull request, please use the Github “Reviewers” widget to add relevant reviewers and track review process.

1.4 Reporting Security Issues

Please do not report security issues in public. Please email security@openedx.org.

1.5 Getting Help

Have a question about this repository, or about Open edX in general? Please refer to this [list of resources](#) if you need any assistance.

GETTING STARTED

2.1 Production setup

Edx-enterprise is developed as a pluggable application for the edx-platform and can't currently be used outside of edx-platform. Edx-enterprise is shipped with default edx-platform setup (see [edx-platform requirements file](#)), so new installs should already have it set up and enabled.

If you're migrating from an earlier (i.e. pre-Ficus) release, the only step you *might* have to do manually is to perform database migrations.

```
$ make migrate  
  
# Or use a more down-to-the-root command (replace aws with your version of config)  
$ ./manage.py lms migrate --settings=aws
```

2.2 Local development

If you have not already done so, create/activate a [virtualenv](#).

Alternatively, [docker](#) can be used to provide a containerized shell to run tests with.

```
$ make test-shell
```

Dependencies can be installed via the command below.

```
$ make requirements
```

Then you might want to run tests to make sure the setup went fine and there are no pre-existing problems (i.e. failed tests or quality checks)

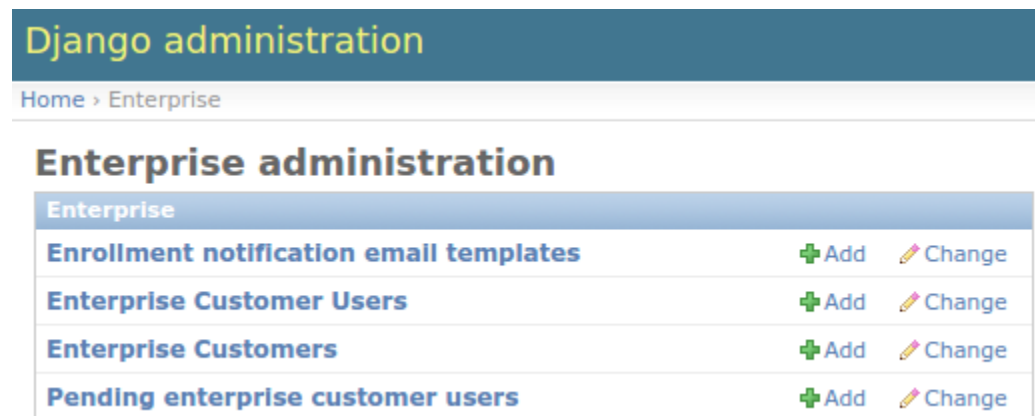
```
$ make test-all
```

For details on performing other development-related tasks and high-level overview of `edx-enterprise` architecture and development principles, see [Development documentation](#)

CONFIGURATION AND USAGE

3.1 Getting around the administrator interface

The `edx-enterprise` project is integrated with `edx-platform` Django admin. It can be accessed at `$LMS_SERVER_ADDRESS/admin/enterprise`. The current administrator interface is considered provisional until an independent full-featured enterprise portal is built, so not much time was spent on improving admin tools UX.



The screenshot shows the Django administration interface for the Enterprise app. At the top, there's a header with the text "Django administration" in yellow on a dark blue background. Below this is a breadcrumb trail: "Home > Enterprise". The main heading is "Enterprise administration". Below the heading is a table with four rows, each representing a different admin view. Each row has a title and two action links: "Add" (with a green plus icon) and "Change" (with a yellow pencil icon).

Enterprise	
Enrollment notification email templates	+ Add Change
Enterprise Customer Users	+ Add Change
Enterprise Customers	+ Add Change
Pending enterprise customer users	+ Add Change

So far, there are four items in the admin interface:

- “Enrollment notification email templates” - manages templates used to build enrollment notifications.
- “Enterprise Customers” - lists and manages Enterprise Customer records.
- “Enterprise Customer Users” - lists and manages students associated with Enterprise Customers.
- “Pending Enterprise Customer Users” - lists and manages “pending” Enterprise Customer students.

Django administration Welcome, **staff**. [View site](#) / [Change password](#) / [Log out](#)

Home > Enterprise > Enterprise Customers > <EnterpriseCustomer 9265c76d-224a-4333-9d6a-8651d5b7c442: Seal Team Ten>

Change Enterprise Customer

[Manage Learners](#) [History](#)

Name:
Enterprise Customer name.

Catalog:

☒ **Active**

Site:

☐ **Enable data sharing consent**
This field is used to determine whether data sharing consent is enabled or disabled for users signing in using this enterprise customer. If disabled, consent will not be requested, and eligible data will not be shared.

Enforce data sharing consent:
This field determines if data sharing consent is optional, if it's required at login, or if it's required when registering for eligible courses.

Branding Configurations

Branding Configuration: #1

Logo: No file chosen
Please add only .PNG files for logo images.

Enterprise customer identity providers

Enterprise customer identity provider: #1

Provider id:

Enterprise Customer Entitlements

[Add another Enterprise Customer Entitlement](#)

[Delete](#) [Save and add another](#) [Save and continue editing](#) [Save](#)

The `edx-enterprise` admin site provides two ways to manage learners: The “Enterprise Customer Users” section mentioned above, and the “Manage Learners” view. The “Manage Learners” view is recommended as it is easier to understand and provides more features. To access it, first go to the “Enterprise Customer” section of the admin site, then click on a customer, and then click on the “Manage Learners” button in the top-right corner of the page.

3.2 Manage Learners View

The Manage Learners view is meant to be used by non-technical staff to administer enterprise learners. It can be used to:

- Associate (“link”) and disassociate (“unlink”) existing students with an Enterprise Customer.
- Associate learners with an Enterprise customer even if they don’t have an account yet (these learners are known as “Pending Enterprise Customer Learners”). Pending Enterprise Customer Learners are identified by their email address only.
- Manually enroll groups of enterprise learners into courses and/or programs.

Django administration Welcome, **staff**. View site / Change password / Log out

Home > Enterprise > Enterprise Customers > <EnterpriseCustomer 9265c76d-224a-4333-9d6a-8651d5b7c442: Seal Team Ten> > Manage Learners

Search email address or username

Linked learners

User Email	Username	Linked Date	
qwe@asd.com	QWE	Jan. 23, 2017, 10:10 a.m.	<input type="button" value="Unlink"/>
seal_ten_alpha@example.com	SealTenAlpha	Dec. 23, 2016, 5:13 a.m.	<input type="button" value="Unlink"/>

Pending linked learners

User Email	Linked Date	
qwe@ad.com	Jan. 13, 2017, 9:28 a.m.	<input type="button" value="Unlink"/>
asd@asd.com	Jan. 13, 2017, 9:31 a.m.	<input type="button" value="Unlink"/>

Link learners

Type in Email or Username to link single user:

Or upload a CSV to enroll multiple users at once:

No file chosen

The CSV must have a column of email addresses, indicated by the heading 'email' in the first row.

Also enroll these learners in this course:

Provide a course ID if enrollment is desired.

Or enroll these learners in this program:

Provide a program name or ID if enrollment is desired.

Course enrollment mode:

----- ▾

Notify learners of enrollment: ▾

3.2.1 Integrating with the Course Catalog

In order for course enrollment features to work, **edx-enterprise** talks to the [Course Catalog Service](#). Instructions on how to set up and configure the Course Catalog service can be found at corresponding [docs section](#) of Course Catalog and in **edx-enterprise Pull Request #7** (describes developer setup, but useful for a broader audience).

One particular quirk is that the Course Catalog Service, being an independent application, restricts access to courses and programs according to ownership rules, so only users with certain roles can list all programs and courses, as the **edx-enterprise** admin interface expects. This is covered in greater detail in aforementioned docs section.

3.2.2 Managing Linked Learners Lists

The Manage Learners view allows for two modes of linking users to an Enterprise Customer:

- Singular - A single user email address or username is provided.
- Bulk - A CSV file containing a single column of email addresses is uploaded.

When linking a single user by username, **edx-enterprise** tries to find an existing user with that username and fails the linking if a match was not found. When email address is used (either in singular or bulk mode), existing users using that email address are linked to the Enterprise Customer. If an email address match was not found, a “Pending Linked Learner” record is created for that email address. If that email address is used to register a new user, then that user is automatically linked with the Enterprise Customer.

Note: Each learner can be linked only to one Enterprise Customer, so linking fails if the learner is already associated with some other enterprise customer.

The Manage Learners view also enables unlinking “Linked Learners” and “Pending Linked Learners.”

3.2.3 Enrolling Learners into a Course or Program

Enrolling learners into courses or programs depends on Course Catalog integration, because `edx-enterprise` uses the Course Catalog API to fetch course and program information.

When “Course ID” input is filled, “Program ID” input is blocked and “Course Enrollment Mode” is automatically populated with course modes available for chosen course.

When “Program ID” input is filled, “Course ID” input is blocked and “Course Enrollment Mode” is reset to a list of all course enrollment modes¹

3.3 Enrollment notification email templates

When learners are enrolled into a course or program, admins might choose to send enrollment notification emails. This admin section manages templates used to build such emails.

The template engine supports plaintext and HTML emails and allows substituting certain placeholders with real values, which are pulled from enrollment data (i.e. username, organization, course or program name, etc.). For details, refer to the help strings provided by the form.

¹ Course Catalog Service API does not expose any means to get a list of modes supported by *all* courses in the program, so it relies on the administrator to choose the right mode. However, validation is performed on the back end, and if an unavailable mode is chosen, a list of available modes is shown in an error message.

Django administration
Welcome, **staff**. View site / Change password / Log out

Home > Enterprise > Enrollment notification email templates > <EnrollmentNotificationEmailTemplate for site with ID 1>

Change enrollment notification email template
Preview (course) Preview (program) History

Plaintext template:

I'm enrollment notification template - nice to meet you {{user_name}}!

{{organization_name}} decided it would be lovely to have a great guy like you in their numbers, so they decided to enroll you into a {{enrolled_in.type}} named "{{enrolled_in.name}}" starting immediately!

Just kidding, it actually starts {{enrolled_in.start|date}}, so you've got some time to prepare. Not that I'm sure about that - I'm just a template and template render didn't pass current date, so I have no idea what day it is now - but as far as I know people try to send emails such as this in advance.

So, have fun and good luck.

Fill in a standard Django template that, when rendered, produces the email you want sent to newly-enrolled Enterprise Customer users. The following variables may be available:

- user_name: A human-readable name for the person being emailed. Be sure to handle the case where this is not defined, as it may be missing in some cases. It may also be a username, if the user hasn't configured their "real" name in the system.
- organization_name: The name of the organization sponsoring the enrollment.
- enrolled_in: Details of the course or program that was enrolled in. Possible items it contains:
 - name: The name of the enrollable item (e.g., "Demo Course").
 - url: A link to the homepage of the enrolled-in item.
 - branding: A custom branding name for the enrolled-in item. For example, the branding of a MicroMasters program would be "MicroMasters".
 - start: The date the enrolled-in item becomes available. Render this to text using the Django `date` template filter (see the Django documentation).
 - type: Whether the enrolled-in item is a course, a program, or something else.

Html template:

```
<p>I'm enrollment notification template - nice to meet you {{user_name}}!</p>
<p>{{organization_name}} decided it would be lovely to have a great guy like you in their numbers, so they decided to enroll you into a {{enrolled_in.type}} named "{{enrolled_in.name}}" starting immediately!</p>
<p>Just kidding, it actually starts {{enrolled_in.start|date}}, so you've got some time to prepare. Not that I'm sure about that - I'm just a template and template render didn't pass current date, so I have no idea what day it is now - but as far as I know people try to send emails such as this in advance.</p>
<p>So, have fun and good luck.</p>
```

Fill in a standard Django template that, when rendered, produces the email you want sent to newly-enrolled Enterprise Customer users. The following variables may be available:

- user_name: A human-readable name for the person being emailed. Be sure to handle the case where this is not defined, as it may be missing in some cases. It may also be a username, if the user hasn't configured their "real" name in the system.
- organization_name: The name of the organization sponsoring the enrollment.
- enrolled_in: Details of the course or program that was enrolled in. Possible items it contains:
 - name: The name of the enrollable item (e.g., "Demo Course").
 - url: A link to the homepage of the enrolled-in item.
 - branding: A custom branding name for the enrolled-in item. For example, the branding of a MicroMasters program would be "MicroMasters".
 - start: The date the enrolled-in item becomes available. Render this to text using the Django `date` template filter (see the Django documentation).
 - type: Whether the enrolled-in item is a course, a program, or something else.

Subject line:

Fill in a string that can be used to generate a dynamic subject line for notification emails. The placeholder {course_name} will be replaced with the name of the course or program that was enrolled in.

Site: example.com

Delete Save and add another Save and continue editing Save

You can preview emails in the template edit view using the “Preview (program)” and “Preview (course)” buttons in top-right corner.

3.4 Integrated Channels

3.4.1 SAP SuccessFactors Configuration

One of the integrated channels available for an `EnterpriseCustomer` is [SAP SuccessFactors](#).

To integrate an `EnterpriseCustomer` with SAP SuccessFactors:

- Set up a [TPA SAML Integration](#) for the SAP SuccessFactors endpoint.
- From the LMS Admin, ensure that your `EnterpriseCustomer` record has its `identity_provider` field pointing to the SAML Integration created in the previous step.

This allows learners to create accounts using Single Sign-On from SAP SuccessFactors, and to have their course completion records sent back to SAP SuccessFactors.

- From the LMS Admin, create an `SAP SuccessFactors Enterprise Customer Configuration` record, and link it to your `EnterpriseCustomer`.

Ensure that the record is marked `Active` to allow the management commands below to send data via this channel.

3.4.2 Management Commands

These Django management commands send data to EnterpriseCustomer’s active, integrated channels.

They must be run from the command line on a server with the Open edX environment installed.

Note: If data sharing consent is enabled for your EnterpriseCustomer, then any Enterprise learners enrolled in the associated courses must consent to data sharing, thus permitting their data to be sent back to SAP SuccessFactors.

If data sharing consent is *not* enabled for your EnterpriseCustomer, then learner data may be sent to SAP SuccessFactors without their explicit consent, so use these settings with care.

Transmit Learner Data

The `transmit_learner_data` command sends course completion data for each EnterpriseCustomerCourseEnrollment where evidence of course completion exists for an enrolled learner.

Note: “Course completion” is determined by differently for different types of courses.

- Instructor-paced course enrollments are deemed “complete” by the presence of a certificate for a given learner and course. The grade reported is pulled from the certificate.
- Self-paced courses with an end date are deemed “complete” once the end date has passed. The grade reported is “Pass” or “Fail”, pulled from the Grades API, as of the current date.
- Self-paced courses with no end date are deemed “complete” once the learner passes the course. If the learner has not yet passed the course, the grade is reported as “In Progress”.

Usage

```
# Login as the edxapp user, and enable the edxapp environment.
$ sudo -u edxapp -Hs
$ cd ~
$ source edxapp_env # adds $EDX_PLATFORM_SETTINGS to the environment, e.g. aws, ↵
↵openstack, devstack..

# View command help
$ ./manage.py lms transmit_learner_data --help --settings=$EDX_PLATFORM_SETTINGS

# Transmit learner data for all EnterpriseCustomers, to all active integrated channels.
# * --api_user must be a user with staff access to all the courses linked to the ↵
↵EnterpriseCustomers.
$ ./manage.py lms transmit_learner_data --api_user staff --settings=$EDX_PLATFORM_ ↵
↵SETTINGS

# Transmit learner data for a single EnterpriseCustomer, e.g. with uuid 12
$ ./manage.py lms transmit_learner_data --api_user staff --enterprise-customer 12 -- ↵
↵settings=$EDX_PLATFORM_SETTINGS

# Transmit learner data only to SAP SuccessFactors
```

(continues on next page)

(continued from previous page)

```
$ ./manage.py lms transmit_learner_data --api_user staff --channel SAP --settings=$EDX_  
↪PLATFORM_SETTINGS
```


DEVELOPMENT DOCUMENTATION

4.1 High-level Architecture Overview

`Edx-enterprise` is a pluggable Django application designed to be run within `edx-platform`. It is shipped with any `edx-platform` install, but can be disabled or even removed from a particular Open edX instance.

General direction is to decouple `edx-enterprise` from the rest of the `edx-platform` as much as possible, since the overall goal is to eventually make it an independent application or a plugin for a separate admin site, and not be part of the `edx-platform` LMS.

In order to maintain the separation of `edx-enterprise` from the `edx-platform`, follow these practices:

- When integrating with existing `edx-platform` features, use REST APIs rather than Python APIs.
- In cases where the API in question only exists as a Python API, it should be encapsulated into a module or class on the `edx-enterprise` side, to simplify future migration to a REST API.
- Both REST API and Python API clients must be covered with extensive unit tests.

Other noteworthy points:

- `edx-enterprise` does not interact with Studio.
- `edx-enterprise` supports multiple python versions:
 - Python 2.7 and python 3.5.
- High test coverage is expected - all new python code must have 100% diff coverage. JS test were only recently introduced, so it's not as strict so far, but aim for 100% diff coverage as well.
- High code quality standards are expected - both on python and JS sides. Multiple static analyzers are used to enforce it (`pycodestyle`, `pylint`, `isort`, `jshint`, etc.).

4.2 Local setup

Although `edx-enterprise` can't be used outside `edx-platform`, some development process steps are intended to be performed in a standalone virtual environment.

If you have not already done so, create/activate a `virtualenv`. Unless otherwise stated, assume all terminal code below is executed within the `virtualenv`.

Alternatively, `docker` can be used to provide a containerized shell to run the commands below inside.

```
$ make test-shell
```

4.2.1 Install dependencies

Dependencies can be installed via the command below.

```
$ make requirements
```

4.2.2 Run tests

This application uses `tox` to run tests under multiple Python and Django version. In addition, the following test utilities are used:

- `pytest` framework and test runner to run test
- `pytest-cov` plugin for collecting test coverage
- `diff-cover` for analyzing changeset coverage

The `edx-enterprise` Makefile provides multiple shortcuts to running tests. The most notable are:

```
$ make test                # run all tests in current virtualenv
$ make diff_cover          # run all tests in current virtualenv and report diff_
↪ coverage
# run quality checks, all tests in every supported env (Python and Django versions) and_
↪ jasmine JS tests
$ make test-all
```

More info about running tests is available in [Testing](#) section.

4.2.3 Check quality

A whole bunch of static analyzers are used to ensure code quality. Most notable are `pycodestyle` (formerly `pep8`), `pylint` and `jshint`.

```
$ tox -e quality          # runs all python and docs quality check
$ make jshint              # runs jshint on files in enterprise folder
```

4.2.4 Build (these) docs

This documentation is generated by `Sphinx`. Essentially, there are two parts of documentation - manually written (located in `docs` folder) and generated from docstrings.

```
$ tox -e docs             # performs docs quality check and generates docs in HTML
$ make docs               # same, but also opens browser at the index page
```

4.2.5 Update translations

See *Internationalization* chapter for details.

4.2.6 Upgrading local setup from older versions

If you're migrating from an older version (i.e. pre Nov 2016) of edx-platform, you might need to ensure edx-enterprise is installed correctly. Three things need to happen:

1. edx-enterprise must be installed in edxapp env.
2. edx-enterprise must be added to INSTALLED_APPS.
3. Migrations need to be run.

All three should happen automatically if you use paver commands to upgrade your setup, but just in case something goes wrong with the setup, here are instructions to manually perform the upgrade.

First, install edx-enterprise into virtualenv. In edxapp virtualenv (`$current_release` is 4.17.0)¹

```
$ cd /edx/app/edxapp/edx-platform
$ pip install edx-enterprise==$current_release
```

Then, make sure edx-enterprise is included in INSTALLED_APPS or OPTIONAL_APPS (see `lms/env/common.py` as an example) and run migrations:

```
$ make migrate
# Or use a more down-to-the-root command (replace aws with your version of config)
$ ./manage.py lms migrate --settings=devstack
```

¹ Due to limitations of Sphinx formatting, it is impossible to inject current version into code block while retaining formatting.

TESTING

edx-enterprise has an assortment of test cases and code quality checks to catch potential problems during development.

5.1 Running all of the python tests

To run all unit tests and quality checks in the version of Python you chose for your virtualenv.

Alternatively, [docker](#) can be used to provide a containerized shell to run the commands below inside.

```
$ make test-shell
```

```
$ make validate
```

To run just the unit tests:

```
$ make test
```

To run just the unit tests and check diff coverage

```
$ make diff_cover
```

To run the unit tests under every supported Python version and the code quality checks:

```
$ make test-all
```

To run all tests under certain python versions and edx-platform dependency environments:

```
$ tox -e py35-master # run all tests under python 3.5 and master branch dependencies
```

5.2 Code coverage

To generate and open an HTML report of how much of the code is covered by test cases:

```
$ make coverage
```

There is a useful `pytest.local.ini` file that helps with looking at coverage of only a single module at a time:

```
$ pytest tests/test_enterprise/api -c pytest.local.ini --cov=enterprise.api
```

5.3 Running subsets of tests

Various options to run only subset of tests:

```
$ pytest tests/test_admin/           # run all tests in tests/admin folder
$ pytest tests/test_enterprise/api    # run all tests in tests/test_enterprise_api_
↳ folder
$ pytest tests/test_enterprise/api/test_permissions.py # run all the tests in the test_
↳ permissions.py file

# run all tests in TestEnterpriseCustomer test suite file
$ pytest tests/test_models.py::TestEnterpriseCustomer

# run only `test_ready_connects_user_post_save_handler` in `TestEnterpriseConfig` suite
$ pytest tests/test_apps.py::TestEnterpriseConfig::test_ready_connects_user_post_save_
↳ handler
```

5.4 Quality

To run just the code quality checks:

```
$ tox -e quality
```

To run quality checks on specific files:

```
# run the PEP8-style checks on one file (fast)
$ pycodestyle enterprise/api/v1/views.py

# run pylint on one file (not as fast)
$ pylint enterprise/api/v1/views.py

# use isort to fix imports, --check-only means see what isort would change without_
↳ actually changing it
$ isort --check-only enterprise/api/v1/views.py

# use isort to actually change the file(s)
$ isort enterprise/api/v1/views.py enterprise/api/v1/permissions.py
```

INTERNATIONALIZATION

All user-facing text content should be marked for translation. Even if this application is only run in English, our open source users may choose to use another language. Marking content for translation ensures our users have this choice.

Follow the [internationalization coding guidelines](#) in the edX Developer's Guide when developing new features.

6.1 Updating Translations

This project uses [Transifex](#) to translate content. After new features are developed the translation source files should be pushed to Transifex. Our translation community will translate the content, after which we can retrieve the translations.

Pushing source translation files to Transifex requires access to the edX-platform. Request access from the Open Source Team if you will be pushing translation files. You should also [configure the Transifex client](#) if you have not done so already.

The *make* targets listed below can be used to push or pull translations.

Target	Description
<code>pull_translations</code>	Pull translations from Transifex
<code>push_translations</code>	Push source translation files to Transifex

6.2 Fake Translations

As you develop features it may be helpful to know which strings have been marked for translation, and which are not. Use the *fake_translations* make target for this purpose. This target will extract all strings marked for translation, generate fake translations in the Esperanto (eo) language directory, and compile the translations.

You can trigger the display of the translations by setting your browser's language to Esperanto (eo), and navigating to a page on the site. Instead of plain English strings, you should see specially-accented English strings that look like this:

Thé Fütüre øf Ønliné Édüçätion # Før änyøné, änywhéré, änytimé #

SEGMENT EVENTS

Edx-enterprise emits several Segment events.

7.1 edx.bi.user.enterprise.onboarding

Emitted when a new `EnterpriseCourseEnrollment` record gets created.

The event contains these properties:

- **pathway**: A string identifying the mechanism through which the `EnterpriseCourseEnrollment` was created. See below for possible values.
- **course_run_id**: EdX course ID associated with the enrollment.
- **url_path**: The url path the user was visiting when the enterprise course enrollment got created. Can be null when the enrollment is created outside of a learner's request (for example admin or REST API enrollments).

The possible values of the **pathway** property are:

- **admin-enrollment**: Enrollment was created by an admin via the “Manage Learners” tool.
- **pending-admin-enrollment**: Enrollment was created from a pending enrollment record previously set up by an admin via the “Manage Learners” tool.
- **rest-api-enrollment**: Enrollment was created via the REST API.
- **data-consent-page-enrollment**: Enrollment was created when learner submitted the form on the data sharing consent page.
- **course-landing-page-enrollment**: Enrollment was created when learner enrolled via the course landing page.
- **direct-audit-enrollment**: Enrollment was created via the direct audit enrollment mechanism.
- **customer-admin-enrollment**: Enrollment was created by a customer admin via the frontend app admin portal.

7.2 edx.bi.user.enterprise.enrollment.course

Emitted when the enterprise app enrolls a user into a course.

The event contains these properties:

- **label**: EdX course ID associated with the enrollment.
- **enterprise_customer_uuid**: UUID of the `EnterpriseCustomer` associated with the enrollment.
- **enterprise_customer_name**: Name of the `EnterpriseCustomer` associated with the enrollment.

- **mode**: The mode of the enrollment (audit, verified, etc.).

7.3 `edx.bi.user.consent_form.shown`

Emitted when the learner is presented with the data sharing consent page.

The event contains these properties:

- **deferCreation**: True when creation of course enrollment is deferred.
- **successUrl**: URL to redirect the user to if consent is provided.
- **failureUrl**: URL to redirect the user to in case consent is not provided.
- **courseId**: EdX course ID of the associated course.
- **programId**: ID of the associated program.

7.4 `edx.bi.user.consent_form.accepted`

Emitted after the learner grants data sharing consent.

The event contains the same properties as the *edx.bi.user.consent_form.shown* event.

7.5 `edx.bi.user.consent_form.denied`

Emitted after the learner denies data sharing consent.

The event contains the same properties as the *edx.bi.user.consent_form.shown* event.

ENTERPRISE

8.1 enterprise package

8.1.1 Subpackages

`enterprise.admin` package

Submodules

`enterprise.admin.actions` module

Custom Django Admin actions used in enterprise app.

```
enterprise.admin.actions.export_as_csv_action(description='Export selected objects as CSV file',  
                                              fields=None, header=True)
```

Return an export csv action.

Parameters

- **description** (*string*) – action description
- **fields** (*[string]*) – list of model fields to include
- **header** (*bool*) – whether or not to output the column names as the first row

```
enterprise.admin.actions.refresh_catalog(self, request, queryset)
```

Kicks off background running tasks for refreshing catalogs

`enterprise.admin.forms` module

Forms to be used in the enterprise djangoapp.

```
class enterprise.admin.forms.AdminNotificationForm(data=None, files=None, auto_id='id_%s',  
                                                    prefix=None, initial=None, error_class=<class  
            'django.forms.utils.ErrorList'>,  
                                                    label_suffix=None, empty_permitted=False,  
                                                    instance=None, use_required_attribute=None,  
                                                    renderer=None)
```

Bases: `ModelForm`

Alternate form for AdminNotification.

```
class Meta
    Bases: object
    fields = '__all__'

    model
        alias of AdminNotification

    widgets = {'text': <django.forms.widgets.Textarea object>}

base_fields = {'admin_notification_filter':
<django.forms.models.ModelMultipleChoiceField object>, 'expiration_date':
<django.forms.fields.DateField object>, 'is_active':
<django.forms.fields.BooleanField object>, 'start_date':
<django.forms.fields.DateField object>, 'text': <django.forms.fields.CharField
object>, 'title': <django.forms.fields.CharField object>}
```

```
clean()
```

1. *start_date* and *expiration_date* are mandatory.
2. *start_date* must always come before the *expiration_date*.
3. There must be only one admin notification in a date range.

```
declared_fields = {}
```

```
property media
```

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.BulkCatalogQueryUpdateCommandConfigurationAdmin(model,
                                                                              admin_site)
```

Bases: *ConfigurationModelAdmin*

Admin form for the BulkCatalogQueryUpdateCommandConfiguration model

```
property media
```

```
class enterprise.admin.forms.EnterpriseCustomerAdminForm(data=None, files=None, auto_id='id_%s',
                                                         prefix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False, instance=None,
                                                         use_required_attribute=None,
                                                         renderer=None)
```

Bases: *ModelForm*

Alternate form for the EnterpriseCustomer admin page.

```
class Meta
```

Bases: *object*


```
fields = ('name', 'slug', 'auth_org_id', 'country', 'active', 'customer_type',
'site', 'enable_data_sharing_consent', 'enforce_data_sharing_consent',
'enable_audit_enrollment', 'enable_audit_data_reporting',
'replace_sensitive_sso_username', 'hide_course_original_price',
'enable_portal_code_management_screen',
'enable_portal_subscription_management_screen', 'enable_learner_portal',
'enable_learner_portal_offers',
'enable_portal_learner_credit_management_screen',
'enable_executive_education_2U_fulfillment', 'hide_labor_market_data',
'enable_integrated_customer_learner_portal_search',
'enable_career_engagement_network_on_learner_portal',
'career_engagement_network_message', 'enable_pathways', 'enable_programs',
'enable_demo_data_for_analytics_and_lpr', 'enable_analytics_screen',
'enable_portal_reporting_config_screen',
'enable_portal_saml_configuration_screen',
'enable_portal_lms_configurations_screen', 'enable_universal_link',
'enable_browse_and_request', 'enable_slug_login', 'contact_email',
'default_contract_discount', 'default_language', 'sender_alias', 'reply_to')
```

model

alias of [EnterpriseCustomer](#)

```
base_fields = {'active': <django.forms.fields.BooleanField object>, 'auth_org_id':
<django.forms.fields.CharField object>, 'career_engagement_network_message':
<django.forms.fields.CharField object>, 'contact_email':
<django.forms.fields.EmailField object>, 'country':
<django_countries.fields.LazyTypedChoiceField object>, 'customer_type':
<django.forms.models.ModelChoiceField object>, 'default_contract_discount':
<django.forms.fields.DecimalField object>, 'default_language':
<django.forms.fields.TypedChoiceField object>, 'enable_analytics_screen':
<django.forms.fields.BooleanField object>, 'enable_audit_data_reporting':
<django.forms.fields.BooleanField object>, 'enable_audit_enrollment':
<django.forms.fields.BooleanField object>, 'enable_browse_and_request':
<django.forms.fields.BooleanField object>,
'enable_career_engagement_network_on_learner_portal':
<django.forms.fields.BooleanField object>, 'enable_data_sharing_consent':
<django.forms.fields.BooleanField object>, 'enable_demo_data_for_analytics_and_lpr':
<django.forms.fields.BooleanField object>,
'enable_executive_education_2U_fulfillment': <django.forms.fields.BooleanField
object>, 'enable_integrated_customer_learner_portal_search':
<django.forms.fields.BooleanField object>, 'enable_learner_portal':
<django.forms.fields.BooleanField object>, 'enable_learner_portal_offers':
<django.forms.fields.BooleanField object>, 'enable_pathways':
<django.forms.fields.BooleanField object>, 'enable_portal_code_management_screen':
<django.forms.fields.BooleanField object>,
'enable_portal_learner_credit_management_screen': <django.forms.fields.BooleanField
object>, 'enable_portal_lms_configurations_screen':
<django.forms.fields.BooleanField object>, 'enable_portal_reporting_config_screen':
<django.forms.fields.BooleanField object>,
'enable_portal_saml_configuration_screen': <django.forms.fields.BooleanField
object>, 'enable_portal_subscription_management_screen':
<django.forms.fields.BooleanField object>, 'enable_programs':
<django.forms.fields.BooleanField object>, 'enable_slug_login':
<django.forms.fields.BooleanField object>, 'enable_universal_link':
<django.forms.fields.BooleanField object>, 'enforce_data_sharing_consent':
<django.forms.fields.TypedChoiceField object>, 'hide_course_original_price':
<django.forms.fields.BooleanField object>, 'hide_labor_market_data':
<django.forms.fields.BooleanField object>, 'name': <django.forms.fields.CharField
object>, 'replace_sensitive_sso_username': <django.forms.fields.BooleanField
object>, 'reply_to': <django.forms.fields.EmailField object>, 'sender_alias':
<django.forms.fields.CharField object>, 'site':
<django.forms.models.ModelChoiceField object>, 'slug':
<django.forms.fields.SlugField object>}
```

```
declared_fields = {}
```

property media

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm(data=None, files=None,
                                                                auto_id='id_%s', prefix=None,
                                                                initial=None,
                                                                error_class=<class
                                                                'django.forms.utils.ErrorList'>,
                                                                label_suffix=None,
                                                                empty_permitted=False,
                                                                instance=None,
                                                                use_required_attribute=None,
                                                                renderer=None)
```

Bases: `ModelForm`

form for `EnterpriseCustomerCatalogAdmin` class.

class Meta

Bases: `object`

fields = `'__all__'`

model

alias of `EnterpriseCustomerCatalog`

```
base_fields = {'content_filter': <jsonfield.forms.JSONField object>,
               'enabled_course_modes': <jsonfield.forms.JSONField object>,
               'enterprise_catalog_query': <django.forms.models.ModelChoiceField object>,
               'enterprise_customer': <django.forms.models.ModelChoiceField object>,
               'preview_button': <django.forms.fields.Field object>,
               'publish_audit_enrollment_urls': <django.forms.fields.BooleanField object>,
               'title': <django.forms.fields.CharField object>}
```

```
declared_fields = {'preview_button': <django.forms.fields.Field object>}
```

```
static get_catalog_preview_uuid(post_data)
```

Return the uuid of the catalog the preview button was clicked on There must be only one preview button in the POST data.

e.g: `'enterprise_customer_catalogs-0-preview_button'`

property media

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm(*args, **kwargs)
```

Bases: `ModelForm`

Alternate form for the `EnterpriseCustomerIdentityProvider` admin page.

This form fetches identity providers from `lms third_party_auth` app. If `third_party_auth` app is not available it displays `provider_id` as a `CharField`.

class Meta

Bases: `object`

fields = `'__all__'`

model

alias of `EnterpriseCustomerIdentityProvider`

```
base_fields = {'default_provider': <django.forms.fields.BooleanField object>,
               'enterprise_customer': <django.forms.models.ModelChoiceField object>,
               'provider_id': <django.forms.fields.SlugField object>}
```

clean()

Final validations of model fields.

1. Validate that selected site for enterprise customer matches with the selected identity provider's site.

declared_fields = {}

property media

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.EnterpriseCustomerReportingConfigAdminForm(data=None,
                                                                       files=None,
                                                                       auto_id='id_%s',
                                                                       prefix=None,
                                                                       initial=None,
                                                                       error_class=<class
                                                                       'django.forms.utils.ErrorList'>,
                                                                       label_suffix=None,
                                                                       empty_permitted=False,
                                                                       instance=None,
                                                                       use_required_attribute=None,
                                                                       renderer=None)
```

Bases: `ModelForm`

Alternate form for the EnterpriseCustomerReportingConfiguration admin page.

This form uses the PasswordInput widget to obscure passwords as they are being entered by the user.

class Meta

Bases: `object`

```
fields = ('enterprise_customer', 'active', 'data_type', 'report_type',
          'delivery_method', 'enable_compression', 'pgp_encryption_key', 'frequency',
          'day_of_month', 'day_of_week', 'hour_of_day', 'include_date', 'email',
          'decrypted_password', 'sftp_hostname', 'sftp_port', 'sftp_username',
          'decrypted_sftp_password', 'sftp_file_path', 'enterprise_customer_catalogs')
```

model

alias of `EnterpriseCustomerReportingConfiguration`

```
widgets = {'decrypted_password': <django.forms.widgets.PasswordInput object>,
           'decrypted_sftp_password': <django.forms.widgets.PasswordInput object>}
```

```

base_fields = {'active': <django.forms.fields.BooleanField object>, 'data_type':
<django.forms.fields.TypedChoiceField object>, 'day_of_month':
<django.forms.fields.IntegerField object>, 'day_of_week':
<django.forms.fields.TypedChoiceField object>, 'decrypted_password':
<django.forms.fields.CharField object>, 'decrypted_sftp_password':
<django.forms.fields.CharField object>, 'delivery_method':
<django.forms.fields.TypedChoiceField object>, 'email':
<multi_email_field.forms.MultiEmailField object>, 'enable_compression':
<django.forms.fields.BooleanField object>, 'enterprise_customer':
<django.forms.models.ModelChoiceField object>, 'enterprise_customer_catalogs':
<django.forms.models.ModelMultipleChoiceField object>, 'frequency':
<django.forms.fields.TypedChoiceField object>, 'hour_of_day':
<django.forms.fields.IntegerField object>, 'include_date':
<django.forms.fields.BooleanField object>, 'pgp_encryption_key':
<django.forms.fields.CharField object>, 'report_type':
<django.forms.fields.TypedChoiceField object>, 'sftp_file_path':
<django.forms.fields.CharField object>, 'sftp_hostname':
<django.forms.fields.CharField object>, 'sftp_port':
<django.forms.fields.IntegerField object>, 'sftp_username':
<django.forms.fields.CharField object>}

```

clean()

Override of clean method to perform additional validation

```

declared_fields = {'enterprise_customer_catalogs':
<django.forms.models.ModelMultipleChoiceField object>}

```

property media

Return all media required to render the widgets on this form.

```

class enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentForm(*args, **kwargs)

```

Bases: `UserRoleAssignmentAdminForm`

Form for EnterpriseFeatureUserRoleAssignments.

class Meta

Bases: `object`

fields = ['user', 'role']

model

alias of `EnterpriseFeatureUserRoleAssignment`

```

base_fields = {'role': <django.forms.models.ModelChoiceField object>, 'user':
<edx_rbac.fields.UserFromEmailField object>}

```

```

declared_fields = {'user': <edx_rbac.fields.UserFromEmailField object>}

```

property media

Return all media required to render the widgets on this form.

```

class enterprise.admin.forms.ManageLearnersDataSharingConsentForm(*args, **kwargs)

```

Bases: `Form`

Form to request DSC from a learner.

```
class Fields
    Bases: object
    Namespace class for field names.

    COURSE = 'course'

    EMAIL_OR_USERNAME = 'email_or_username'

base_fields = {'course': <django.forms.fields.CharField object>,
               'email_or_username': <django.forms.fields.CharField object>}

clean_course()
    Verify course ID has an associated course in LMS.

clean_email_or_username()
    Verify email_or_username has associated user in our database.

declared_fields = {'course': <django.forms.fields.CharField object>,
                   'email_or_username': <django.forms.fields.CharField object>}

is_course_in_catalog(course_id)
    Check whether course exists in enterprise customer catalog.

is_user_linked(email)
    Check whether user is linked to the enterprise customer or not.

property media
    Return all media required to render the widgets on this form.

class enterprise.admin.forms.ManageLearnersForm(*args, **kwargs)
    Bases: Form
    Form to manage learner additions.

class CsvColumns
    Bases: object
    Namespace class for CSV column names.

    COURSE_ID = 'course_id'

    EMAIL = 'email'

class Fields
    Bases: object
    Namespace class for field names.

    BULK_UPLOAD = 'bulk_upload_csv'

    COURSE = 'course'

    COURSE_MODE = 'course_mode'

    DISCOUNT = 'discount'

    EMAIL_OR_USERNAME = 'email_or_username'

    GENERAL_ERRORS = '__all__'
```

```

MODE = 'mode'

NOTIFY = 'notify_on_enrollment'

REASON = 'reason'

SALES_FORCE_ID = 'sales_force_id'

class Modes
    Bases: object
    Namespace class for form modes.

    MODE_BULK = 'bulk'

    MODE_SINGULAR = 'singular'

class NotificationTypes
    Bases: object
    Namespace class for notification types

    BY_EMAIL = 'by_email'

    DEFAULT = 'by_email'

    NO_NOTIFICATION = 'do_not_notify'

base_fields = {'bulk_upload_csv': <django.forms.fields.FileField object>, 'course':
<django.forms.fields.CharField object>, 'course_mode':
<django.forms.fields.ChoiceField object>, 'discount':
<django.forms.fields.DecimalField object>, 'email_or_username':
<django.forms.fields.CharField object>, 'notify_on_enrollment':
<django.forms.fields.ChoiceField object>, 'reason': <django.forms.fields.CharField
object>, 'sales_force_id': <django.forms.fields.CharField object>}

clean()
    Clean fields that depend on each other.

    In this case, the form can be used to link single user or bulk link multiple users. These are mutually exclusive
    modes, so this method checks that only one field is passed.

clean_course()
    Verify course ID and retrieve course details.

clean_discount()
    Verify that discount value should be from 0 to 100.

clean_email_or_username()
    Clean email form field

    Returns
        the cleaned value, converted to an email address (or an empty string)

    Return type
        str

clean_notify()
    Clean the notify_on_enrollment field.

```

clean_reason()

Clean the reason for enrollment field

```
declared_fields = {'bulk_upload_csv': <django.forms.fields.FileField object>,
'course': <django.forms.fields.CharField object>, 'course_mode':
<django.forms.fields.ChoiceField object>, 'discount':
<django.forms.fields.DecimalField object>, 'email_or_username':
<django.forms.fields.CharField object>, 'notify_on_enrollment':
<django.forms.fields.ChoiceField object>, 'reason': <django.forms.fields.CharField
object>, 'sales_force_id': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentForm(*args, **kwargs)
```

Bases: `UserRoleAssignmentAdminForm`

Form for SystemWideEnterpriseUserRoleAssignments.

class Meta

Bases: `object`

```
fields = ['user', 'role', 'enterprise_customer', 'applies_to_all_contexts']
```

model

alias of `SystemWideEnterpriseUserRoleAssignment`

```
base_fields = {'applies_to_all_contexts': <django.forms.fields.BooleanField
object>, 'enterprise_customer': <django.forms.models.ModelChoiceField object>,
'role': <django.forms.models.ModelChoiceField object>, 'user':
<edx_rbac.fields.UserFromEmailField object>}
```

clean()

Ensure that the form's enterprise customer value defaults to the first active, linked enterprise customer by the provided user

```
declared_fields = {'user': <edx_rbac.fields.UserFromEmailField object>}
```

property media

Return all media required to render the widgets on this form.

```
class enterprise.admin.forms.TransmitEnterpriseCoursesForm(data=None, files=None,
auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute=None,
renderer=None)
```

Bases: `Form`

Form to transmit courses metadata for enterprise customers.

```
base_fields = {'channel_worker_username': <django.forms.fields.CharField object>}
```


clean_channel_worker_username()

Clean enterprise channel worker user form field

Returns

the cleaned value of channel user username for transmitting courses metadata.

Return type

str

```
declared_fields = {'channel_worker_username': <django.forms.fields.CharField
object>}
```

property media

Return all media required to render the widgets on this form.

enterprise.admin.paginator module

Custom paginator to implement smart pagination.

```
class enterprise.admin.paginator.CustomPaginator(object_list, per_page, orphans=0,
allow_empty_first_page=True)
```

Bases: `Paginator`

Adopted from django/core/paginator so as to implement smart links pagination in custom views.

property page_range

We have customized the getter so that it can return the value of the page_range property instead of always calculating the result.

enterprise.admin.utils module

Admin utilities.

```
class enterprise.admin.utils.UrlNames
```

Bases: `object`

Collection on URL names used in admin

```
MANAGE_LEARNERS = 'enterprise_manage_learners'
```

```
MANAGE_LEARNERS_DSC = 'enterprise_manage_learners_data_sharing_consent'
```

```
PREVIEW_EMAIL_TEMPLATE = 'enterprise_preview_email_template'
```

```
PREVIEW_QUERY_RESULT = 'enterprise_preview_query_result'
```

```
SETUP_AUTH_ORG_ID = 'enterprise_setup_auth_org_id'
```

```
TRANSMIT_COURSES_METADATA = 'enterprise_transmit_courses_metadata'
```

```
URL_PREFIX = 'enterprise_'
```

```
enterprise.admin.utils.email_or_username__to__email(email_or_username)
```

Convert email_or_username to email.

Returns

If *email_or_username* was a username returns user's email, otherwise assumes it was an email and returns as is.

Return type

`str`

`enterprise.admin.utils.paginated_list(object_list, page, page_size=25)`

Returns paginated list.

Parameters

- **object_list** (`QuerySet`) – A list of records to be paginated.
- **page** (`int`) – Current page number.
- **page_size** (`int`) – Number of records displayed in each paginated set.
- **show_all** (`bool`) – Whether to show all records.

Adopted from `django/contrib/admin/templatetags/admin_list.py` https://github.com/django/django/blob/1.11.1/django/contrib/admin/templatetags/admin_list.py#L50

`enterprise.admin.utils.parse_csv(file_stream, expected_columns=None)`

Parse csv file and return a stream of dictionaries representing each row.

First line of CSV file must contain column headers.

Parameters

- **file_stream** – input file
- **expected_columns** (`set[unicode]`) – columns that are expected to be present

Yields

dict – CSV line parsed into a dictionary.

`enterprise.admin.utils.split_usernames_and_emails(email_field)`

Split the contents of the email field into a list.

In some cases, a user could enter a comma-separated value inline in the Manage Learners form. We should check to see if that's the case, and provide a list of email addresses or usernames if it is.

`enterprise.admin.utils.validate_csv(file_stream, expected_columns=None)`

Validate csv file for encoding and expected header.

Parameters

- **file_stream** – input file
- **expected_columns** – list of column names that are expected to be present in csv

Returns

an iterable for csv data if csv passes the validation

Return type

`reader`

Raises

ValidationError –

enterprise.admin.views module

Custom Django Admin views used in enterprise app.

```
class enterprise.admin.views.BaseEnterpriseCustomerView(**kwargs)
```

Bases: `View`

Base class for Enterprise Customer views.

```
get_form_view(request, customer_uuid, additional_context=None)
```

render the form with appropriate context.

```
template = None
```

```
class enterprise.admin.views.CatalogQueryPreviewView(**kwargs)
```

Bases: `APIView`

Renders a search/all response from Discovery for a particular catalog query.

```
get(request, catalog_query_id)
```

Render the response for a particular catalog query.

```
static get_response_from_discovery(request, catalog_query)
```

POST a query to Discovery's search/all endpoint and return the results.

```
class enterprise.admin.views.EnterpriseCustomerManageLearnerDataSharingConsentView(**kwargs)
```

Bases: `BaseEnterpriseCustomerView`

Manage Learners Data Sharing Consent View.

Allows to request the DSC from a learner for a specific course.

```
class ContextParameters
```

Bases: `object`

Namespace-style class for custom context parameters.

```
ENTERPRISE_CUSTOMER = 'enterprise_customer'
```

```
MANAGE_LEARNERS_DSC_FORM = 'manage_learners_data_sharing_consent_form'
```

```
get(request, customer_uuid)
```

Handle GET request - render "Request a DSC from Learner" form.

Parameters

- **request** (`django.http.request.HttpRequest`) – Request instance
- **customer_uuid** (`str`) – Enterprise Customer UUID

Returns

HttpResponse

Return type

`django.http.response.HttpResponse`

```
post(request, customer_uuid)
```

Handle POST request - handle form submissions.

Parameters

- **request** (`django.http.request.HttpRequest`) – Request instance

- **customer_uuid** (*str*) – Enterprise Customer UUID

```
template = 'enterprise/admin/clear_learners_data_sharing_consent.html'
```

```
class enterprise.admin.views.EnterpriseCustomerManageLearnersView(**kwargs)
```

Bases: *BaseEnterpriseCustomerView*

Manage Learners view.

Lists learners linked to chosen Enterprise Customer and allows adding and deleting them.

```
class ContextParameters
```

Bases: *object*

Namespace-style class for custom context parameters.

```
ENROLLMENT_URL = 'ENROLLMENT_API_ROOT_URL'
```

```
ENTERPRISE_CUSTOMER = 'enterprise_customer'
```

```
LEARNERS = 'learners'
```

```
MANAGE_LEARNERS_FORM = 'manage_learners_form'
```

```
PENDING_LEARNERS = 'pending_learners'
```

```
SEARCH_KEYWORD = 'search_keyword'
```

```
delete(request, customer_uuid)
```

Handle DELETE request - handle unlinking learner.

Parameters

- **request** (*django.http.request.HttpRequest*) – Request instance
- **customer_uuid** (*str*) – Enterprise Customer UUID

Returns

HttpResponse

Return type

django.http.response.HttpResponse

```
get(request, customer_uuid)
```

Handle GET request - render linked learners list and “Link learner” form.

Parameters

- **request** (*django.http.request.HttpRequest*) – Request instance
- **customer_uuid** (*str*) – Enterprise Customer UUID

Returns

HttpResponse

Return type

django.http.response.HttpResponse

```
get_enterprise_customer_user_queryset(request, search_keyword, customer_uuid, page_size=25)
```

Get the list of EnterpriseCustomerUsers we want to render. Note that, if no **search_keyword** is provided, the resulting queryset will only include up to some constant limit of results, so that we can support searching for particular user records in customers with very large numbers of learners.

Parameters

- **request** (*HttpRequest*) – HTTP Request instance.
- **search_keyword** (*str*) – The keyword to search for in users' email addresses and user-names.
- **customer_uuid** (*str*) – A unique identifier to filter down to only users linked to a
- **EnterpriseCustomer.** (*particular*) –
- **page_size** (*int*) – Number of learners displayed in each paginated set.

classmethod `get_failed_enrollment_message(users, enrolled_in)`

Create message for the users who were not able to be enrolled in a course.

Parameters

- **users** – An iterable of users who were not successfully enrolled
- **enrolled_in** (*str*) – A string identifier for the course with which enrollment was attempted

Returns: tuple: A 2-tuple containing a message type and message text

classmethod `get_pending_enrollment_message(pending_users, enrolled_in)`

Create message for the users who were enrolled in a course.

Parameters

- **users** – An iterable of PendingEnterpriseCustomerUsers who were successfully linked with a pending enrollment
- **enrolled_in** (*str*) – A string identifier for the course the pending users were linked to

Returns

A 2-tuple containing a message type and message text

Return type

tuple

get_pending_users_queryset(*search_keyword, customer_uuid*)

Get the list of PendingEnterpriseCustomerUsers we want to render.

Parameters

- **search_keyword** (*str*) – The keyword to search for in pending users' email addresses.
- **customer_uuid** (*str*) – A unique identifier to filter down to only pending users
- **EnterpriseCustomer.** (*linked to a particular*) –

get_search_keyword(*request*)

Retrieve the search querystring from the GET parameters.

classmethod `get_success_enrollment_message(users, enrolled_in)`

Create message for the users who were enrolled in a course.

Parameters

- **users** – An iterable of users who were successfully enrolled
- **enrolled_in** (*str*) – A string identifier for the course the users were enrolled in

Returns

A 2-tuple containing a message type and message text

Return type`tuple`**post**(*request*, *customer_uuid*)

Handle POST request - handle form submissions.

Parameters

- **request** (*django.http.request.HttpRequest*) – Request instance
- **customer_uuid** (*str*) – Enterprise Customer UUID

Returns`HttpResponse`**Return type**`django.http.response.HttpResponse`**classmethod** **send_messages**(*http_request*, *message_requests*)

Deduplicate any outgoing message requests, and send the remainder.

Parameters

- **http_request** – The HTTP request in whose response we want to embed the messages
- **message_requests** – A list of unduplicated messages in the form of tuples of message type and text- for example, ('error', 'Something went wrong')

template = 'enterprise/admin/manage_learners.html'**class** `enterprise.admin.views.EnterpriseCustomerSetupAuthOrgIDView`(***kwargs*)Bases: `BaseEnterpriseCustomerView`

Setup Auth org id View.

This action will configure SSO to GetSmarter using edX credentials via Auth0.

get(*request*, *customer_uuid*)

Handle GET request - render "Setup Auth org id" form.

Parameters

- **request** (*django.http.request.HttpRequest*) – Request instance
- **customer_uuid** (*str*) – Enterprise Customer UUID

Returns`HttpResponse`**Return type**`django.http.response.HttpResponse`**post**(*request*, *customer_uuid*)

Handle POST request - handle form submissions.

Parameters

- **request** (*django.http.request.HttpRequest*) – Request instance
- **customer_uuid** (*str*) – Enterprise Customer UUID

template = 'enterprise/admin/setup_auth_org_id.html'

```
class enterprise.admin.views.EnterpriseCustomerTransmitCoursesView(**kwargs)
```

Bases: [BaseEnterpriseCustomerView](#)

Transmit courses view.

Allows transmitting of courses metadata for provided enterprise.

```
class ContextParameters
```

Bases: [object](#)

Namespace-style class for custom context parameters.

```
ENTERPRISE_CUSTOMER = 'enterprise_customer'
```

```
TRANSMIT_COURSES_METADATA_FORM = 'transmit_courses_metadata_form'
```

```
get(request, customer_uuid)
```

Handle GET request - render “Transmit courses metadata” form.

Parameters

- **request** ([django.http.request.HttpRequest](#)) – Request instance
- **customer_uuid** ([str](#)) – Enterprise Customer UUID

Returns

[HttpResponse](#)

Return type

[django.http.response.HttpResponse](#)

```
post(request, customer_uuid)
```

Handle POST request - handle form submissions.

Parameters

- **request** ([django.http.request.HttpRequest](#)) – Request instance
- **customer_uuid** ([str](#)) – Enterprise Customer UUID

```
template = 'enterprise/admin/transmit_courses_metadata.html'
```

```
class enterprise.admin.views.TemplatePreviewView(**kwargs)
```

Bases: [View](#)

Renders a given NotificationTemplate object to HTML for online viewing.

```
get(request, template_id, view_type)
```

Render the given template with the stock data.

```
static get_user_name(request)
```

Get a human-readable name for the user.

```
view_type_contexts = {'course': {'enrolled_in': {'name': 'OpenEdX Demo Course',
'start': datetime.datetime(2016, 1, 1, 0, 0), 'type': 'course', 'url':
'http://example.com/courses/edx-demo-course'}, 'organization_name': 'OpenEdX'},
'program': {'enrolled_in': {'branding': 'MicroMasters', 'name': 'OpenEdX Demo
Program', 'start': datetime.datetime(2016, 1, 1, 0, 0), 'type': 'program', 'url':
'http://example.com/programs/edx-demo-program'}, 'organization_name': 'OpenEdX'}}
```

enterprise.admin.widgets module

Widgets to be used in the enterprise djangoapp.

```
class enterprise.admin.widgets.SubmitInput(attrs=None)
    Bases: Input
    Widget for input type field
    input_type = 'submit'
    property media
    template_name = 'django/forms/widgets/text.html'
```

Module contents

Django admin integration for enterprise app.

```
class enterprise.admin.AdminNotificationAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin for AdminNotification model.
    filter_horizontal = ('admin_notification_filter',)
    form
        alias of AdminNotificationForm
    list_display = ('id', 'title', 'text', 'is_active', 'start_date', 'expiration_date',
    'created', 'modified')
    property media
    model
        alias of AdminNotification

class enterprise.admin.AdminNotificationFilterAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin for models.AdminNotificationFilter model.
    list_display = ('id', 'filter', 'created', 'modified')
    property media
    model
        alias of AdminNotificationFilter

class enterprise.admin.AdminNotificationReadAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin for AdminNotificationRead model.
    list_display = ('id', 'enterprise_customer_user', 'admin_notification', 'is_read',
    'created', 'modified')
    property media
```


modelalias of [AdminNotificationRead](#)

```
class enterprise.admin.BigTableMysqlPaginator(object_list, per_page, orphans=0,
                                             allow_empty_first_page=True)
```

Bases: [Paginator](#)

A paginator that uses INFORMATION_SCHEMA.TABLES to estimate the total number of rows in a table.

ARBITRARILY_LARGE_NUMBER = 100000000**property count**

Returns the number of items in the object list (possibly an estimate).

```
class enterprise.admin.ChatGPTResponseAdmin(model, admin_site)
```

Bases: [ModelAdmin](#)

Django admin for ChatGPTResponse model.

```
list_display = ('uuid', 'prompt_type', 'enterprise_customer', 'prompt_hash',
               'created')
```

```
list_filter = ('prompt_type',)
```

property media**model**alias of [ChatGPTResponse](#)

```
readonly_fields = ('prompt_type', 'prompt', 'response', 'prompt_hash', 'created',
                  'modified')
```

```
class enterprise.admin.EnrollmentNotificationEmailTemplateAdmin(model, admin_site)
```

Bases: [DjangoObjectActions](#), [ModelAdmin](#)

Django admin for EnrollmentNotificationEmailTemplate model

class MetaBases: [object](#)**model**alias of [EnrollmentNotificationEmailTemplate](#)

```
change_actions = ('preview_as_course', 'preview_as_program')
```

get_urls()

Returns the additional urls used by the custom object tools.

property media

```
preview(obj, preview_type)
```

Object tool handler method - redirects to “Preview” view

```
preview_as_course(request, obj)
```

Redirect to preview the HTML template in the context of a course.

```
preview_as_program(request, obj)
```

Redirect to preview the HTML template in the context of a program.

```
class enterprise.admin.EnterpriseCatalogQueryAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for EnterpriseCatalogQuery.
    class Meta
        Bases: object
        model
            alias of EnterpriseCatalogQuery
    discovery_query_url(obj)
        Return discovery url for preview.
    fields = ('uuid', 'title', 'discovery_query_url', 'content_filter')
    get_urls()
        Returns the additional urls used by the custom object tools.
    has_delete_permission(request, obj=None)
        Return True if the given request has permission to delete the given Django model instance, the default implementation doesn't examine the obj parameter.
        Can be overridden by the user in subclasses. In such case it should return True if the given request has permission to delete the obj model instance. If obj is None, this should return True if the given request has permission to delete any object of the given type.
    list_display = ('title', 'discovery_query_url')
    property media
    readonly_fields = ('discovery_query_url', 'uuid')

class enterprise.admin.EnterpriseCourseEnrollmentAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for EnterpriseCourseEnrollment
    class Meta
        Bases: object
        model
            alias of EnterpriseCourseEnrollment
    change_list_template = 'enterprise/admin/enterprise_course_enrollments_list.html'
    changelist_view(request, extra_context=None)
        Override to conditionally show the button.
    get_urls()
        Append Enrollment Attribute Override view url with default urls
    has_add_permission(request)
        Disable add permission for EnterpriseCourseEnrollment.
    has_delete_permission(request, obj=None)
        Disable deletion for EnterpriseCourseEnrollment.
    license_uuid(obj)
        Return the subscription license UUID (if any exists) associated with this enrollment.
```

```
list_display = ('enterprise_customer_user', 'course_id', 'saved_for_later',
               'unenrolled_at')

property media

readonly_fields = ('enterprise_customer_user', 'course_id', 'saved_for_later',
                  'license_uuid')

search_fields = ('enterprise_customer_user__user_id', 'course_id')

class enterprise.admin.EnterpriseCustomerAdmin(model, admin_site)
    Bases: DjangoObjectActions, SimpleHistoryAdmin
    Django admin model for EnterpriseCustomer.
    EXPORT_AS_CSV_FIELDS = ['name', 'active', 'site', 'uuid', 'identity_provider']

    class Meta
        Bases: object
        model
            alias of EnterpriseCustomer

    actions = [<function export_as_csv_action.<locals>.export_as_csv>]

    change_actions = ('setup_auth_org_id', 'manage_learners',
                     'manage_learners_data_sharing_consent', 'transmit_courses_metadata')

    change_view(request, object_id, form_url="", extra_context=None)

    enable_dsc(instance)
        Return True if data sharing consent is enabled for EnterpriseCustomer.

        Parameters
            instance (enterprise.models.EnterpriseCustomer) – EnterpriseCustomer model instance
```

```
fieldsets = (('Enterprise info', {'fields': ('name', 'active', 'slug',
'auth_org_id', 'country')}), ('Subsidy management screens ', {'fields':
('enable_portal_learner_credit_management_screen',
'enable_portal_subscription_management_screen',
'enable_portal_code_management_screen'), 'description': "Select the check boxes
below to enable specific subsidy management screens on the organization's
administrator portal. If an option is left unchecked, the customer administrator
will not see the screen in their portal and will not be able to apply the associated
configurations via self-service."}), ('Subsidy settings', {'fields':
('enable_browse_and_request', 'enable_universal_link'), 'description': 'Select the
check boxes below to enable specific subsidy management settings for the
administrator portal for subscription and codes customers. These should not be
selected for customers that only have learner credit.'}), ('Data sharing consent',
{'fields': ('enable_data_sharing_consent', 'enforce_data_sharing_consent')}),
('Email and language ', {'fields': ('contact_email', 'reply_to', 'sender_alias',
'default_language', 'hide_labor_market_data')}), ('Reporting', {'fields':
('enable_portal_reporting_config_screen',)}), ('Integration and learning platform
settings', {'fields': ('enable_portal_lms_configurations_screen',
'enable_portal_saml_configuration_screen', 'enable_slug_login',
'replace_sensitive_sso_username', 'hide_course_original_price',
'enable_generation_of_api_credentials')}), ('Recommended default settings for all
enterprise customers', {'fields': ('site', 'customer_type',
'enable_learner_portal', 'enable_integrated_customer_learner_portal_search',
'enable_analytics_screen', 'enable_audit_enrollment', 'enable_audit_data_reporting',
'enable_learner_portal_offers', 'enable_executive_education_2U_fulfillment',
'enable_career_engagement_network_on_learner_portal',
'career_engagement_network_message', 'enable_pathways', 'enable_programs',
'enable_demo_data_for_analytics_and_lpr', 'enable_academies', 'enable_one_academy'),
'description': 'The following default settings should be the same for the majority
of enterprise customers, and are either rarely used, unlikely to be sold, or
unlikely to be changed from the default.'}))
```

form

alias of [EnterpriseCustomerAdminForm](#)

get_change_actions(*args, **kwargs)

Buttons that appear at the top of the “Change Enterprise Customer” page.

Due to a known deficiency in the upstream `django_object_actions` library, we must STILL define `change_actions` above with all possible values.

get_form(request, obj=None, change=False, **kwargs)

Retrieve the appropriate form to use, saving the request user into the form for use in loading catalog details

get_search_results(request, queryset, search_term)

Return a tuple containing a queryset to implement the search and a boolean indicating if the results may contain duplicates.

get_urls()

Returns the additional urls used by the custom object tools.

has_identity_provider(instance)

Return True if EnterpriseCustomer has related identity provider.

Parameters

instance ([enterprise.models.EnterpriseCustomer](#)) – *EnterpriseCustomer* model instance

has_logo(*instance*)

Return True if EnterpriseCustomer has a logo.

Parameters

instance (`enterprise.models.EnterpriseCustomer`) – *EnterpriseCustomer* model instance

```
inlines = [<class 'enterprise.admin.EnterpriseCustomerBrandingConfigurationInline'>,
<class 'enterprise.admin.EnterpriseCustomerIdentityProviderInline'>, <class
'enterprise.admin.EnterpriseCustomerCatalogInline'>, <class
'enterprise.admin.PendingEnterpriseCustomerAdminUserInline'>]
```

```
list_display = ('name', 'slug', 'customer_type', 'site', 'country', 'active',
'has_logo', 'enable_dsc', 'has_identity_provider', 'uuid')
```

```
list_filter = ('active',)
```

manage_learners(*request, obj*)

Object tool handler method - redirects to “Manage Learners” view

manage_learners_data_sharing_consent(*request, obj*)

Object tool handler method - redirects to “Clear Learners Data Sharing Consent” view

property media

```
ordering = ('name',)
```

```
search_fields = ('name', 'uuid')
```

setup_auth_org_id(*request, obj*)

Object tool handler method - redirects to *Setup Auth org id* view.

transmit_courses_metadata(*request, obj*)

Object tool handler method - redirects to *Transmit Courses Metadata* view.

class `enterprise.admin.EnterpriseCustomerBrandingConfigurationInline`(*parent_model*,
admin_site)

Bases: `StackedInline`

Django admin model for EnterpriseCustomerBrandingConfiguration.

The admin interface has the ability to edit models on the same page as a parent model. These are called inlines. <https://docs.djangoproject.com/en/1.8/ref/contrib/admin/#django.contrib.admin.StackedInline>

```
can_delete = False
```

property media

model

alias of `EnterpriseCustomerBrandingConfiguration`

class `enterprise.admin.EnterpriseCustomerCatalogAdmin`(*model, admin_site*)

Bases: `ModelAdmin`

Django admin model for EnterpriseCustomerCatalog.

class `Media`

Bases: `object`

```
js = ('enterprise/admin/enterprise_customer_catalog.js',)

class Meta
    Bases: object

    model
        alias of EnterpriseCustomerCatalog

    actions = [<function refresh_catalog>]

    autocomplete_fields = ['enterprise_customer']

    fields = ('title', 'enterprise_customer', 'enterprise_catalog_query',
        'content_filter', 'enabled_course_modes', 'publish_audit_enrollment_urls')

    get_actions(request)
        Disallow the delete selected action as that does not send a DELETE request to enterprise-catalog

    get_form(request, obj=None, change=False, **kwargs)
        Return a Form class for use in the admin add view. This is used by add_view and change_view.

    list_display = ('uuid_nowrap', 'enterprise_customer', 'title',
        'preview_catalog_url')

    property media

    ordering = ('enterprise_customer__name', 'title')

    preview_catalog_url(obj)
        Return enterprise catalog url for preview.

    readonly_fields = ('preview_catalog_url',)

    search_fields = ('uuid', 'title', 'enterprise_customer__name',
        'enterprise_customer__uuid')

    uuid_nowrap(obj)
        Inject html for disabling wrap for uuid

class enterprise.admin.EnterpriseCustomerCatalogInline(parent_model, admin_site)
    Bases: TabularInline

    Django admin model for EnterpriseCustomerCatalog. The admin interface has the ability to edit models on the
    same page as a parent model. These are called inlines. https://docs.djangoproject.com/en/1.8/ref/contrib/admin/#django.contrib.admin.StackedInline

    can_delete = False

    extra = 0

    form
        alias of EnterpriseCustomerCatalogAdminForm

    get_formset(request, obj=None, **kwargs)
        Return a BaseInlineFormSet class for use in admin add/change views.

    property media
```

modelalias of *EnterpriseCustomerCatalog***class** `enterprise.admin.EnterpriseCustomerIdentityProviderInline`(*parent_model*, *admin_site*)Bases: `StackedInline`Django admin model for `EnterpriseCustomerIdentityProvider`.

The admin interface has the ability to edit models on the same page as a parent model. These are called inlines. <https://docs.djangoproject.com/en/1.8/ref/contrib/admin/#django.contrib.admin.StackedInline>

extra = 0**form**alias of *EnterpriseCustomerIdentityProviderAdminForm***property media****model**alias of *EnterpriseCustomerIdentityProvider***class** `enterprise.admin.EnterpriseCustomerInviteKeyAdmin`(*model*, *admin_site*)Bases: `ModelAdmin`Django admin model for `EnterpriseCustomerInviteKey`.**class Meta**Bases: `object`**model**alias of *EnterpriseCustomerInviteKey***fields** = ('enterprise_customer', 'usage_count', 'usage_limit', 'expiration_date', 'is_active')**get_readonly_fields**(*request*, *obj=None*)

Hook for specifying custom readonly fields.

list_display = ('uuid', 'enterprise_customer_id', 'usage_limit', 'expiration_date', 'is_active')**list_filter** = ('is_active',)**property media****readonly_fields** = ('uuid', 'usage_count')**search_fields** = ('uuid__startswith', 'enterprise_customer__name__startswith')**class** `enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin`(*model*, *admin_site*)Bases: `ModelAdmin`Django admin model for `EnterpriseCustomerReportingConfiguration`.**class Meta**Bases: `object`**model**alias of *EnterpriseCustomerReportingConfiguration*

```
autocomplete_fields = ['enterprise_customer']

form
    alias of EnterpriseCustomerReportingConfigAdminForm

get_fields(request, obj=None)
    Return the fields that should be displayed on the admin form.

list_display = ('enterprise_customer', 'active', 'delivery_method', 'frequency',
               'data_type', 'report_type')

list_filter = ('active',)

property media

ordering = ('enterprise_customer__name',)

search_fields = ('enterprise_customer__name', 'email')

class enterprise.admin.EnterpriseCustomerSsoConfigurationAdmin(model, admin_site)
    Bases: DjangoObjectActions, ModelAdmin
    Django admin for models.EnterpriseCustomerSsoConfigurationAdmin model.
    change_actions = ['mark_configured']
    list_display = ('uuid', 'enterprise_customer', 'active', 'identity_provider',
                   'created', 'configured_at')
    mark_configured(request, obj)
        Object tool handler method - marks the config as configured.
    property media
    model
        alias of EnterpriseCustomerSsoConfiguration

class enterprise.admin.EnterpriseCustomerTypeAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for EnterpriseCustomerType.
    class Meta
        Bases: object
        model
            alias of EnterpriseCustomerType
    fields = ('name',)
    list_display = ('name',)
    property media
    search_fields = ('name',)

class enterprise.admin.EnterpriseCustomerUserAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for EnterpriseCustomerUser.
```


class Meta

Bases: `object`

model

alias of `EnterpriseCustomerUser`

enterprise_enrollments(*enterprise_customer_user*)

Return a string representing a given EnterpriseCustomerUser's enterprise course enrollments

Parameters

enterprise_customer_user – The instance of EnterpriseCustomerUser being rendered with this admin form.

fields = ('user_id', 'enterprise_customer', 'user_email', 'username', 'created', 'enterprise_enrollments', 'other_enrollments', 'invite_key', 'active', 'should_inactivate_other_customers')

get_enrolled_course_string(*course_ids*)

Get an HTML string representing the courses the user is enrolled in.

get_enterprise_customer(*obj*)

Returns the name of enterprise customer linked with the enterprise customer user.

get_readonly_fields(*request, obj=None*)

Make all fields readonly when editing existing model.

get_search_results(*request, queryset, search_term*)

Return a tuple containing a queryset to implement the search and a boolean indicating if the results may contain duplicates.

list_display = ('username', 'user_email', 'get_enterprise_customer')

property media

other_enrollments(*enterprise_customer_user*)

Return a string representing a given EnterpriseCustomerUser's non-enterprise course enrollments

Parameters

enterprise_customer_user – The instance of EnterpriseCustomerUser being rendered with this admin form.

readonly_fields = ('user_email', 'username', 'created', 'enterprise_enrollments', 'other_enrollments')

search_fields = ('user_id', 'user_email')

class `enterprise.admin.EnterpriseFeatureUserRoleAssignmentAdmin`(*model, admin_site*)

Bases: `UserRoleAssignmentAdmin`

Django admin model for EnterpriseFeatureUserRoleAssignment.

class Meta

Bases: `object`

model

alias of `EnterpriseFeatureUserRoleAssignment`

form

alias of `EnterpriseFeatureUserRoleAssignmentForm`

property media

class enterprise.admin.**EnterpriseGroupAdmin**(*model, admin_site*)

Bases: `ModelAdmin`

Django admin for EnterpriseGroup model.

count(*obj*)

Return the number of members in a group

list_display = ('uuid', 'enterprise_customer', 'applies_to_all_contexts')

list_filter = ('applies_to_all_contexts',)

property media

members(*obj*)

Return the non-deleted members of a group

model

alias of `EnterpriseGroup`

readonly_fields = ('count', 'members')

search_fields = ('uuid', 'name', 'enterprise_customer__name',
'enterprise_customer__uuid')

class enterprise.admin.**EnterpriseGroupMembershipAdmin**(*model, admin_site*)

Bases: `ModelAdmin`

Django admin for EnterpriseGroupMembership model.

autocomplete_fields = ('group', 'enterprise_customer_user',
'pending_enterprise_customer_user')

list_display = ('group', 'membership_user')

property media

model

alias of `EnterpriseGroupMembership`

search_fields = ('uuid', 'group__enterprise_customer_user',
'enterprise_customer_user', 'pending_enterprise_customer_user')

class enterprise.admin.**PendingEnrollmentAdmin**(*model, admin_site*)

Bases: `ModelAdmin`

Django admin model for PendingEnrollment

class Meta

Bases: `object`

model

alias of `PendingEnrollment`

has_add_permission(*request*)

Disable add permission for PendingEnrollment.

```

has_delete_permission(request, obj=None)
    Disable deletion for PendingEnrollment.

list_display = ('user', 'course_id', 'course_mode')

property media

readonly_fields = ('user', 'course_id', 'course_mode')

search_fields = ('user__user_email', 'course_id')

class enterprise.admin.PendingEnterpriseCustomerAdminUserAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for PendingEnterpriseCustomerAdminUser

    class Meta
        Bases: object
        model
            alias of PendingEnterpriseCustomerAdminUser

    fields = ('user_email', 'enterprise_customer', 'get_admin_registration_url')

    get_admin_registration_url(obj)
        Formats the admin_registration_url model property as an HTML link.

    get_enterprise_customer(obj)
        Returns the name of the associated EnterpriseCustomer.

    list_display = ('user_email', 'get_enterprise_customer',
        'get_admin_registration_url')

    property media

    readonly_fields = ('get_admin_registration_url',)

    search_fields = ('user_email', 'enterprise_customer__name')

class enterprise.admin.PendingEnterpriseCustomerAdminUserInline(parent_model, admin_site)
    Bases: TabularInline
    Django admin inline model for PendingEnterpriseCustomerAdminUser.

    extra = 0

    fieldsets = ((None, {'fields': ('user_email', 'get_admin_registration_url')}),)

    get_admin_registration_url(obj)
        Formats the admin_registration_url model property as an HTML link.

    property media

    model
        alias of PendingEnterpriseCustomerAdminUser

    readonly_fields = ('get_admin_registration_url',)

```

```
class enterprise.admin.PendingEnterpriseCustomerUserAdmin(model, admin_site)
    Bases: ModelAdmin
    Django admin model for PendingEnterpriseCustomerUser
    class Meta
        Bases: object
        model
            alias of PendingEnterpriseCustomerUser
        fields = ('user_email', 'enterprise_customer', 'created')
        property media
        readonly_fields = ('user_email', 'enterprise_customer', 'created')
        search_fields = ('user_email', 'id')

class enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin(model, admin_site)
    Bases: UserRoleAssignmentAdmin
    Django admin model for SystemWideEnterpriseUserRoleAssignment.
    class Meta
        Bases: object
        model
            alias of SystemWideEnterpriseUserRoleAssignment
        fields = ('user', 'role', 'enterprise_customer', 'applies_to_all_contexts')
        form
            alias of SystemWideEnterpriseUserRoleAssignmentForm
        list_display = ('user', 'role', 'enterprise_customer', 'applies_to_all_contexts')
        list_per_page = 25
        list_select_related = ('user', 'role', 'enterprise_customer')
        property media
        paginator
            alias of BigTableMysqlPaginator
        search_fields = ('user__email', 'role__name', 'enterprise_customer__name')
        show_full_result_count = False
```

enterprise.api package

Subpackages

enterprise.api.v1 package

Subpackages

enterprise.api.v1.views package

Submodules

enterprise.api.v1.views.analytics_summary module

Views for the *ChatGPTResponse* API endpoint.

```
class enterprise.api.v1.views.analytics_summary.AnalyticsSummaryView(**kwargs)
    Bases: GenericAPIView
    API to generate a signed token for an enterprise admin to use Plotly analytics.
    authentication_classes = [<class
'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class
'rest_framework.authentication.SessionAuthentication'>]
    http_method_names = ['post']
    permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)
    post(request, enterprise_uuid)
        Generate auth token for plotly.
```

enterprise.api.v1.views.base_views module

Base API views for the enterprise app.

```
class enterprise.api.v1.views.base_views.EnterpriseModelViewSet
    Bases: EnterpriseViewSet
    Base class for attribute and method definitions common to all view sets.
    USER_ID_FILTER = 'id'
    filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class
'django_filters.rest_framework.backends.DjangoFilterBackend'>, <class
'enterprise.api.filters.UserFilterBackend'>)
    permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class
'rest_framework.permissions.DjangoModelPermissions'>)

class enterprise.api.v1.views.base_views.EnterpriseReadOnlyModelViewSet(**kwargs)
    Bases: EnterpriseModelViewSet, ReadOnlyModelViewSet
    Base class for all read only Enterprise model view sets.
```

```
class enterprise.api.v1.views.base_views.EnterpriseReadWriteModelViewSet(**kwargs)
    Bases: EnterpriseModelViewSet, ModelViewSet
    Base class for all read/write Enterprise model view sets.

    permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class
    'rest_framework.permissions.DjangoModelPermissions'>)

class enterprise.api.v1.views.base_views.EnterpriseViewSet
    Bases: object
    Base class for all Enterprise view sets.

    authentication_classes = (<class
    'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class
    'rest_framework.authentication.SessionAuthentication'>)

    ensure_data_exists(request, data, error_message=None)
        Ensure that the wrapped API client's response brings us valid data. If not, raise an error and log it.

    permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)

    throttle_classes = (<class 'enterprise.api.throttles.ServiceUserThrottle'>,)

class enterprise.api.v1.views.base_views.EnterpriseWrapperApiViewSet(**kwargs)
    Bases: EnterpriseViewSet, ViewSet
    Base class for attribute and method definitions common to all view sets which wrap external APIs.

class enterprise.api.v1.views.base_views.EnterpriseWriteOnlyModelViewSet(**kwargs)
    Bases: EnterpriseModelViewSet, CreateModelMixin, UpdateModelMixin, GenericViewSet
    Base class for all write only Enterprise model view sets.

    permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class
    'rest_framework.permissions.DjangoModelPermissions'>)
```

[enterprise.api.v1.views.coupon_codes module](#)

Views for coupon codes.

```
class enterprise.api.v1.views.coupon_codes.CouponCodesView(**kwargs)
    Bases: APIView
    API to request coupon codes.

    MISSING_REQUIRED_PARAMS_MSG = 'Some required parameter(s) missing: {}'

    OPTIONAL_PARAM_NOTES = 'notes'

    OPTIONAL_PARAM_NUMBER_OF_CODES = 'number_of_codes'

    REQUIRED_PARAM_EMAIL = 'email'

    REQUIRED_PARAM_ENTERPRISE_NAME = 'enterprise_name'

    authentication_classes = (<class
    'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class
    'rest_framework.authentication.SessionAuthentication'>)
```

get_missing_params_message(*parameter_state*)

Get a user-friendly message indicating a missing parameter for the API endpoint.

get_required_query_params(*request*)

Gets email, enterprise_name, number_of_codes, and notes, which are the relevant parameters for this API endpoint.

Parameters

request – The request to this endpoint.

Returns

The email, enterprise_name, number_of_codes and notes from the request.

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

post(*request*)

POST /enterprise/api/v1/request_codes

Requires a JSON object of the following format:

```
{
  "email": "bob@alice.com",
  "enterprise_name": "IBM",
  "number_of_codes": "50",
  "notes": "Help notes for codes request",
}
```

Keys

- **email** – Email of the customer who has requested more codes.
- **enterprise_name** – The name of the enterprise requesting more codes.
- **number_of_codes** – The number of codes requested.
- **notes** – Help notes related to codes request.

throttle_classes = (<class 'enterprise.api.throttles.ServiceUserThrottle'>,,)

enterprise.api.v1.views.enterprise_catalog_query module

Views for the enterprise-catalog-query API endpoint.

class enterprise.api.v1.views.enterprise_catalog_query.**EnterpriseCatalogQueryViewSet**(**kwargs)

Bases: [EnterpriseReadOnlyModelViewSet](#)

API views for the enterprise_catalog_query API endpoint.

authentication_classes = (<class 'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class 'rest_framework.authentication.SessionAuthentication'>)

basename = None

description = None

detail = None

```
name = None

pagination_class
    alias of ExpandDefaultPageSize

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class
'rest_framework.permissions.IsAdminUser'>)

queryset

serializer_class
    alias of EnterpriseCatalogQuerySerializer

suffix = None
```

```
class enterprise.api.v1.views.enterprise_catalog_query.ExpandDefaultPageSize
```

```
    Bases: PageNumberPagination
```

```
    Expands page size for the API. Used to populate support-tools repo's provisioning form catalog query dropdown
    component.
```

```
    page_size = 100
```

enterprise.api.v1.views.enterprise_course_enrollment module

Views for the enterprise-course-enrollment API endpoint.

```
class
```

```
enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentPagination
```

```
    Bases: DefaultPagination
```

```
    django_paginator_class
```

```
        alias of PaginatorWithOptimizedCount
```

```
class enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet(**kwargs)
```

```
    Bases: EnterpriseReadWriteModelViewSet
```

```
    API views for the enterprise-course-enrollment API endpoint.
```

```
    FIELDS = ('enterprise_customer_user', 'course_id')
```

```
    USER_ID_FILTER = 'enterprise_customer_user__user_id'
```

```
    basename = None
```

```
    description = None
```

```
    detail = None
```

```
    filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class
'django_filters.rest_framework.backends.DjangoFilterBackend'>, <class
'enterprise.api.filters.EnterpriseCourseEnrollmentFilterBackend'>)
```

```
    filterset_fields = ('enterprise_customer_user', 'course_id')
```


get_queryset()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

get_serializer_class()

Use a special serializer for any requests that aren't read-only.

name = None

ordering_fields = ('enterprise_customer_user', 'course_id')

pagination_class

alias of *EnterpriseCourseEnrollmentPagination*

queryset

suffix = None

```
class enterprise.api.v1.views.enterprise_course_enrollment.PaginatorWithOptimizedCount(object_list,
                                                                 per_page,
                                                                 or-
                                                                 phans=0,
                                                                 al-
                                                                 low_empty_first_page)
```

Bases: *Paginator*

Django < 4.2 ORM doesn't strip unused annotations from count queries.

For example, if we execute this query:

```
Book.objects.annotate(Count('chapters')).count()
```

it will generate SQL like this:

```
SELECT COUNT(*) FROM (SELECT COUNT(...), ... FROM ...) subquery
```

This isn't optimal on its own, but it's not a big deal. However, this becomes problematic when annotations use subqueries, because it's terribly inefficient to execute the subquery for every row in the outer query.

This class overrides the `count()` method of Django's `Paginator` to strip unused annotations from the query by requesting only the primary key instead of all fields.

This is a temporary workaround until Django is updated to 4.2, which will include a fix for this issue.

See <https://code.djangoproject.com/ticket/32169> for more details.

TODO: remove this class once Django is updated to 4.2 or higher.

count

```
enterprise.api.v1.views.enterprise_course_enrollment.read_replica_or_default()
```

enterprise.api.v1.views.enterprise_customer module

Views for the enterprise-customer API endpoint.

```
class enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet(**kwargs)
```

Bases: [EnterpriseReadWriteModelViewSet](#)

API views for the enterprise-customer API endpoint.

```
FIELDS = ('uuid', 'slug', 'name', 'active', 'site', 'enable_data_sharing_consent',  
'enforce_data_sharing_consent')
```

```
USER_ID_FILTER = 'enterprise_customer_users__user_id'
```

```
basename = None
```

```
basic_list(request, *arg, **kwargs)
```

Enterprise Customer's Basic data list without pagination

Two query parameters are supported: - name_or_uuid: filter by name or uuid substring search in a single query parameter. Primarily used for frontend debounced input search. - startswith: filter by name starting with the given string

```
contains_content_items(request, pk, course_run_ids, program_uuids)
```

Return whether or not the specified content is available to the EnterpriseCustomer.

Multiple course_run_ids and/or program_uuids query parameters can be sent to this view to check for their existence in the EnterpriseCustomerCatalogs associated with this EnterpriseCustomer. At least one course run key or program UUID value must be included in the request.

```
course_enrollments(request, pk)
```

Creates a course enrollment for an EnterpriseCustomerUser.

```
create(request, *args, **kwargs)
```

POST /enterprise/api/v1/enterprise-customer/

```
dashboard_list(request, *args, **kwargs)
```

Supports listing dashboard enterprises for frontend-app-admin-portal.

```
description = None
```

```
detail = None
```

```
enroll_learners_in_courses(request, pk)
```

Creates a set of enterprise enrollments for specified learners by bulk enrolling them in provided courses. This endpoint is not transactional, in that any one or more failures will not affect other successful enrollments made within the same request.

Parameters

- **enrollments_info** (*list of dicts*) – an array of dictionaries, each containing the necessary information to create an enrollment based on a subsidy for a user in a specified course. Each dictionary must contain the following keys:
 - 'user_id' OR 'email': Either unique identifier describing the user to enroll.
 - 'course_run_key': The course to enroll into.
 - 'license_uuid' OR 'transaction_id': ID of either accepted form of subsidy. *license_uuid* refers to subscription licenses, and *transaction_id* refers to Learner Credit transactions.

- 'force_enrollment' (bool, optional): Enroll even if enrollment deadline is expired (default False).

licenses_info is also accepted as a body param name.

Example:

```
enrollments_info: [
  {
    'email': 'newuser@test.com',
    'course_run_key': 'course-v1:edX+DemoX+Demo_Course',
    'license_uuid': '5b77bdbade7b4fcb838f8111b68e18ae',
  },
  {
    'email': 'newuser2@test.com',
    'course_run_key': 'course-v2:edX+FunX+Fun_Course',
    'transaction_id': '84kdbdbade7b4fcb838f8asjke8e18ae',
  },
  {
    'user_id': 1234,
    'course_run_key': 'course-v2:edX+SadX+Sad_Course',
    'transaction_id': 'ba1f7b61951987dc2e1743fa4886b62d',
  },
  ...
]
```

- **discount** (*int*) – the percent discount to be applied to all enrollments. Defaults to 100.

Returns

- **All users exist and are enrolled -**
{ 'successes': [], 'pending': [], 'failures': [] }, 201
- **Some or none of the users exist but are enrolled -**
{ 'successes': [], 'pending': [], 'failures': [] }, 202

Failure cases:

- **Some or all of the users can't be enrolled, no users were enrolled -**
{ 'successes': [], 'pending': [], 'failures': [] }, 409
- **Some or all of the provided emails are invalid**
{ 'successes': [], 'pending': [], 'failures': [] 'invalid_email_addresses': [] }, 409

Return type

Success cases

```
filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class
'django_filters.rest_framework.backends.DjangoFilterBackend'>, <class
'enterprise.api.filters.UserFilterBackend'>, <class
'enterprise.api.filters.EnterpriseLinkedUserFilterBackend'>)
```

```
filterset_fields = ('uuid', 'slug', 'name', 'active', 'site',
'enable_data_sharing_consent', 'enforce_data_sharing_consent')
```

get_permissions()

Instantiates and returns the list of permissions that this view requires.

get_serializer_class()

Return the class to use for the serializer. Defaults to using *self.serializer_class*.

You may want to override this if you need to provide different serializations depending on the incoming request.

(Eg. admins get full serialization, others get basic serialization)

name = None

ordering_fields = ('uuid', 'slug', 'name', 'active', 'site',
'enable_data_sharing_consent', 'enforce_data_sharing_consent')

pagination_class

alias of *PaginationWithFeatureFlags*

partial_update(*request*, **args*, ***kwargs*)

queryset

serializer_class

alias of *EnterpriseCustomerSerializer*

suffix = None

throttle_classes = (<class 'enterprise.api.throttles.HighServiceUserThrottle'>,))

toggle_universal_link(*request*, *pk=None*)

Enables/Disables universal link config.

unlink_users(*request*, *pk=None*)

Unlinks users with the given emails from the enterprise.

with_access_to(*request*, **args*, ***kwargs*)

Returns the list of enterprise customers the user has a specified group permission access to.

enterprise.api.v1.views.enterprise_customer_api_credentials module

Views for the Enterprise Customer API Credentials.

class

enterprise.api.v1.views.enterprise_customer_api_credentials.APICredEnabledPermission

Bases: *BasePermission*

Permission that checks to see if the request user matches the user indicated in the request body.

has_permission(*request*, *view*)

Return *True* if permission is granted, *False* otherwise.

class **enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsRegenerateViewSet**(***kwargs*)

Bases: *APICredentialsViewSet*

API views for the enterprise-customer-api-credentials API endpoint.

basename = None

description = None

detail = None

get_queryset()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.
(Eg. return a list of items that is specific to the user)

get_serializer_class()

Return the class to use for the serializer. Defaults to using *self.serializer_class*.

You may want to override this if you need to provide different serializations depending on the incoming request.

(Eg. admins get full serialization, others get basic serialization)

name = None

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class 'enterprise.api.v1.views.enterprise_customer_api_credentials.APICredEnabledPermission'>)

suffix = None

update(request, *args, **kwargs)

Regenerates the API application credentials (client ID and secret) for the requesting user. Throws a 404 if the user does not yet have any credentials.

Method: PUT

URL: /enterprise/api/v1/enterprise-customer-api-credentials/{enterprise_uuid}/regenerate_credentials

class enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsViewSet(kwargs)**

Bases: [EnterpriseReadWriteModelViewSet](#)

API views for the enterprise-customer-api-credentials API endpoint.

basename = None

create(request, *args, **kwargs)

Creates and returns a new enterprise API credential application record.

Method: POST

URL:

/enterprise/api/v1/enterprise-customer-api-credentials/{enterprise_uuid}

Returns

201 if a new API application credentials was created. If an application already exists for the user, throw a 409.

description = None

destroy(*request*, *args, **kwargs)

Removes the enterprise API application credentials for the requesting user.

Method: DELETE

URL: /enterprise/api/v1/enterprise-customer-api-credentials/{enterprise_uuid}

detail = None

get_queryset()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

get_serializer_class()

Return the class to use for the serializer. Defaults to using *self.serializer_class*.

You may want to override this if you need to provide different serializations depending on the incoming request.

(Eg. admins get full serialization, others get basic serialization)

name = None

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class 'enterprise.api.v1.views.enterprise_customer_api_credentials.APICredEnabledPermission'>)

retrieve(*request*, *args, **kwargs)

Returns the enterprise API application credentials details for the requesting user.

Method: GET

URL: /enterprise/api/v1/enterprise-customer-api-credentials/{enterprise_uuid}

suffix = None

update(*request*, *args, **kwargs)

Updates the enterprise API application credentials details for the requesting user.

Method: PUT

URL: /enterprise/api/v1/enterprise-customer-api-credentials/{enterprise_uuid}

enterprise.api.v1.views.enterprise_customer_branding_configuration module

Views for the enterprise-customer-branding API endpoint.

class enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConf

Bases: [EnterpriseReadWriteModelViewSet](#)

API views for the enterprise-customer-branding API endpoint.

FIELDS = ('enterprise_customer__slug',)

```

USER_ID_FILTER = 'enterprise_customer__enterprise_customer_users__user_id'

basename = None

description = None

detail = None

filterset_fields = ('enterprise_customer__slug',)

lookup_field = 'enterprise_customer__slug'

name = None

ordering_fields = ('enterprise_customer__slug',)

parser_classes = [<class 'rest_framework.parsers.MultiPartParser'>, <class
'rest_framework.parsers.FormParser'>]

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)

queryset

serializer_class
    alias of EnterpriseCustomerBrandingConfigurationSerializer

suffix = None

update_branding(request, enterprise_uuid)
    PATCH /enterprise/api/v1/enterprise-customer-branding/update_branding/uuid
    Requires enterprise customer uuid path parameter

```

enterprise.api.v1.views.enterprise_customer_catalog module

Write views for the enterprise-customer-catalog API endpoint.

```

class enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet(**kwargs)
    Bases: EnterpriseReadOnlyModelViewSet
    API Views for performing search through course discovery at the enterprise_catalogs API endpoint.
    FIELDS = ('uuid', 'title', 'enterprise_customer', 'enterprise_catalog_query',
    'created', 'modified')
    USER_ID_FILTER = 'enterprise_customer__enterprise_customer_users__user_id'
    basename = None
    contains_content_items(request, pk, course_run_ids, program_uuids)
        Return whether or not the EnterpriseCustomerCatalog contains the specified content.
        Multiple course_run_ids and/or program_uuids query parameters can be sent to this view to check for their
        existence in the EnterpriseCustomerCatalog. At least one course run key or program UUID value must be
        included in the request.

```

course_detail(*request, pk, course_key*)

Return the metadata for the specified course.

The course needs to be included in the specified EnterpriseCustomerCatalog in order for metadata to be returned from this endpoint.

course_run_detail(*request, pk, course_id*)

Return the metadata for the specified course run.

The course run needs to be included in the specified EnterpriseCustomerCatalog in order for metadata to be returned from this endpoint.

description = None

detail = None

filterset_fields = ('uuid', 'title', 'enterprise_customer',
'enterprise_catalog_query', 'created', 'modified')

get_serializer_class()

Return the class to use for the serializer. Defaults to using *self.serializer_class*.

You may want to override this if you need to provide different serializations depending on the incoming request.

(Eg. admins get full serialization, others get basic serialization)

list(*request, *args, **kwargs*)

name = None

ordering_fields = ('uuid', 'title', 'enterprise_customer',
'enterprise_catalog_query', 'created', 'modified')

program_detail(*request, pk, program_uuid*)

Return the metadata for the specified program.

The program needs to be included in the specified EnterpriseCustomerCatalog in order for metadata to be returned from this endpoint.

queryset

renderer_classes = (<class 'rest_framework.renderers.JSONRenderer'>, <class
'rest_framework_xml.renderers.XMLRenderer'>)

retrieve(*request, *args, **kwargs*)

suffix = None

class enterprise.api.v1.views.enterprise_customer_catalog.**EnterpriseCustomerCatalogWriteViewSet**(**kwargs)

Bases: [EnterpriseWriteOnlyModelViewSet](#)

API write only views for the enterprise-customer-catalog API endpoint.

basename = None

create(*request*, *args, **kwargs)

Creates a new EnterpriseCustomerCatalog and returns the created object.

If an EnterpriseCustomerCatalog already exists for the given enterprise_customer and enterprise_catalog_query, returns the existing object.

URL: /enterprise/api/v1/enterprise-customer-catalog/

Method: POST

Payload:

```
{
  "title": string - Title of the catalog,
  "enterprise_customer": string - UUID of an existing enterprise customer,
  "enterprise_catalog_query": string - id of an existing enterprise catalog_
  ↳ query,
}
```

Returns 201 if a new EnterpriseCustomerCatalog was created, 200 if an existing EnterpriseCustomerCatalog was

description = None

detail = None

has_enterprise_customer_catalog(*uuid*)

name = None

partial_update(*request*, *args, **kwargs)

Partially updates only the title in EnterpriseCustomerCatalog and returns the updated object.

URL: /enterprise/api/v1/enterprise-customer-catalog/

Method: PATCH

Payload:

```
{
  "uuid": string - UUID of an existing enterprise customer catalog,
  "title": string - Title of the catalog,
}
```

Returns 200 along with the updated object.

permission_classes = (<class 'rest_framework.permissions.IsAdminUser'>,,)

queryset

serializer_class

alias of *EnterpriseCustomerCatalogWriteOnlySerializer*

suffix = None

enterprise.api.v1.views.enterprise_customer_invite_key module

Views for enterprise customer invite keys.

class enterprise.api.v1.views.enterprise_customer_invite_key.**EnterpriseCustomerInviteKeyViewSet**(**kwargs)

Bases: [EnterpriseReadWriteModelViewSet](#)

API for accessing enterprise customer keys.

```
authentication_classes = (<class
'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class
'rest_framework.authentication.SessionAuthentication'>)
```

basename = None

basic_list(request, *args, **kwargs)

Unpaginated list of all invite keys matching the filters.

create(request, *args, **kwargs)

description = None

destroy(request, *args, **kwargs)

detail = None

```
filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class
'django_filters.rest_framework.backends.DjangoFilterBackend'>, <class
'enterprise.api.filters.EnterpriseCustomerInviteKeyFilterBackend'>)
```

get_serializer_class()

Use a special serializer for any requests that aren't read-only.

http_method_names = ['get', 'post', 'patch']

link_user(request, pk=None)

Post Links user using enterprise_customer_key /enterprise/api/enterprise-customer-invite-key/{enterprise_customer_key}/link-user

Given a enterprise_customer_key, link user to the appropriate enterprise.

If the key is not found, returns 404 If the key is not valid, returns 422 If we create an *EnterpriseCustomerUser* returns 201 If an *EnterpriseCustomerUser* if found returns 200

list(request, *args, **kwargs)

name = None

partial_update(request, *args, **kwargs)

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

queryset

retrieve(request, *args, **kwargs)

suffix = None

enterprise.api.v1.views.enterprise_customer_reporting module

Views for the Enterprise Customer Reporting API.

```
class enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportTypesView(**kwargs)
    Bases: APIView
    API for getting the report types associated with an enterprise customer

    authentication_classes = [<class
'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class
'rest_framework.authentication.SessionAuthentication'>]

    get(request, enterprise_uuid)
        Get the dropdown choices for EnterpriseCustomerReportingConfiguration

    http_method_names = ['get']

    permission_classes = [<class 'rest_framework.permissions.IsAuthenticated'>]

class enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingConfigurationView
    Bases: EnterpriseReadWriteModelViewSet
    API views for the enterprise-customer-reporting API endpoint.

    FIELDS = ('enterprise_customer',)

    USER_ID_FILTER = 'enterprise_customer__enterprise_customer_users__user_id'

    basename = None

    create(request, *args, **kwargs)

    description = None

    destroy(request, *args, **kwargs)

    detail = None

    filterset_fields = ('enterprise_customer',)

    list(request, *args, **kwargs)

    lookup_field = 'uuid'

    name = None

    ordering_fields = ('enterprise_customer',)

    partial_update(request, *args, **kwargs)

    permission_classes = [<class 'rest_framework.permissions.IsAuthenticated'>]

    queryset

    retrieve(request, *args, **kwargs)

    serializer_class
        alias of EnterpriseCustomerReportingConfigurationSerializer
```

suffix = None

update(*request*, *args, **kwargs)

enterprise.api.v1.views.enterprise_customer_sso_configuration module

Views for the enterprise-customer-sso-configuration API endpoint.

exception enterprise.api.v1.views.enterprise_customer_sso_configuration.
EnterpriseCustomerInactiveException

Bases: [Exception](#)

Exception raised when an enterprise customer is inactive.

class enterprise.api.v1.views.enterprise_customer_sso_configuration.**EnterpriseCustomerSsoConfigurationV**

Bases: [ModelViewSet](#)

API views for the EnterpriseCustomerSsoConfiguration model.

basename = None

create(*request*, *args, **kwargs)

description = None

destroy(*request*, *args, **kwargs)

detail = None

list(*request*, *args, **kwargs)

name = None

oauth_orchestration_complete(*request*, *configuration_uuid*, *args, **kwargs)

SSO orchestration completion callback. This endpoint is called by the SSO orchestrator when it has completed the configuration process.

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

queryset

retrieve(*request*, *args, **kwargs)

serializer_class

alias of [EnterpriseCustomerSsoConfiguration](#)

suffix = None

update(*request*, *args, **kwargs)

exception

enterprise.api.v1.views.enterprise_customer_sso_configuration.**EntityIdNotFoundError**

Bases: [Exception](#)

Exception raised by the SSO configuration api when it fails to fetch a customer IDP's entity ID from the metadata.

exception `enterprise.api.v1.views.enterprise_customer_sso_configuration.SsoConfigurationApiError(*args, **kwargs)`

Bases: `RequestException`

Exception raised when the Sso configuration api encounters an error while fetching provider metadata.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.check_user_part_of_customer(user, enterprise_customer)`

Checks if a user is in an enterprise customer.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.fetch_configuration_record(kwarg)`
Fetches the configuration record for the given uuid.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.fetch_entity_id_from_metadata_xml(metadata)`
Fetches the entity id from the metadata xml.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.fetch_request_data_from_request(request)`
Helper method to fetch the request data dictionary from the request object.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.get_customer_from_request(request)`
Gets the enterprise customer from the request or the user.

`enterprise.api.v1.views.enterprise_customer_sso_configuration.get_metadata_xml_from_url(url)`
Gets the metadata xml from the given url.

enterprise.api.v1.views.enterprise_customer_user module

Views for the enterprise-customer-user API endpoint.

class `enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserViewSet(**kwargs)`

Bases: `EnterpriseReadWriteModelViewSet`

API views for the enterprise-learner API endpoint.

FIELDS = ('enterprise_customer', 'user_id', 'active')

basename = None

description = None

detail = None

filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class 'django_filters.rest_framework.backends.DjangoFilterBackend'>, <class 'enterprise.api.filters.EnterpriseCustomerUserFilterBackend'>)

filterset_fields = ('enterprise_customer', 'user_id', 'active')

get_serializer_class()

Use a flat serializer for any requests that aren't read-only.

name = None

ordering_fields = ('enterprise_customer', 'user_id', 'active')

pagination_class

alias of *PaginationWithFeatureFlags*

queryset

suffix = None

enterprise.api.v1.views.enterprise_group module

Views for the enterprise-group API endpoint.

class enterprise.api.v1.views.enterprise_group.**EnterpriseGroupViewSet**(**kwargs)

Bases: *EnterpriseReadWriteModelViewSet*

API views for the enterprise-group API endpoint.

assign_learners(request, group_uuid)

POST /enterprise/api/v1/enterprise-group/<group uuid>/assign_learners

Request Arguments: - learner_emails: List of learner emails to associate with the group. Note: only processes the first 1000 records provided.

Optional request data: - act_by_date (datetime, optional): The expiration date for the subsidy. - catalog_uuid (string, optional): The uuid of the catalog that is part of the subsidy.

Returns: - records_processed: Total number of group membership records processed. - new_learners: Total number of group membership records associated with new pending enterprise learners that were processed. - existing_learners: Total number of group membership records associated with existing enterprise learners that were processed.

basename = None

create(request, *args, **kwargs)

POST /enterprise/api/v1/enterprise-group/

description = None

detail = None

filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class 'django_filters.rest_framework.backends.DjangoFilterBackend'>)

get_learners(request, *args, **kwargs)

Endpoint Location to return all learners: GET api/v1/enterprise-group/<group_uuid>/learners/

Endpoint Location to return pending learners: GET api/v1/enterprise-group/<group_uuid>/learners/?pending_users_only=true

Request Arguments: - group_uuid (URL location, required): The uuid of the group from which learners should be listed.

Optional query params: - pending_users_only (string, optional): Specify that results should only contain pending learners - q (string, optional): Filter the returned members by user email and name with a provided sub-string - sort_by (string, optional): Specify how the returned members should be ordered. Supported sorting values are *memberDetails*, *memberStatus*, and *recentAction*. Ordering can be reversed by supplying a - at the beginning of the sorting value ie *-memberStatus*. - page (int, optional): Which page of paginated data to return. - show_removed (bool, optional): Include removed learners in the response.

Returns: Paginated list of learners that are associated with the enterprise group uuid:

```
{
  'count': 1,
  'next': null,
  'previous': null,
  'results': [
    {
      'learner_id': integer or None,
      'pending_enterprise_customer_user_id': integer or None,
      'enterprise_group_membership_uuid': UUID,
      'member_details': {
        'user_email': string,
        'user_name': string,
      },
      'recent_action': string,
      'status': string,
    },
  ],
}
```

get_queryset(kwargs)**

- Filter down the queryset of groups available to the requesting user.
- Account for requested filtering query params

name = None

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

queryset

queryset_with_removed

remove_learners(request, group_uuid)

POST /enterprise/api/v1/enterprise-group/<group uuid>/remove_learners

Required Arguments:

- **learner_emails:**
List of learner emails to associate with the group.

Optional request data:

- **catalog_uuid** (string, optional): The uuid of the catalog that is part of the subsidy.

Returns

Number of membership records removed

Return type

- **records_deleted**

serializer_class

alias of *EnterpriseGroupSerializer*

suffix = None

update(request, *args, **kwargs)

PATCH /enterprise/api/v1/enterprise-group/<group uuid>

enterprise.api.v1.views.enterprise_subsidy_fulfillment module

Views for the Enterprise Subsidy Fulfillment API.

exception

`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnrollmentModificationException`

Bases: `Exception`

An exception that represents an error when modifying the state of an enrollment via the EnrollmentApiClient.

class `enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet(**kwargs)`

Bases: `EnterpriseWrapperApiViewSet`

General API views for subsidized enterprise course enrollments.

Supported operations:

- **Fetch a subsidy fulfillment record by uuid.**
`/enterprise/api/v1/subsidy-fulfillment/{fulfillment_source_uuid}/`
- **Cancel a subsidy fulfillment enrollment record by uuid.**
`/enterprise/api/v1/subsidy-fulfillment/{fulfillment_source_uuid}/cancel-enrollment/`
- **Fetch all unenrolled subsidy fulfillment records.**
`/enterprise/api/v1/operator/subsidy-fulfillment/unenrolled/`

Cancel and fetch endpoints require a fulfillment source uuid query parameter. Fetching unenrollments supports an optional `unenrolled_after` query parameter to filter the returned queryset down to only enterprise enrollments unenrolled after the supplied datetime.

Arguments (Fetch & Cancel):

`fulfillment_source_uuid` (str): The uuid of the subsidy fulfillment record.

Arguments (Unenrolled):

`unenrolled_after` (str): A datetime string. Only return enrollments unenrolled after this time.

Returns (Fetch):

(Response): JSON response containing the subsidy fulfillment record.

Returns (Unenrolled):

(Response): JSON list response containing the unenrolled subsidy fulfillment records.

```
api_response = [
    {
        enterprise_course_enrollment: {
            enterprise_customer_user: <user_id>,
            course_id: <course_id>,
            unenrolled: <datetime>
            created: <datetime>
        }
        license_uuid/transaction_id: <uuid>,
        uuid: <uuid>,
    },
]
```

Raises

(Http404): If the subsidy fulfillment record does not exist or if subsidy fulfillment exists under a separate enterprise. (Http403): If the requesting user does not have the appropriate permissions. (`EnrollmentModificationException`): If something goes wrong while updating the platform `CourseEnrollment` object.

basename = None

cancel_enrollment(*request*, *fulfillment_source_uuid*)

Cancel a single subsidized enrollment. Assumes fulfillment source has a valid enterprise enrollment.

/enterprise/api/v1/subsidy-fulfillment/{fulfillment_source_uuid}/cancel-enrollment/

description = None

detail = None

get_subsidy_fulfillment_queryset()

Return the queryset for this view. Queries across subsidy types until it finds a match for the provided uuid. Returns a 404 if no subsidy fulfillment record is found.

get_subsidy_fulfillment_serializer_class()

Fetch the correct serializer class based on the subsidy type.

get_unenrolled_fulfillment_queryset()

Return the queryset for unenrolled subsidy fulfillment records. Applies a modified timestamp filter to fetch records modified after if provided from query params.

get_unenrolled_fulfillment_serializer_class()

Fetch the correct recently unenrolled serializer class based on provided querysets.

name = None

retrieve(*request*, *fulfillment_source_uuid*, **args*, ***kwargs*)

Retrieve a single subsidized enrollment.

/enterprise/api/v1/subsidy-fulfillment/{fulfillment_source_uuid}/

suffix = None

unenrolled(*request*, **args*, ***kwargs*)

List all unenrolled subsidy fulfillments.

/enterprise/api/v1/operator/enterprise-subsidy-fulfillment/unenrolled/

Parameters

- **modified** (*str*) – A datetime string. Only return enrollments modified after this time.
- **retrieve_licensed_enrollments** (*bool*) – If true, return data related to licensed enrollments instead of learner credit

class enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseCourseEnrollmentViewSet(

Bases: [EnterpriseWrapperApiViewSet](#)

API views for the licensed-enterprise-course-enrollment API endpoint.

class EnrollmentTerminationStatus

Bases: [object](#)

Defines statuses related to enrollment states during the course unenrollment process.

COURSE_COMPLETED = 'course already completed'

MOVED_TO_AUDIT = 'moved to audit'

```
UNENROLLED = 'unenrolled'
```

```
UNENROLL_FAILED = 'unenroll_user_from_course returned false.'
```

```
OPT_IGNORE_ENROLLMENTS_MODIFIED_AFTER_PARAM = 'ignore_enrollments_modified_after'
```

```
REQ_EXP_LICENSE_UUIDS_PARAM = 'expired_license_uuids'
```

```
basename = None
```

```
bulk_licensed_enrollments_expiration(request)
```

Changes the mode for licensed enterprise course enrollments to the “audit” course mode, or unenroll the user if no audit mode exists for each expired license uuid

Parameters

- **expired_license_uuids** – The expired license uuids.
- **ignore_enrollments_modified_after** – All course enrollments modified past this given date will be ignored, i.e. the enterprise subscription plan expiration date.

```
description = None
```

```
detail = None
```

```
license_revoke(request, *args, **kwargs)
```

Changes the mode for a user’s licensed enterprise course enrollments to the “audit” course mode, or unenroll the user if no audit mode exists for a given course.

Will return a response with status 200 if no errors were encountered while modifying the course enrollment, or a 422 if any errors were encountered. The content of the response is of the form:

```
{
  'course-v1:puppies': {'success': true, 'message': 'unenrolled'},
  'course-v1:birds': {'success': true, 'message': 'moved to audit'},
  'course-v1:kittens': {'success': true, 'message': 'course already completed'},
  'course-v1:snakes': {'success': false, 'message': 'unenroll_user_from_
course returned false'},
  'course-v1:lizards': {'success': false, 'message': 'Some other exception'},
}
```

The first four messages are the values of constants that a client may expect to receive and parse accordingly.

```
name = None
```

```
queryset
```

```
serializer_class
```

alias of *LicensedEnterpriseCourseEnrollmentReadOnlySerializer*

```
suffix = None
```

enterprise.api.v1.views.notifications module

Views for the Admin Notification API.

class enterprise.api.v1.views.notifications.**NotificationReadView**(**kwargs)

Bases: APIView

API to mark notifications as read.

MISSING_REQUIRED_PARAMS_MSG = 'Some required parameter(s) missing: {}'

REQUIRED_PARAM_ENTERPRISE_SLUG = 'enterprise_slug'

REQUIRED_PARAM_NOTIFICATION_ID = 'notification_id'

authentication_classes = (<class 'edx_rest_framework_extensions.auth.jwt.authentication.JwtAuthentication'>, <class 'rest_framework.authentication.SessionAuthentication'>)

get_missing_params_message(parameter_state)

Get a user-friendly message indicating a missing parameter for the API endpoint.

get_required_query_params(request)

Gets notification_id and enterprise_slug. which are the relevant parameters for this API endpoint.

Parameters

request – The request to this endpoint.

Returns

The notification_id and enterprise_slug from the request.

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

post(request)

POST /enterprise/api/v1/read_notification

Requires a JSON object of the following format:

```
{
  'notification_id': 1,
  'enterprise_slug': 'enterprise_slug',
}
```

Keys

- **notification_id** – Notification ID which is read by Current User.
- **enterprise_slug** – The slug of the enterprise.

throttle_classes = (<class 'enterprise.api.throttles.ServiceUserThrottle'>,,)

enterprise.api.v1.views.pending_enterprise_customer_user module

Views for the pending-enterprise-customer-user API endpoint.

class enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserEnterpriseA

Bases: *PendingEnterpriseCustomerUserViewSet*

ViewSet for allowing enterprise admins to create linked learners Endpoint url: link_pending_enterprise_users/(?P<enterprise_uuid>[A-Za-z0-9-]+)/?\$ Admin must be an administrator for the enterprise in question

basename = None

description = None

detail = None

link_learners(request, enterprise_uuid)

Creates a PendingEnterpriseCustomerUser if no EnterpriseCustomerUser for the given (customer, email) combination(s) exists. Can accept one user or a list of users.

Returns 201 if any users were created, 204 if no users were created.

name = None

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,,)

serializer_class

alias of *LinkLearnersSerializer*

suffix = None

class enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserViewSet(**k

Bases: *EnterpriseReadWriteModelViewSet*

API views for the pending-enterprise-learner API endpoint. Requires staff permissions

FIELDS = ('enterprise_customer', 'user_email')

UNIQUE = 'unique'

USER_EXISTS_ERROR = 'EnterpriseCustomerUser record already exists'

basename = None

create(request, *args, **kwargs)

Creates a PendingEnterpriseCustomerUser if no EnterpriseCustomerUser for the given (customer, email) combination(s) exists. Can accept one user or a list of users.

Returns 201 if any users were created, 204 if no users were created.

description = None

detail = None

filter_backends = (<class 'rest_framework.filters.OrderingFilter'>, <class 'django_filters.rest_framework.backends.DjangoFilterBackend'>)

filterset_fields = ('enterprise_customer', 'user_email')

```

name = None

ordering_fields = ('enterprise_customer', 'user_email')

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>, <class
'rest_framework.permissions.IsAdminUser'>)>

queryset

serializer_class
    alias of PendingEnterpriseCustomerUserSerializer

suffix = None

```

enterprise.api.v1.views.plotly_auth module

Views for Plotly auth.

```
class enterprise.api.v1.views.plotly_auth.PlotlyAuthView(**kwargs)
```

Bases: `GenericAPIView`

API to generate a signed token for an enterprise admin to use Plotly analytics.

```
get(request, enterprise_uuid)
```

Generate auth token for plotly.

```
permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,>)
```

Module contents

API views for the enterprise app.

Submodules

enterprise.api.v1.decorators module

Decorators for Enterprise API views.

```
enterprise.api.v1.decorators.require_at_least_one_query_parameter(*query_parameter_names)
```

Ensure at least one of the specified query parameters are included in the request.

This decorator checks for the existence of at least one of the specified query parameters and passes the values as function parameters to the decorated view. If none of the specified query parameters are included in the request, a `ValidationError` is raised.

Usage:

```

@require_at_least_one_query_parameter('program_uuids', 'course_run_ids')
def my_view(request, program_uuids, course_run_ids):
    # Some functionality ...

```

enterprise.api.v1.fields module

Fields for Enterprise API serializers.

```
class enterprise.api.v1.fields.Base64EmailCSVField(*args, **kwargs)
```

Bases: `Field`

Serializers a Base64 encoded CSV with emails into an array of emails

```
to_internal_value(data)
```

Transform the *incoming* primitive data into a native value.

```
to_representation(value)
```

Transform the *outgoing* native value into primitive data.

enterprise.api.v1.permissions module

Custom API permissions.

```
class enterprise.api.v1.permissions.IsInEnterpriseGroup
```

Bases: `BasePermission`

Find out if the requesting user belongs to a django group meant for granting access to an enterprise feature. This check applies to both staff and non-staff users.

```
ALLOWED_API_GROUPS = []
```

```
has_permission(request, view)
```

Return *True* if permission is granted, *False* otherwise.

```
message = 'User is not allowed to access the view.'
```

enterprise.api.v1.serializers module

Serializers for enterprise api version 1.

```
class enterprise.api.v1.serializers.AdminNotificationSerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for AdminNotification model.

```
class Meta
```

Bases: `object`

```
fields = ('id', 'title', 'text')
```

```
model
```

alias of `AdminNotification`

```
class enterprise.api.v1.serializers.AnalyticsSummarySerializer(*args, **kwargs)
```

Bases: `Serializer`

Serializer for the payload data of analytics summary endpoint.

```
class LearnerEngagementSerializer(*args, **kwargs)
```

Bases: `Serializer`

Serializer for the summary related data in the analytics summary endpoint.

```
class LearnerProgressSerializer(*args, **kwargs)
```

Bases: `Serializer`

Serializer for the learner progress data in the analytics summary endpoint.

```
class enterprise.api.v1.serializers.BaseEnterpriseCustomerInviteKeySerializer(*args,
                                                                              **kwargs)
```

Bases: `ModelSerializer`

Base serializer for writing to the EnterpriseCustomerInviteKey model.

```
class Meta
```

Bases: `object`

```
fields = ('uuid', 'enterprise_customer_uuid', 'usage_limit', 'expiration_date',
          'is_active', 'is_valid')
```

```
model
```

alias of `EnterpriseCustomerInviteKey`

```
class enterprise.api.v1.serializers.CourseDetailSerializer(*args, **kwargs)
```

Bases: `ImmutableStateSerializer`

Serializer for course data retrieved from the discovery service course detail API endpoint.

This serializer updates the course and course run data with the EnterpriseCustomer-specific enrollment page URL for the given course and course runs.

```
to_representation(instance)
```

Return the updated course data dictionary.

Parameters

instance (`dict`) – The course data.

Returns

The updated course data.

Return type

`dict`

```
class enterprise.api.v1.serializers.CourseRunDetailSerializer(*args, **kwargs)
```

Bases: `ImmutableStateSerializer`

Serializer for course run data retrieved from the discovery service course_run detail API endpoint.

This serializer updates the course run data with the EnterpriseCustomer-specific enrollment page URL for the given course run.

```
to_representation(instance)
```

Return the updated course run data dictionary.

Parameters

instance (`dict`) – The course run data.

Returns

The updated course run data.

Return type`dict`

```
class enterprise.api.v1.serializers.EnrollmentsInfoSerializer(*args, **kwargs)
```

Bases: `Serializer`

Nested serializer class to allow for many license info dictionaries.

```
create(validated_data)
```

```
validate(data)
```

```
class enterprise.api.v1.serializers.EnterpriseCatalogQuerySerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for the `EnterpriseCatalogQuery` model.

```
class Meta
```

Bases: `object`

```
fields = '__all__'
```

```
model
```

alias of `EnterpriseCatalogQuery`

```
class enterprise.api.v1.serializers.EnterpriseCourseEnrollmentReadOnlySerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for `EnterpriseCourseEnrollment` model.

```
class Meta
```

Bases: `object`

```
fields = ('enterprise_customer_user', 'course_id', 'unenrolled_at', 'created')
```

```
model
```

alias of `EnterpriseCourseEnrollment`

```
class enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWithAdditionalFieldsReadOnlySerializer(*args, **kwargs)
```

Bases: `EnterpriseCourseEnrollmentReadOnlySerializer`

Serializer for `EnterpriseCourseEnrollment` model with additional fields.

```
class Meta
```

Bases: `object`

```
fields = ('enterprise_customer_user', 'course_id', 'created', 'unenrolled_at', 'enrollment_date', 'enrollment_track', 'user_email', 'course_start', 'course_end')
```

```
model
```

alias of `EnterpriseCourseEnrollment`

```
class enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWriteSerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for writing to the `EnterpriseCourseEnrollment` model.


```

class Meta
    Bases: object
    fields = ('username', 'course_id')

    model
        alias of EnterpriseCourseEnrollment

enterprise_customer_user = None

save()
    Save the model with the found EnterpriseCustomerUser.

validate_username(value)
    Verify that the username has a matching user, and that the user has an associated EnterpriseCustomerUser.
class enterprise.api.v1.serializers.EnterpriseCustomerApiCredentialRegeneratePatchSerializer(*args,
                                                                                          **kwargs)

    Bases: Serializer
    Serializer for the EnterpriseCustomerApiCredential

    class Meta
        Bases: object
        lookup_field = 'user'

class enterprise.api.v1.serializers.EnterpriseCustomerApiCredentialSerializer(*args,
                                                                              **kwargs)

    Bases: Serializer
    Serializer for the EnterpriseCustomerApiCredential

    class Meta
        Bases: object
        lookup_field = 'user'

    update(instance, validated_data)

class enterprise.api.v1.serializers.EnterpriseCustomerBasicSerializer(*args, **kwargs)
    Bases: ModelSerializer
    Serializer for EnterpriseCustomer model only for name and id fields.

    class Meta
        Bases: object
        fields = ('id', 'name')

    model
        alias of EnterpriseCustomer

class enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationSerializer(*args,
                                                                                      **kwargs)

    Bases: ModelSerializer
    Serializer for EnterpriseCustomerBrandingConfiguration model.

```

class `Meta`

Bases: `object`

`fields = ('enterprise_customer', 'enterprise_slug', 'logo', 'primary_color', 'secondary_color', 'tertiary_color')`

model

alias of `EnterpriseCustomerBrandingConfiguration`

get_enterprise_customer(*obj*)

Return a string representation of the associated enterprise customer's UUID.

get_enterprise_slug(*obj*)

Return the slug of the associated enterprise customer.

get_logo(*obj*)

Use `EnterpriseCustomerBrandingConfiguration.safe_logo_url` to return an absolute URL for either the saved customer logo or the platform logo by default

get_primary_color(*obj*)

Return the primary color of the branding config OR the default primary color code

get_secondary_color(*obj*)

Return the secondary color of the branding config OR the default secondary color code

get_tertiary_color(*obj*)

Return the tertiary color of the branding config OR the default tertiary color code

class `enterprise.api.v1.serializers.EnterpriseCustomerBulkEnrollmentsSerializer`(*args,
**kwargs)

Bases: `Serializer`

Serializes a `email_csv` or `email` field for bulk enrollment requests.

create(*validated_data*)

validate(*data*)

class `enterprise.api.v1.serializers.EnterpriseCustomerBulkSubscriptionEnrollmentsSerializer`(*args,
**kwargs)

Bases: `Serializer`

Serializes a `licenses_info` field for bulk enrollment requests.

create(*validated_data*)

validate(*data*)

class `enterprise.api.v1.serializers.EnterpriseCustomerCatalogDetailSerializer`(*args,
**kwargs)

Bases: `EnterpriseCustomerCatalogSerializer`

Serializer for the `EnterpriseCustomerCatalog` model which includes the catalog's discovery service search query results.

to_representation(*instance*)

Serialize the `EnterpriseCustomerCatalog` object.

Parameters

instance ([EnterpriseCustomerCatalog](#)) – The EnterpriseCustomerCatalog to serialize.

Returns

The EnterpriseCustomerCatalog converted to a dict.

Return type

dict

```
class enterprise.api.v1.serializers.EnterpriseCustomerCatalogSerializer(*args, **kwargs)
```

Bases: [ModelSerializer](#)

Serializer for the EnterpriseCustomerCatalog model.

class Meta

Bases: [object](#)

```
fields = ('uuid', 'title', 'enterprise_customer', 'enterprise_catalog_query',
          'created', 'modified')
```

model

alias of [EnterpriseCustomerCatalog](#)

```
class enterprise.api.v1.serializers.EnterpriseCustomerCatalogWriteOnlySerializer(*args,
                                                                                   **kwargs)
```

Bases: [EnterpriseCustomerCatalogSerializer](#)

Serializer for the EnterpriseCustomerCatalog model which includes the catalog's discovery service search query results.

class Meta

Bases: [object](#)

```
extra_kwargs = {'enterprise_catalog_query': {'required': False},
                 'enterprise_customer': {'required': True}, 'title': {'required': True},
                 'uuid': {'required': False}}
```

```
fields = ('uuid', 'title', 'enterprise_customer', 'enterprise_catalog_query')
```

model

alias of [EnterpriseCustomerCatalog](#)

```
class enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsListSerializer(*args,
                                                                                       **kwargs)
```

Bases: [ListSerializer](#)

Serializes a list of enrollment requests.

Meant to be used in conjunction with EnterpriseCustomerCourseEnrollmentsSerializer.

create(validated_data)

This selectively calls the child create method based on whether or not validation failed for each payload.

to_internal_value(data)

This implements the same relevant logic as ListSerializer except that if one or more items fail validation, processing for other items that did not fail will continue.

to_representation(data)

This selectively calls to_representation on each result that was processed by create.

```
class enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer(*args,
                                                                                **kwargs)
```

Bases: `Serializer`

Serializes enrollment information for a collection of students/emails.

This is mainly useful for implementing validation when performing enrollment operations.

```
class Meta
```

Bases: `object`

```
list_serializer_class
```

alias of `EnterpriseCustomerCourseEnrollmentsListSerializer`

```
create(validated_data)
```

Perform the enrollment for existing enterprise customer users, or create the pending objects for new users.

```
validate(data)
```

Validate that at least one of the user identifier fields has been passed in.

```
validate_course_run_id(value)
```

Validates that the course run id is part of the Enterprise Customer's catalog.

```
validate_lms_user_id(value)
```

Validates the lms_user_id, if is given, to see if there is an existing EnterpriseCustomerUser for it.

```
validate_tpa_user_id(value)
```

Validates the tpa_user_id, if is given, to see if there is an existing EnterpriseCustomerUser for it.

It first uses the third party auth api to find the associated username to do the lookup.

```
validate_user_email(value)
```

Validates the user_email, if given, to see if an existing EnterpriseCustomerUser exists for it.

If it does not, it does not fail validation, unlike for the other field validation methods above.

```
class enterprise.api.v1.serializers.EnterpriseCustomerIdentityProviderSerializer(*args,
                                                                                **kwargs)
```

Bases: `ModelSerializer`

Serializer for EnterpriseCustomerIdentityProvider model.

```
class Meta
```

Bases: `object`

```
fields = ('provider_id', 'default_provider')
```

```
model
```

alias of `EnterpriseCustomerIdentityProvider`

```
class enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyPartialUpdateSerializer(*args,
                                                                                       **kwargs)
```

Bases: `ModelSerializer`

Serializer for updating the EnterpriseCustomerInviteKey model.

```
class Meta
```

Bases: `object`

```
fields = ('expiration_date', 'is_active')
```

model

alias of *EnterpriseCustomerInviteKey*

```
class enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyReadOnlySerializer(*args,
                                                                                **kwargs)
```

Bases: *BaseEnterpriseCustomerInviteKeySerializer*

Serializer for reading the EnterpriseCustomerInviteKey model.

class Meta

Bases: *Meta*

additional_fields = ('enterprise_customer_name', 'usage_count', 'created')

fields = ('uuid', 'enterprise_customer_uuid', 'usage_limit', 'expiration_date', 'is_active', 'is_valid', 'enterprise_customer_name', 'usage_count', 'created')

get_enterprise_customer_name(*obj*)

get_enterprise_customer_uuid(*obj*)

```
class enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyWriteSerializer(*args,
                                                                              **kwargs)
```

Bases: *BaseEnterpriseCustomerInviteKeySerializer*

Serializer for writing to the EnterpriseCustomerInviteKey model.

save()

validate_enterprise_customer_uuid(*value*)

Validates an EnterpriseCustomer with the given enterprise_customer_uuid exists.

```
class enterprise.api.v1.serializers.EnterpriseCustomerReportingConfigurationSerializer(*args,
                                                                                      **kwargs)
```

Bases: *ModelSerializer*

Serializer for EnterpriseCustomerReportingConfiguration model.

class Meta

Bases: *object*

fields = ('enterprise_customer', 'enterprise_customer_id', 'active', 'delivery_method', 'email', 'frequency', 'day_of_month', 'day_of_week', 'hour_of_day', 'include_date', 'encrypted_password', 'sftp_hostname', 'sftp_port', 'sftp_username', 'encrypted_sftp_password', 'sftp_file_path', 'data_type', 'report_type', 'pgp_encryption_key', 'enterprise_customer_catalogs', 'uuid', 'enterprise_customer_catalog_uuids', 'enable_compression')

model

alias of *EnterpriseCustomerReportingConfiguration*

create(*validated_data*)

Perform the creation of model instance and link the enterprise customer catalogs.

Parameters

validated_data (*dict*) – A dictionary containing serializer's validated data.

Returns

Instance of the newly created enterprise customer reporting configuration.

Return type

(*EnterpriseCustomerReportingConfiguration*)

update(*instance*, *validated_data*)

Update the instance of enterprise customer reporting configuration and link the enterprise customer catalogs.

Parameters

- **instance** (*EnterpriseCustomerReportingConfiguration*) – Instance of the enterprise customer reporting configuration being updated.
- **validated_data** (*dict*) – A dictionary containing serializer's validated data.

Returns

Instance of the newly created enterprise customer reporting configuration.

Return type

(*EnterpriseCustomerReportingConfiguration*)

validate(*data*)

validate_pgp_encryption_key(*value*)

Validate that pgp_encryption_key is correctly set or left empty.

class enterprise.api.v1.serializers.**EnterpriseCustomerSerializer**(*args, **kwargs)

Bases: *ModelSerializer*

Serializer for EnterpriseCustomer model.

class **Meta**

Bases: *object*

```
fields = ('uuid', 'name', 'slug', 'active', 'auth_org_id', 'site',
'enable_data_sharing_consent', 'enforce_data_sharing_consent',
'branding_configuration', 'identity_provider', 'enable_audit_enrollment',
'replace_sensitive_sso_username', 'enable_portal_code_management_screen',
'sync_learner_profile_data', 'enable_audit_data_reporting',
'enable_learner_portal', 'enable_learner_portal_offers',
'enable_portal_learner_credit_management_screen',
'enable_executive_education_2U_fulfillment',
'enable_portal_reporting_config_screen',
'enable_portal_saml_configuration_screen', 'contact_email',
'enable_portal_subscription_management_screen', 'hide_course_original_price',
'enable_analytics_screen', 'enable_integrated_customer_learner_portal_search',
'enable_generation_of_api_credentials',
'enable_portal_lms_configurations_screen', 'sender_alias',
'identity_providers', 'enterprise_customer_catalogs', 'reply_to',
'enterprise_notification_banner', 'hide_labor_market_data', 'modified',
'enable_universal_link', 'enable_browse_and_request', 'admin_users',
'enable_career_engagement_network_on_learner_portal',
'career_engagement_network_message', 'enable_pathways', 'enable_programs',
'enable_demo_data_for_analytics_and_lpr', 'enable_academies',
'enable_one_academy', 'active_integrations')
```

```

    model
        alias of EnterpriseCustomer

    get_active_integrations(obj)

    get_admin_users(obj)

    get_branding_configuration(obj)
        Return the serialized branding configuration object OR default object if null

    get_enterprise_customer_catalogs(obj)
        Return list of catalog uuids associated with the enterprise customer.

    get_enterprise_notification_banner(obj)
        Return the notification text if exist OR None

class enterprise.api.v1.serializers.EnterpriseCustomerSsoConfiguration(*args, **kwargs)
    Bases: ModelSerializer

    Serializer for the EnterpriseCustomerSsoConfiguration model.

    class Meta
        Bases: object
        fields = '__all__'

    model
        alias of EnterpriseCustomerSsoConfiguration

    get_enterprise_customer(obj)
        Return a string representation of the associated enterprise customer's UUID.

    get_is_pending_configuration(obj)
        Return whether the SSO configuration is pending configuration.

class enterprise.api.v1.serializers.EnterpriseCustomerToggleUniversalLinkSerializer(*args,
                                                                                    **kwargs)

    Bases: Serializer

    Serializer for toggling an EnterpriseCustomer enable_universal_link field.

class enterprise.api.v1.serializers.EnterpriseCustomerUnlinkUsersSerializer(*args,
                                                                              **kwargs)

    Bases: Serializer

    Serializer for the EnterpriseCustomerViewSet unlink_users action.

class enterprise.api.v1.serializers.EnterpriseCustomerUserReadOnlySerializer(*args,
                                                                              **kwargs)

    Bases: ModelSerializer

    Serializer for EnterpriseCustomerUser model.

    class Meta
        Bases: object
        fields = ('id', 'enterprise_customer', 'active', 'user_id', 'user',
                  'data_sharing_consent_records', 'groups', 'created', 'invite_key',
                  'role_assignments', 'enterprise_group')

```

model

alias of *EnterpriseCustomerUser*

get_data_sharing_consent_records(*obj*)

Return serialization of EnterpriseCustomerUser.data_sharing_consent_records property.

Parameters

EnterpriseCustomerUser – The EnterpriseCustomerUser.

Returns

The serialized DataSharingConsent records associated with the EnterpriseCustomerUser.

Return type

list of dict

get_enterprise_group(*obj*)

Return the enterprise group membership for this enterprise customer user.

get_groups(*obj*)

Return the enterprise related django groups that this user is a part of.

get_role_assignments(*obj*)

Return the enterprise role assignments for this enterprise customer user.

class enterprise.api.v1.serializers.**EnterpriseCustomerUserWriteSerializer**(*args, **kwargs)

Bases: ModelSerializer

Serializer for writing to the EnterpriseCustomerUser model.

class Meta

Bases: object

fields = ('enterprise_customer', 'username', 'active')

model

alias of *EnterpriseCustomerUser*

USER_DOES_NOT_EXIST = 'User does not exist'

save()

Save the EnterpriseCustomerUser.

user = None

validate_username(*value*)

Verify that the username has a matching user.

class enterprise.api.v1.serializers.**EnterpriseGroupLearnersRequestQuerySerializer**(*args, **kwargs)

Bases: Serializer

Serializer for the Enterprise Group Learners endpoint query filter

class enterprise.api.v1.serializers.**EnterpriseGroupMembershipSerializer**(*args, **kwargs)

Bases: ModelSerializer

Serializer for EnterpriseGroupMembership model.

class Meta

Bases: object


```

    fields = ('enterprise_customer_user_id', 'lms_user_id',
              'pending_enterprise_customer_user_id', 'enterprise_group_membership_uuid',
              'member_details', 'recent_action', 'status', 'activated_at')

    model
        alias of EnterpriseGroupMembership

get_member_details(obj)
    Return either the member's name and email if it's the case that the member is realized, otherwise just email

get_recent_action(obj)
    Return the timestamp and name of the most recent action associated with the membership.

class enterprise.api.v1.serializers.EnterpriseGroupRequestDataSerializer(*args, **kwargs)
    Bases: Serializer

    Serializer for the Enterprise Group Assign Learners endpoint query params

class enterprise.api.v1.serializers.EnterpriseGroupSerializer(*args, **kwargs)
    Bases: ModelSerializer

    Serializer for EnterpriseGroup model.

class Meta
    Bases: object

    fields = ('enterprise_customer', 'name', 'uuid', 'applies_to_all_contexts')

    model
        alias of EnterpriseGroup

class enterprise.api.v1.serializers.ImmutableStateSerializer(*args, **kwargs)
    Bases: Serializer

    Base serializer for any serializer that inhibits state changing requests.

create(validated_data)
    Do not perform any operations for state changing requests.

update(instance, validated_data)
    Do not perform any operations for state changing requests.

class enterprise.api.v1.serializers.LearnerCreditEnterpriseCourseEnrollmentReadOnlySerializer(*args,
                                                                                           **kwargs)

    Bases: ModelSerializer

    Serializer for LearnerCreditEnterpriseCourseEnrollment model.

class Meta
    Bases: object

    fields = ('enterprise_course_enrollment', 'transaction_id', 'uuid')

    model
        alias of LearnerCreditEnterpriseCourseEnrollment

class enterprise.api.v1.serializers.LicensedEnterpriseCourseEnrollmentReadOnlySerializer(*args,
                                                                                           **kwargs)

    Bases: ModelSerializer

    Serializer for LicensedEnterpriseCourseEnrollment model.

```

```
class Meta
    Bases: object

    fields = ('enterprise_course_enrollment', 'license_uuid', 'uuid')

    model
        alias of LicensedEnterpriseCourseEnrollment
```

class `enterprise.api.v1.serializers.LinkLearnersSerializer(*args, **kwargs)`

Bases: *PendingEnterpriseCustomerUserSerializer*

Extends the PendingEnterpriseCustomerSerializer to validate that the enterprise customer uuid matches the uuid the user has permissions to update

NOT_AUTHORIZED_ERROR = 'Not authorized for this enterprise'

validate_enterprise_customer(*value*)

Check that the enterprise customer is the same as the one the user has permissions for The value recieved is an EnterpriseCustomer object

class `enterprise.api.v1.serializers.PendingEnterpriseCustomerUserSerializer(*args, **kwargs)`

Bases: *ModelSerializer*

Serializer for writing to the PendingEnterpriseCustomerUser model.

```
class Meta
    Bases: object

    fields = ('enterprise_customer', 'user_email')

    model
        alias of PendingEnterpriseCustomerUser
```

create(*attrs*)

Create the PendingEnterpriseCustomerUser, or EnterpriseCustomerUser if a user with the validated_email already exists.

to_representation(*instance*)

Because we are returning whether or not the instance was created from the create method, we must use the instance for to_representation and ignore the “created” half of the tuple

class `enterprise.api.v1.serializers.ProgramDetailSerializer(*args, **kwargs)`

Bases: *ImmutableStateSerializer*

Serializer for program data retrieved from the discovery service program detail API endpoint.

This serializer updates the program data and child course run data with EnterpriseCustomer-specific enrollment page URLs for the given content types.

to_representation(*instance*)

Return the updated program data dictionary.

Parameters

instance (*dict*) – The program data.

Returns

The updated program data.

Return type

dict

```
class enterprise.api.v1.serializers.ResponsePaginationSerializer(*args, **kwargs)
```

Bases: `ImmutableStateSerializer`

Serializer for responses that require pagination.

```
class enterprise.api.v1.serializers.SiteField(*args, **kwargs)
```

Bases: `Field`

Custom Site field to facilitate with creation of parent objects, while also keeping output pretty.

When used in a `ModelSerializer`, the site field can be provided to the `create()` REST API endpoint as follows, which performs a lookup for a site with the domain “example.com”:

```
"site": {"domain": "example.com"}
```

Output serializations render sites with all Site fields.

```
to_internal_value(data)
```

Transform the *incoming* primitive data into a native value.

```
to_representation(value)
```

Transform the *outgoing* native value into primitive data.

```
class enterprise.api.v1.serializers.SiteSerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for Site model.

```
class Meta
```

Bases: `object`

```
fields = ('domain', 'name')
```

```
model
```

alias of `Site`

```
class enterprise.api.v1.serializers.UserSerializer(*args, **kwargs)
```

Bases: `ModelSerializer`

Serializer for User model.

```
class Meta
```

Bases: `object`

```
fields = ('id', 'username', 'first_name', 'last_name', 'email', 'is_staff',
          'is_active', 'date_joined')
```

```
model
```

alias of `User`

enterprise.api.v1.urls module

URL definitions for enterprise api version 1 endpoint.

Module contents

API endpoint for enterprise app.

Submodules

enterprise.api.filters module

Filters for enterprise API.

class enterprise.api.filters.**EnterpriseCourseEnrollmentFilterBackend**

Bases: BaseFilterBackend

Filter backend to return enrollments under the user's enterprise(s) only.

- Staff users will bypass this filter.
- Non-staff users will receive enrollments under their linked enterprises, only if they have the *enterprise.can_enroll_learners* permission.
- Non-staff users without the *enterprise.can_enroll_learners* permission will receive only their own enrollments.

filter_queryset(*request, queryset, view*)

Filter out enrollments if learner is not linked

class enterprise.api.filters.**EnterpriseCustomerInviteKeyFilterBackend**

Bases: BaseFilterBackend

Filter backend to return invite keys under the user's enterprise(s) only. Supports filtering by enterprise_customer_uuid.

- Staff users will bypass this filter.

filter_queryset(*request, queryset, view*)

Return a filtered queryset.

class enterprise.api.filters.**EnterpriseCustomerUserFilterBackend**

Bases: BaseFilterBackend

Allow filtering on the enterprise customer user api endpoint.

filter_queryset(*request, queryset, view*)

Apply incoming filters only if user is staff. If not, only filter by user's ID.

class enterprise.api.filters.**EnterpriseLinkedUserFilterBackend**

Bases: BaseFilterBackend

Filter backend to return user's linked enterprises only

- Staff users will bypass this filter.
- Non-staff users will receive only their linked enterprises.

filter_queryset(*request, queryset, view*)

Filter out enterprise customer if learner is not linked

class enterprise.api.filters.**UserFilterBackend**

Bases: BaseFilterBackend

Filter backend for any view that needs to filter against the requesting user's ID.

- Staff users will bypass this filter.
- Non-staff users will receive only those objects that match their own user ID.

This requires that *USER_ID_FILTER* be set in the view as a class variable, to identify the object's relationship to a user ID.

filter_queryset(*request, queryset, view*)

Filter only for the user's ID if non-staff.

enterprise.api.pagination module

Pagination helpers for enterprise api.

class enterprise.api.pagination.**PaginationWithFeatureFlags**

Bases: DefaultPagination

Adds a *features* dictionary to the default paginated response provided by *edx_rest_framework_extensions*. The *features* dict represents a collection of Waffle-based feature flags/samples/switches that may be used to control whether certain aspects of the system are enabled or disabled (e.g., feature flag turned on for all staff users but not turned on for real customers/learners).

get_paginated_response(*data*)

Modifies the default paginated response to include *enterprise_features* dict.

Parameters

- **self** – *PaginationWithFeatureFlags* instance.
- **data** (*dict*) – Results for current page.

Returns

DRF response object containing *enterprise_features* dict.

Return type

(*Response*)

enterprise.api.pagination.**get_paginated_response**(*data, request*)

Update pagination links in course catalog data and return DRF Response.

Parameters

- **data** (*dict*) – Dictionary containing catalog courses.
- **request** (*HttpRequest*) – Current request object.

Returns

DRF response object containing pagination links.

Return type

(*Response*)

enterprise.api.throttles module

Throttle classes for enterprise API.

class enterprise.api.throttles.**BaseThrottle**

Bases: `UserRateThrottle`

Base throttle class with common functionality.

allow_request(*request*, *view*)

Modify throttling for service users.

Updates throttling rate if the request is coming from the service user, and defaults to `UserRateThrottle`'s configured setting otherwise.

Updated throttling rate comes from `DEFAULT_THROTTLE_RATES` key in `REST_FRAMEWORK` setting. specific user throttling is specified in `DEFAULT_THROTTLE_RATES` by it's corresponding key.

```
REST_FRAMEWORK = {
    'DEFAULT_THROTTLE_RATES': {
        'service_user': '50/day',
        'high_service_user': '2000/minute',
    }
}
```

get_scope()

Get the scope of the throttle.

Raises

`NotImplementedError` – Must be implemented by subclasses

update_throttle_scope()

Update throttle scope based on the specific subclass.

class enterprise.api.throttles.**HighServiceUserThrottle**

Bases: `BaseThrottle`

A throttle for high service users.

get_scope()

Get the scope of the throttle.

Returns

The scope of the throttle.

Return type

`str`

class enterprise.api.throttles.**ServiceUserThrottle**

Bases: `BaseThrottle`

A throttle allowing the service user to override rate limiting.

get_scope()

Get the scope of the throttle.

Returns

The scope of the throttle.

Return type

`str`

enterprise.api.urls module

URL definitions for enterprise API endpoint.

enterprise.api.utils module

Utility functions for the Enterprise API.

`enterprise.api.utils.assign_feature_roles(user)`

Add the ENTERPRISE_DASHBOARD_ADMIN_ROLE, ENTERPRISE_REPORTING_CONFIG_ADMIN_ROLE, ENTERPRISE_CATALOG_ADMIN_ROLE feature roles if the user does not already have them

`enterprise.api.utils.create_message_body(email, enterprise_name, number_of_codes=None, notes=None)`

Return the message body with extra information added by user.

`enterprise.api.utils.delta_format(current, prior)`

Formate delta of the given numbers.

If the delta is positive, number is '+10'. If negative, change nothing, it will come through as '-10' by default.

`enterprise.api.utils.generate_prompt_for_learner_engagement_summary(engagement_data)`

Generate an OpenAI prompt to get the summary of learner engagement from engagement data.

Parameters

engagement_data (*dict*) – A dictionary containing learner engagement numbers for some enterprise customer.

Returns

OpenAI prompt for getting engagement summary.

Return type

(*str*)

`enterprise.api.utils.generate_prompt_for_learner_progress_summary(progress_data)`

Generate an OpenAI prompt to get the summary of learner progress from progress data.

Parameters

progress_data (*dict*) – A dictionary containing learner progress numbers for some enterprise customer.

Returns

OpenAI prompt for getting learner progress summary.

Return type

(*str*)

`enterprise.api.utils.get_ent_cust_from_enterprise_customer_key(enterprise_customer_key)`

Get the enterprise customer id given an enterprise customer key.

`enterprise.api.utils.get_ent_cust_from_report_config_uuid(uuid)`

Get the enterprise customer id given an enterprise report configuration UUID.

`enterprise.api.utils.get_enterprise_customer_from_catalog_id(catalog_id)`

Get the enterprise customer id given an enterprise customer catalog id.

`enterprise.api.utils.get_enterprise_customer_from_enterprise_group_id(group_id)`

Get the enterprise customer id given an enterprise customer group id.

`enterprise.api.utils.get_enterprise_customer_from_user_id(user_id)`

Get the enterprise customer id given an user id

`enterprise.api.utils.get_service_usernames()`

Return the set of service usernames that are given extended permissions in the API.

`enterprise.api.utils.has_api_credentials_enabled(enterprise_uuid)`

Check whether the enterprise customer can access to api credentials or not

`enterprise.api.utils.percentage_format(number)`

Turn float representation of percentage into a cleaner format (0.89 -> 89%)

Parameters

number (*float*) – Floating point number to format.

Returns

String representation of the float in percentage form.

Return type

(*str*)

`enterprise.api.utils.set_application_name_from_user_id(user_id)`

Get the enterprise customer user's name given a user id.

Module contents

Python API for various enterprise functionality.

`enterprise.api.activate_admin_permissions(enterprise_customer_user)`

Activates admin permissions for an existing PendingEnterpriseCustomerAdminUser.

Specifically, the “enterprise_admin” system-wide role is assigned to the user and the PendingEnterpriseCustomerAdminUser record is removed.

Requires an EnterpriseCustomerUser record to exist which ensures the user already has the “enterprise_learner” role as a prerequisite.

Parameters

enterprise_customer_user – an EnterpriseCustomerUser instance

enterprise.api_client package

Submodules

enterprise.api_client.braze module

API clients to communicate with Braze services.

class `enterprise.api_client.braze.BrazeAPIClient`

Bases: `BrazeClient`

API client for calls to Braze.

create_recipient (*user_email*, *lms_user_id*, *trigger_properties=None*)

Create a recipient object using the given *user_email* and *lms_user_id*. Identifies the given email address with any existing Braze alias records via the provided *lms_user_id*.

create_recipient_no_external_id(*user_email*)

Create a Braze recipient dict identified only by an alias based on their email.

generate_mailto_link(*emails*)

Generate a mailto link for the given emails.

enterprise.api_client.client module

Base API clients to communicate with edx services.

class enterprise.api_client.client.**APIClientMixin**

Bases: `object`

API client mixin.

API_BASE_URL = ''

APPEND_SLASH = False

get_api_url(*path*)

Helper method to construct the API URL.

Parameters

path (*str*) – the API endpoint path string.

class enterprise.api_client.client.**BackendServiceAPIClient**

Bases: `APIClientMixin`

API client based on OAuthAPIClient to communicate with edx services.

Uses the backend service user to make requests.

class enterprise.api_client.client.**NoAuthAPIClient**

Bases: `APIClientMixin`

API client based on requests.Session to communicate with edx services.

Used to call APIs which don't require authentication.

get_health()

Retrieve health details for the service.

Returns

Response containing the service health.

Return type

`dict`

class enterprise.api_client.client.**UserAPIClient**(*user*, *expires_in=3600*)

Bases: `APIClientMixin`

API client based on requests.Session to communicate with edx services.

Requires user object to instantiate the client with the jwt token authentication.

connect()

Connect to the REST API, authenticating with a JWT for the current user.

static refresh_token(*func*)

Use this method decorator to ensure the JWT token is refreshed when needed.

token_is_expired()

Return True if the JWT token has expired, False if not.

enterprise.api_client.discovery module

Utilities to get details from the course catalog API.

class enterprise.api_client.discovery.CourseCatalogApiClient(*user, site=None*)

Bases: *UserAPIClient*

The API client to make calls to the Catalog API.

APPEND_SLASH = True

CATALOGS_COURSES_ENDPOINT = 'catalogs/{}/courses/'

COURSES_ENDPOINT = 'courses'

COURSE_RUNS_ENDPOINT = 'course_runs'

DEFAULT_VALUE_SAFEGUARD = <object object>

PROGRAMS_ENDPOINT = 'programs'

PROGRAM_TYPES_ENDPOINT = 'program_types'

SEARCH_ALL_ENDPOINT = 'search/all/'

get_catalog_results(*content_filter_query, query_params=None, traverse_pagination=False*)

Return results from the cache or discovery service's search/all endpoint.

Parameters

- **content_filter_query** (*dict*) – query parameters used to filter catalog results.
- **query_params** (*dict*) – query parameters used to paginate results.
- **traverse_pagination** (*bool*) – True to return all results, False to return the paginated response. Defaults to False.

Returns

Paginated response or all the records.

Return type

dict

get_catalog_results_from_discovery(*content_filter_query, query_params=None, traverse_pagination=False*)

Return results from the discovery service's search/all endpoint.

get_course_and_course_run(*course_run_id*)

Return the course and course run metadata for the given course run ID.

Parameters

course_run_id (*str*) – The course run ID.

Returns

The course metadata and the course run metadata.

Return type

tuple

get_course_details(*course_id*)

Return the details of a single course by id - not a course run id.

Parameters

course_id (*str*) – The unique id for the course in question.

Returns

Details of the course in question.

Return type

dict

get_course_id(*course_identifier*)

Return the course id for the given course identifier. The *course_identifier* may be a course id or a course run id; in either case the course id will be returned.

The 'course id' is the identifier for a course (ex. edX+DemoX) The 'course run id' is the identifier for a run of a course (ex. edX+DemoX+demo_run)

Parameters

course_identifier (*str*) – The course id or course run id

Returns

course id

Return type

(*str*)

get_course_run(*course_run_id*)

Return course_run data, including name, ID and seats.

Parameters

course_run_id (*string*) – Course run ID (aka Course Key) in string format.

Returns

Course run data provided by Course Catalog API.

Return type

dict

get_course_run_identifiers(*course_run_id*)

Return all course and course run keys and uuids for the specified course run id

get_program_by_uuid(*program_uuid*)

Return single program by UUID, or None if not found.

Parameters

program_uuid (*string*) – Program UUID in string form

Returns

Program data provided by Course Catalog API

Return type

dict

get_program_course_keys(*program_uuid*)

Get a list of the course IDs (not course run IDs) contained in the program.

Parameters

program_uuid (*str*) – Program UUID in string form

Returns

List of course keys in string form that are included in the program

Return type`list(str)`**get_program_type_by_slug**(*slug*)

Get a program type by its slug.

Parameters**slug** (*str*) – The slug to identify the program type.**Returns**

A program type object.

Return type`dict`**traverse_pagination**(*response*, *api_url*, *content_filter_query*, *query_params*)

Traverse a paginated API response and extracts and concatenates “results” returned by API.

Parameters

- **response** (*dict*) – API response object.
- **api_url** (*str*) – API endpoint path.
- **content_filter_query** (*dict*) – query parameters used to filter catalog results.
- **query_params** (*dict*) – query parameters used to paginate results.

Returns

all the results returned by the API.

Return type`list`**class** `enterprise.api_client.discovery.CourseCatalogApiClient`(*site=None*)Bases: `CourseCatalogApiClient`

Catalog API client which uses the configured Catalog service user.

program_exists(*program_uuid*)

Get whether the program exists or not.

class `enterprise.api_client.discovery.NoAuthDiscoveryClient`Bases: `NoAuthAPIClient`

Class to build a course discovery client to make calls to the discovery service.

API_BASE_URL = 'http://localhost:18381/'**APPEND_SLASH** = `False``enterprise.api_client.discovery.get_course_catalog_api_service_client`(*site=None*)Returns an instance of the `CourseCatalogApiClient`**Parameters****site** – (Site)**Returns**

(CourseCatalogServiceClient)

enterprise.api_client.ecommerce module

Client for communicating with the E-Commerce API.

class enterprise.api_client.ecommerce.**EcommerceApiClient**(*user*)

Bases: `object`

The API client to make calls to the E-Commerce API.

create_manual_enrollment_orders(*enrollments*)

Calls ecommerce to create orders for the manual enrollments passed in.

enrollments should be a list of enrollments with the following format:

```
{
    "lms_user_id": <int>,
    "username": <str>,
    "email": <str>,
    "course_run_key": <str>
}
```

Since *student.CourseEnrollment* lives in LMS, we're just passing around dicts of the relevant information.

get_course_final_price(*mode*, *currency*='\$', *enterprise_catalog_uuid*=None)

Get course mode's SKU discounted price after applying any entitlement available for this user.

Returns

Discounted price of the course mode.

Return type

`str`

class enterprise.api_client.ecommerce.**NoAuthEcommerceClient**

Bases: `NoAuthAPIClient`

The E-Commerce API client without authentication.

API_BASE_URL = 'http://localhost:18130'

APPEND_SLASH = False

enterprise.api_client.enterprise module

Client for communicating with the Enterprise API.

class enterprise.api_client.enterprise.**EnterpriseApiClient**(*user*, *expires_in*=3600)

Bases: `UserAPIClient`

The API client to make calls to the Enterprise API.

API_BASE_URL = 'http://localhost:8000/enterprise/api/v1/'

APPEND_SLASH = True

DEFAULT_VALUE_SAFEGUARD = <object object>

ENTERPRISE_CUSTOMER_CATALOGS_ENDPOINT = 'enterprise_catalogs'

```
ENTERPRISE_CUSTOMER_ENDPOINT = 'enterprise-customer'
```

```
get_content_metadata(enterprise_customer, enterprise_catalogs=None)
```

Return all content metadata contained in the catalogs associated with the EnterpriseCustomer.

Parameters

- **enterprise_customer** ([EnterpriseCustomer](#)) – The EnterpriseCustomer to return content metadata for.
- **enterprise_catalogs** ([EnterpriseCustomerCatalog](#)) – Optional list of EnterpriseCustomerCatalog objects.

Returns

List of dicts containing content metadata.

Return type

[list](#)

[enterprise.api_client.enterprise_catalog module](#)

Client for communicating with the Enterprise API.

```
class enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient(user=None)
```

Bases: [UserAPIClient](#)

The API client to make calls to the Enterprise Catalog API.

```
API_BASE_URL = 'http://localhost:18160/api/v1/'
```

```
APPEND_SLASH = True
```

```
CATALOG_DIFF_ENDPOINT = 'enterprise-catalogs/{}/generate_diff'
```

```
CONTENT_METADATA_IDENTIFIER_ENDPOINT = 'enterprise-customer/{}/content-metadata/{}'
```

```
ENTERPRISE_CATALOG_ENDPOINT = 'enterprise-catalogs'
```

```
ENTERPRISE_CUSTOMER_ENDPOINT = 'enterprise-customer'
```

```
GET_CONTENT_METADATA_ENDPOINT = 'enterprise-catalogs/{}/get_content_metadata'
```

```
GET_CONTENT_METADATA_PAGE_SIZE = 50
```

```
REFRESH_CATALOG_ENDPOINT = 'enterprise-catalogs/{}/refresh_metadata'
```

```
contains_content_items(catalog_uuid, content_ids)
```

Checks whether an enterprise catalog contains the given content.

The enterprise catalog endpoint does not differentiate between `course_run_ids` and `program_uuids` so they can be used interchangeably. The two query parameters are left in for backwards compatibility with edx-enterprise.

```
create_enterprise_catalog(catalog_uuid, enterprise_id, enterprise_name, title, content_filter,
                          enabled_course_modes, publish_audit_enrollment_urls,
                          catalog_query_uuid, query_title=None,
                          include_exec_ed_2u_courses=False)
```

Creates an enterprise catalog.

delete_enterprise_catalog(*catalog_uuid*)

Deletes an enterprise catalog.

enterprise_contains_content_items(*enterprise_uuid, content_ids*)

Checks whether an enterprise customer has any catalogs that contain the provided content ids.

The endpoint does not differentiate between *course_run_ids* and *program_uuids* so they can be used interchangeably. The two query parameters are left in for backwards compatability with edx-enterprise.

get_catalog_content_count(*catalog_uuid*)

Gets the content metadata count for a catalog from the first page of results

get_catalog_diff(*enterprise_customer_catalog, content_keys, should_raise_exception=True*)

Gets the representational difference between a list of course keys and the current state of content under an enterprise catalog. This difference is returned as three buckets of data: *items_not_found*, *items_not_included* and *items_found*.

Parameters

- **enterprise_customer_catalog** ([EnterpriseCustomerCatalog](#)) – The catalog object whose content is being diffed.
- **content_keys** (*list*) – List of string content keys
- **should_raise_exception** (*Bool*) – Optional param for whether or not api response exceptions should be raised.

Returns

dictionaries of *content_keys* to create *items_to_delete* (*list*): dictionaries of *content_keys* to delete *items_found* (*list*): dictionaries of *content_keys* and *date_updated* datetimes of content to update

Return type

items_to_create (*list*)

get_content_metadata(*enterprise_customer, enterprise_catalogs=None, content_keys_filter=None*)

Return all content metadata contained in the catalogs associated with the *EnterpriseCustomer*.

Parameters

- **enterprise_customer** ([EnterpriseCustomer](#)) – The *EnterpriseCustomer* to return content metadata for.
- **enterprise_catalogs** ([EnterpriseCustomerCatalog](#)) – Optional list of *EnterpriseCustomerCatalog* objects.
- **content_keys_filter** (*List*) – List of content keys to filter by in the content metadata endpoint

Returns

List of dicts containing content metadata.

Return type

list

get_content_metadata_content_identifier(*enterprise_uuid, content_id*)

Return all content metadata contained in the catalogs associated with the given *EnterpriseCustomer* and *content_id*.

static get_content_metadata_url(*uuid*)

Get the url for the preview information for an enterprise catalog

get_enterprise_catalog(*catalog_uuid*, *should_raise_exception=True*)

Gets an enterprise catalog.

Parameters

- **catalog_uuid** (*uuid*) – The uuid of an EnterpriseCatalog instance
- **should_raise_exception** (*bool*) – Whether an exception should be logged if a catalog does not yet exist in enterprise-catalog. Defaults to True.

Returns

a dictionary representing an enterprise catalog

Return type

`dict`

refresh_catalogs(*enterprise_catalogs*)

Kicks off async tasks to refresh catalogs so recent changes will populate to production without needing to wait for the daily jobs to run

Parameters

enterprise_catalogs (`EnterpriseCustomerCatalog`) – List of EnterpriseCustomerCatalog objects to refresh

Returns

Dict of async task ids for each catalog id

Return type

`Dict`

traverse_get_content_metadata(*api_url*, *query*, *catalog_uuid*)

Helper method to traverse over a paginated response from the enterprise-catalog service's `get_content_metadata` endpoint.

update_enterprise_catalog(*catalog_uuid*, ***kwargs*)

Updates an enterprise catalog.

class `enterprise.api_client.enterprise_catalog.NoAuthEnterpriseCatalogClient`

Bases: `NoAuthAPIClient`

The enterprise catalog API client to make calls to the discovery service.

API_BASE_URL = `'http://localhost:18160'`

APPEND_SLASH = `False`

enterprise.api_client.lms module

Utilities to get details from the course catalog API.

class `enterprise.api_client.lms.CertificatesApiClient`(*user*, *expires_in=3600*)

Bases: `UserAPIClient`

The API client to make calls to the LMS Certificates API.

Note that this API client requires a JWT token, and so it keeps its token alive.

API_BASE_URL = `'http://localhost:8000/api/certificates/v0/'`

APPEND_SLASH = `True`

get_course_certificate(*course_id*, *username*)

Retrieve the certificate for the given username for the given *course_id*.

Args: * *course_id* (str): The string value of the course's unique identifier * *username* (str): The user-name ID identifying the user for which to retrieve the certificate

Returns:

a dict containing:

- **username**: A string representation of an user's username passed in the request.
- **course_id**: A string representation of a Course ID.
- **certificate_type**: A string representation of the certificate type.
- **created_date**: Datetime the certificate was created (tz-aware).
- **status**: A string representation of the certificate status.
- **is_passing**: True if the certificate has a passing status, False if not.
- **download_url**: A string representation of the certificate url.
- **grade**: A string representation of a float for the user's course grade.

class enterprise.api_client.lms.**CourseApiClient**

Bases: *NoAuthAPIClient*

The API client to make calls to the Course API.

API_BASE_URL = 'http://localhost:8000/api/courses/v1/'

APPEND_SLASH = True

get_course_details(*course_id*)

Retrieve all available details about a course.

Parameters

course_id (*str*) – The course ID identifying the course for which to retrieve details.

Returns

Contains keys identifying those course details available from the courses API (e.g., name).

Return type

dict

class enterprise.api_client.lms.**EmbargoApiClient**

Bases: *object*

Client interface for using the edx-platform embargo API.

static redirect_if_blocked(*request*, *course_run_ids*, *user=None*)

Return redirect to embargo error page if the given user is blocked.

class enterprise.api_client.lms.**EnrollmentApiClient**

Bases: *BackendServiceAPIClient*

The API client to make calls to the Enrollment API.

API_BASE_URL = 'http://localhost:8000/api/enrollment/v1/'

enroll_user_in_course(*username, course_id, mode, cohort=None, enterprise_uuid=None*)

Call the enrollment API to enroll the user in the course specified by *course_id*.

Parameters

- **username** (*str*) – The username by which the user goes on the OpenEdX platform
- **course_id** (*str*) – The string value of the course’s unique identifier
- **mode** (*str*) – The enrollment mode which should be used for the enrollment
- **cohort** (*str*) – Add the user to this named cohort
- **enterprise_uuid** (*str*) – Add course enterprise uuid

Returns

A dictionary containing details of the enrollment, including course details, mode, username, etc.

Return type

dict

get_course_details(*course_id*)

Query the Enrollment API for the course details of the given *course_id*.

Parameters

- **course_id** (*str*) – The string value of the course’s unique identifier

Returns

A dictionary containing details about the course, in an enrollment context (allowed modes, etc.)

Return type

dict

get_course_enrollment(*username, course_id*)

Query the enrollment API to get information about a single course enrollment.

Parameters

- **username** (*str*) – The username by which the user goes on the OpenEdX platform
- **course_id** (*str*) – The string value of the course’s unique identifier

Returns

A dictionary containing details of the enrollment, including course details, mode, username, etc.

Return type

dict

get_course_modes(*course_id*)

Query the Enrollment API for the specific course modes that are available for the given *course_id*.

Parameters

- **course_id** (*str*) – The string value of the course’s unique identifier

Returns

A list of course mode dictionaries.

Return type

list

get_enrolled_courses(*username*)

Query the enrollment API to get a list of the courses a user is enrolled in.

Parameters

username (*str*) – The username by which the user goes on the OpenEdX platform

Returns

A list of course objects, along with relevant user-specific enrollment details.

Return type

list

has_course_mode(*course_run_id*, *mode*)

Query the Enrollment API to see whether a course run has a given course mode available.

Parameters

course_run_id (*str*) – The string value of the course run’s unique identifier

Returns

Whether the course run has the given mode available for enrollment.

Return type

bool

is_enrolled(*username*, *course_run_id*)

Query the enrollment API and determine if a learner is enrolled in a course run.

Parameters

- **username** (*str*) – The username by which the user goes on the OpenEdX platform
- **course_run_id** (*str*) – The string value of the course’s unique identifier

Returns

Indicating whether the user is enrolled in the course run. Returns False under any errors.

Return type

bool

unenroll_user_from_course(*username*, *course_id*)

Call the enrollment API to unenroll the user in the course specified by *course_id*. :param username: The username by which the user goes on the OpenEdx platform :type username: str :param course_id: The string value of the course’s unique identifier :type course_id: str

Returns

Whether the unenrollment succeeded

Return type

bool

update_course_enrollment_mode_for_user(*username*, *course_id*, *mode*)

Call the enrollment API to update a user’s course enrollment to the specified mode, e.g. “audit”.

Parameters

- **username** (*str*) – The username by which the user goes on the OpenEdx platform
- **course_id** (*str*) – The string value of the course’s unique identifier
- **mode** (*str*) – The string value of the course mode, e.g. “audit”

Returns

A dictionary containing details of the enrollment, including course details, mode, user-name, etc.

Return type`dict`**class** `enterprise.api_client.lms.GradesApiClient`(*user*, *expires_in=3600*)Bases: `UserAPIClient`

The API client to make calls to the LMS Grades API.

Note that this API client requires a JWT token, and so it keeps its token alive.

API_BASE_URL = `'http://localhost:8000/api/grades/v1/'`**APPEND_SLASH** = `True`**MAX_RETRIES** = `4`**get_course_assessment_grades**(*course_id*, *username*)Retrieve the assessment grades for the given username for the given *course_id*.Args: * *course_id* (str): The string value of the course's unique identifier * *username* (str): The user-name ID identifying the user for which to retrieve the grade.

Raises:

HTTPError if no grade found for the given user+course.

Returns:

a list of dicts containing:

- **attempted**: A boolean representing whether the learner has attempted the subsection yet.
- **subsection_name**: String representation of the subsection's name.
- **category**: String representation of the subsection's category.
- **label**: String representation of the subsection's label.
- **score_possible**: The total amount of points that the learner could have earned on the subsection.
- **score_earned**: The total amount of points that the learner earned on the subsection.
- **percent**: A float representing the overall grade for the course.
- **module_id**: The ID of the subsection.

get_course_grade(*course_id*, *username*)Retrieve the grade for the given username for the given *course_id*.Args: * *course_id* (str): The string value of the course's unique identifier * *username* (str): The user-name ID identifying the user for which to retrieve the grade.

Raises:

HTTPError if no grade found for the given user+course.

Returns:

a dict containing:

- **username**: A string representation of a user's username passed in the request.
- **course_key**: A string representation of a Course ID.
- **passed**: Boolean representing whether the course has been passed according the course's grading policy.
- **percent**: A float representing the overall grade for the course

- `letter_grade`: A letter grade as defined in `grading_policy` (e.g. 'A' 'B' 'C' for 6.002x) or None

class `enterprise.api_client.lms.NoAuthLMSCient`

Bases: `NoAuthAPIClient`

The LMS API client to make calls to the LMS without authentication.

API_BASE_URL = 'http://localhost:8000'

APPEND_SLASH = False

get_health()

Retrieve health details for LMS service.

Returns

Response containing LMS service health.

Return type

`dict`

class `enterprise.api_client.lms.ThirdPartyAuthApiClient`(*user*, *expires_in=3600*)

Bases: `UserAPIClient`

The API client to make calls to the Third Party Auth API.

API_BASE_URL = 'http://localhost:8000/api/third_party_auth/v0/'

get_remote_id(*identity_provider*, *username*)

Retrieve the remote identifier for the given username.

Args: * *identity_provider* (str): identifier slug for the third-party authentication service used during SSO. * *username* (str): The username ID identifying the user for which to retrieve the remote name.

Returns

the remote name of the given user. None if not found.

Return type

string or None

get_username_from_remote_id(*identity_provider*, *remote_id*)

Retrieve the remote identifier for the given username.

Args: * *identity_provider* (str): identifier slug for the third-party authentication service used during SSO. * *remote_id* (str): The remote id identifying the user for which to retrieve the username.

Returns

the username of the given user. None if not found.

Return type

string or None

enterprise.api_client.open_ai module

OpenAI client

`enterprise.api_client.open_ai.chat_completion(prompt, role, model='gpt-3.5-turbo')`

Use chatGPT <https://api.openai.com/v1/chat/completions> endpoint to generate a response.

Parameters

- **prompt** (*str*) – ChatGPT prompt
- **role** (*str*) – ChatGPT role to assume for the prompt.
- **model** (*str*) – ChatGPT model to use for the prompt completion.

Returns

Prompt response from OpenAI.

Return type

(*str*)

enterprise.api_client.sso_orchestrator module

Api client for the SSO Orchestrator API.

class `enterprise.api_client.sso_orchestrator.EnterpriseSSOOrchestratorApiClient`

Bases: `object`

The enterprise API client to communicate with the SSO Orchestrator API. Reads conf settings values to determine orchestration paths and credentials.

Required settings values: - `LMS_ROOT_URL` - `SSO_ORCHESTRATOR_API_BASE_URL` - `SSO_ORCHESTRATOR_CONFIGURE_PATH` - `SSO_ORCHESTRATOR_BASIC_AUTH_USERNAME` - `SSO_ORCHESTRATOR_BASIC_AUTH_PASSWORD`

configure_edx_oauth(*enterprise_customer*)

Configure SSO to GetSmarter using edX credentials via Auth0.

Parameters

enterprise_customer (`EnterpriseCustomer`) – The enterprise customer for which to configure edX OAuth.

Returns

Auth0 Organization ID.

Return type

str

Raises

`SsoOrchestratorClientError` – If the request to the SSO Orchestrator API failed.

configure_sso_orchestration_record(*config_data*, *config_pk*, *enterprise_data*, *is_sap=False*, *updating_existing_record=False*, *sap_config_data=None*)

configure an SSO orchestration record

exception `enterprise.api_client.sso_orchestrator.SsoOrchestratorClientError`(*args, **kwargs)

Bases: `RequestException`

Indicate a problem when interacting with an enterprise api client.

Module contents

enterprise.config package

Submodules

enterprise.config.models module

Module for defining models needed for configuration of internal concerns like management command options and parameters.

class enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfig(*args, **kwargs)

Bases: ConfigurationModel

Model that specifies parameters for the update_role_assignments_with_customers management command.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

ROLE_CHOICES = [('enterprise_admin', 'enterprise_admin'), ('enterprise_learner', 'enterprise_learner'), ('enterprise_catalog_admin', 'enterprise_catalog_admin'), ('enterprise_openedx_operator', 'enterprise_openedx_operator')]

batch_size

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

change_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

changed_by

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

changed_by_id

dry_run

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enabled

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_change_date(*, field=<django.db.models.fields.DateTimeField: change_date>,
                        is_next=True, **kwargs)
```

```
get_previous_by_change_date(*, field=<django.db.models.fields.DateTimeField: change_date>,
                           is_next=False, **kwargs)
```

```
get_role_display(*, field=<django.db.models.fields.CharField: role>)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

role

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

Module contents

enterprise.heartbeat package

Submodules

enterprise.heartbeat.checks module

Perform health checks on the services enterprise is dependant on.

`enterprise.heartbeat.checks.check_discovery()`

Check if course discovery service is up and running and accessible via API.

Raises

(*DiscoveryNotAvailable*) – raised if LMS service is not accessible for some reason.

Returns

A tuple containing service name and a message.

Return type

(str, str)

`enterprise.heartbeat.checks.check_ecommerce()`

Check if E-Commerce service is up and running and accessible via API.

Raises

(*EcommerceNotAvailable*) – raised if LMS service is not accessible for some reason.

Returns

A tuple containing service name and a message.

Return type

(str, str)

`enterprise.heartbeat.checks.check_enterprise_catalog()`

Check if enterprise catalog service is up and running and accessible via API.

Raises

([`EnterpriseCatalogNotAvailable`](#)) – raised if LMS service is not accessible for some reason.

Returns

A tuple containing service name and a message.

Return type

(str, str)

`enterprise.heartbeat.checks.check_lms()`

Check if LMS service is up and running and accessible via API.

Raises

([`LMSNotAvailable`](#)) – raised if LMS service is not accessible for some reason.

Returns

A tuple containing service name and a message.

Return type

(str, str)

enterprise.heartbeat.exceptions module

Exceptions for use by enterprise heartbeat module.

exception `enterprise.heartbeat.exceptions.DiscoveryNotAvailable(message, *args)`

Bases: [`ServiceNotAvailable`](#)

Raised when Course Discovery service is not available.

service_name = 'Course Discovery'

exception `enterprise.heartbeat.exceptions.EcommerceNotAvailable(message, *args)`

Bases: [`ServiceNotAvailable`](#)

Raised when E-Commerce service is not available.

service_name = 'E-Commerce'

exception `enterprise.heartbeat.exceptions.EnterpriseCatalogNotAvailable(message, *args)`

Bases: [`ServiceNotAvailable`](#)

Raised when Enterprise Catalog service is not available.

service_name = 'Enterprise Catalog'

exception `enterprise.heartbeat.exceptions.LMSNotAvailable(message, *args)`

Bases: [`ServiceNotAvailable`](#)

Raised when Learning Management System (LMS) service is not available.

service_name = 'Learning Management System (LMS)'

exception `enterprise.heartbeat.exceptions.ServiceNotAvailable(message, service_name, *args)`

Bases: [`Exception`](#)

All heartbeat service status exception inherit from this exception.

```
message = ''  
service_name = ''
```

enterprise.heartbeat.throttles module

Throttles for heartbeat API endpoints.

```
class enterprise.heartbeat.throttles.HeartbeatRateThrottle  
    Bases: UserRateThrottle  
    Throttle class for streamlining heartbeat traffic to manage server load.  
    rate = '60/min'  
    scope = 'heartbeat'
```

enterprise.heartbeat.utils module

Run health checks on all the services enterprise service is dependant on.

```
class enterprise.heartbeat.utils.Status  
    Bases: object  
    Status constants for service health check.  
    OK = 'OK'  
    UNAVAILABLE = 'UNAVAILABLE'
```

```
enterprise.heartbeat.utils.run_checks()
```

Run health checks on all the services and return the status of each service.

Returns

first boolean un the tuple tells of all services are up and running or not,
Dictionary containing the following key value pairs status: 'OK' if all the services are up

Return type

(bool, dict)

enterprise.heartbeat.views module

Views for enterprise heartbeat check.

```
enterprise.heartbeat.views.heartbeat(request, *args, **kwargs)
```

Simple view that an external service can use to check if the app is up.

Returns

A json doc of service id: status or message. If the status for any service is anything other than True, it returns HTTP code 503 (Service Unavailable); otherwise, it returns 200.

Return type

(JsonResponse)

Module contents

Enterprise heartbeat related code.

enterprise.management package

Subpackages

enterprise.management.commands package

Submodules

enterprise.management.commands.add_exec_ed_exclusion_to_catalogs module

Django management command to add Exec Ed exclusion flag to catalogs

```
class enterprise.management.commands.add_exec_ed_exclusion_to_catalogs.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: BaseCommand

Enumerate the catalog filters and add course_type exclusions where required

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.backfill_learner_role_assignments module

Management command for assigning enterprise_learner roles to existing linked enterprise users that are missing them.

```
class enterprise.management.commands.backfill_learner_role_assignments.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: BaseCommand

Management command for assigning enterprise_learner roles to existing enterprise users.

Example usage:

```
$ ./manage.py backfill_learner_role_assignments
```

add_arguments(parser)

Entry point for subclassed commands to add custom arguments.

backfill_learner_role_assignments(options)

Assigns enterprise_learner role to users.

handle(*args, **options)

Entry point for management command execution.

help = 'Assigns enterprise_learner role to linked enterprise users missing them.'

enterprise.management.commands.bulk_update_catalog_query_id module

Django management command for bulk updating EnterpriseCustomerCatalog record's EnterpriseCatalogQuery's.

```
class enterprise.management.commands.bulk_update_catalog_query_id.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: BaseCommand

Takes a new and old Enterprise Catalog Query ID, find all most recent records within the historical Enterprise Catalog table where enterprise catalog query ID is the old ID provided and updates the corresponding Enterprise Catalog table entries with the new ID.

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

get_args_from_database()

Returns an options dictionary from the current BulkCatalogQueryUpdateCommandConfiguration model.

handle(*args, **options)

Entry point for management command execution.

help = "Updates specified EnterpriseCatalog's enterprise catalog query ID records
with a provided new ID"

enterprise.management.commands.compare_discovery_and_enterprise_catalogs module

Django management command to explore Exec Ed inclusion flag migration

```
class enterprise.management.commands.compare_discovery_and_enterprise_catalogs.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: BaseCommand

Enumerate the catalog filters and log information about how we might migrate them.

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.create_enterprise_course_enrollments module

Django management command for creating EnterpriseCourseEnrollment records.

```
class enterprise.management.commands.create_enterprise_course_enrollments.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: BaseCommand

Creates EnterpriseCourseEnrollment records (if they do not already exist) for CourseEnrollment records that are associated with an enterprise user and a course run that exists in the enterprise's catalog.

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

help = 'Create EnterpriseCourseEnrollment records for CourseEnrollment records associated with an enterprise.'

enterprise.management.commands.create_missing_dsc_records module

Django management command for creating DataSharingConsent records.

```
class enterprise.management.commands.create_missing_dsc_records.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: BaseCommand

Django management command for creating DataSharingConsent records

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.email_drip_for_missing_dsc_records module

Django management command for sending an email to learners with missing DataSharingConsent records.

```
class enterprise.management.commands.email_drip_for_missing_dsc_records.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: BaseCommand

Django management command for sending an email to learners with missing DataSharingConsent records

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

emit_event(*ec_user, course_id, enterprise_customer, greeting_name*)

Emit the Segment event which will be used by Braze to send the email

get_enterprise_course_enrollments(*options*)

Get EnterpriseCourseEnrollment records according to the options and with dsc enabled

handle(*args, **options)

Management command for sending an email to learners with a missing DSC. Designed to run daily.

Example usage:

```
$ ./manage.py email_drip_for_missing_dsc_records $ ./manage.py
email_drip_for_missing_dsc_records --no-commit $ ./manage.py
email_drip_for_missing_dsc_records --enrollment-before 2021-05-06 --no-commit $ ./manage.py
email_drip_for_missing_dsc_records --enrollment-before 2021-05-06
```

enterprise.management.commands.ensure_singular_active_enterprise_customer_user module

Django management command to ensure there is at most a single EnterpriseCustomerUser with *active=True* for each enterprise user.

```
class enterprise.management.commands.ensure_singular_active_enterprise_customer_user.Command(stdout=None, stderr=None, no_color=False, force_color=False)
```

Bases: BaseCommand

Django management command to ensure there is at most a single EnterpriseCustomerUser with *active=True* for each enterprise user.

Example usage: `./manage.py lms ensure_singular_active_enterprise_customer_user`

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(**args*, ***options*)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.fix_dsc_records module

Django management command to Fix DSC records having spaces in there course Id.

```
class enterprise.management.commands.fix_dsc_records.Command(stdout=None, stderr=None, no_color=False, force_color=False)
```

Bases: BaseCommand

Django management command to Fix DSC records having spaces in there course Id.

This Command fixes the DSC records what were saved due to bug in our system and DSC records were saved with spaces.

Example usage: `./manage.py lms fix_dsc_records ./manage.py lms fix_dsc_records --no-commit`

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(**args*, ***options*)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.manufacture_data module

Management command for making things with test factories

```
class enterprise.management.commands.manufacture_data.Command(stdout=None, stderr=None, no_color=False, force_color=False)
```

Bases: BaseCommand

Management command for generating Django records from factories with custom attributes

Example usage:

```
$ ./manage.py manufacture_data --model enterprise.models.enterprise_customer --name "Test Enterprise" --slug "test-enterprise"
```

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(**args*, ***options*)

Entry point for management command execution.

run_from_argv(*argv*)

Re-implemented from <https://github.com/django/django/blob/main/django/core/management/base.py#L395> in order to support individual field customization. We will need to keep this method up to date with our current version of Django BaseCommand.

Uses `parse_known_args` instead of `parse_args` to not throw an error when encountering unknown arguments

https://docs.python.org/3.8/library/argparse.html#argparse.ArgumentParser.parse_known_args

class `enterprise.management.commands.manufacture_data.Node`(*data*)

Bases: `object`

Non-binary tree node class for building out a dependency tree of objects to create with customizations.

add_child(*obj*)

Add a child to the current node

build_records()

Recursively build out the tree of objects by first dealing with children nodes before getting to the parent.

find_value(*value*)

Find a value in the tree

set_single_customization(*field*, *value*)

Set a single customization value to the current node, overrides existing values under the same key.

`enterprise.management.commands.manufacture_data.all_subclasses`(*cls*)

Recursively get all subclasses of a class <https://stackoverflow.com/a/3862957>

`enterprise.management.commands.manufacture_data.build_tree_from_field_list`(*list_of_fields*,
provided_factory,
base_node, *customization_value*)

Builds a non-binary tree of nodes based on a list of children nodes, using a base node and it's associated data factory as the parent node the user provided value as a reference to a potential, existing record.

- **list_of_fields (list of strings): the linked list of associated objects to create.** Example- `['enterprise_customer_user', 'enterprise_customer', 'site']`
- **provided_factory** (`factory.django.DjangoModelFactory`): The data factory of the `base_node`.
- **base_node** (`Node`): The parent node of the desired tree to build.
- **customization_value (string): The value to be assigned to the object associated with the last value in the** `list_of_fields` param. Can either be a FK if the last value is a subfactory, or alternatively a custom value to be assigned to the field. Example- `list_of_fields = ['enterprise_customer_user', 'enterprise_customer', 'site'], customization_value = 9` or `list_of_fields = ['enterprise_customer_user', 'enterprise_customer', 'name'], customization_value = "FRED"`

`enterprise.management.commands.manufacture_data.convert_to_pascal`(*string*)

helper method to convert strings to Pascal case.

`enterprise.management.commands.manufacture_data.pairwise(iterable)`

Convert a list into a list of tuples of adjacent elements. `s -> [(s0, s1), (s2, s3), (s4, s5), ...]`

`enterprise.management.commands.migrate_enterprise_catalogs` module

Django management command for migrating EnterpriseCustomerCatalog data to new service.

```
class enterprise.management.commands.migrate_enterprise_catalogs.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: `BaseCommand`

Migrate EnterpriseCustomerCatalog data to new Enterprise Catalog service.

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

handle(**args*, ***options*)

The actual logic of the command. Subclasses must implement this method.

help = 'Migrate EnterpriseCustomerCatalog data to new Enterprise Catalog service.'

`enterprise.management.commands.monthly_impact_report` module

Django management command to send monthly impact report to enterprise admins.

```
class enterprise.management.commands.monthly_impact_report.Command(stdout=None, stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)
```

Bases: `BaseCommand`

Django management command to send monthly impact report to enterprise admins.

Example usage: `./manage.py lms monthly_impact_report ./manage.py lms monthly_impact_report --no-commit`

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

emit_event(***kwargs*)

Emit the Segment event which will be used by Braze to send the email

get_query_results_from_snowflake()

Get query results from Snowflake and yield each row.

handle(**args*, ***options*)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.nudge_dormant_enrolled_enterprise_learners module

Django management command to send nudge email to dormant enrolled enterprise learners.

```
class enterprise.management.commands.nudge_dormant_enrolled_enterprise_learners.Command(stdout=None,  
                                                                 stderr=None,  
                                                                 no_color=False,  
                                                                 force_color=False)
```

Bases: BaseCommand

Django management command to send nudge email to dormant enrolled enterprise learners.

Example usage: `./manage.py lms nudge_dormant_enrolled_enterprise_learners` `./manage.py lms nudge_dormant_enrolled_enterprise_learners --no-commit`

add_arguments(*parser*)

Entry point for subclassed commands to add custom arguments.

emit_event(***kwargs*)

Emit the Segment event which will be used by Braze to send the email

get_query_results_from_snowflake()

Get query results from Snowflake and yield each row.

handle(**args, **options*)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.reencrypt_enterprise_customer_reporting_config_passwords module

Django management command to reencrypt passwords in enterprise custom reporting configs.

```
class enterprise.management.commands.reencrypt_enterprise_customer_reporting_config_passwords.Command(st  
                                                                 st  
                                                                 no  
                                                                 fo)
```

Bases: BaseCommand

Django management command to reencrypt passwords in enterprise custom reporting configs It's useful when following encryption keys are rotated - FERNET_KEYS - LMS_FERNET_KEY

Example usage: `./manage.py lms reencrypt_enterprise_customer_reporting_config_passwords`

handle(**args, **options*)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.remove_expired_pending_group_memberships module

Management command for ensuring any pending group membership, ie memberships associated with a pending enterprise user, are removed after 90 days.

```
class enterprise.management.commands.remove_expired_pending_group_memberships.Command(stdout=None,  
                                                                 stderr=None,  
                                                                 no_color=False,  
                                                                 force_color=False)
```

Bases: `BaseCommand`

Management command for ensuring any pending group membership, ie memberships associated with a pending enterprise user, are removed after 90 days. Optionally supply a `--enterprise_customer` arg to only run this command on a singular customer.

Example usage:

```
$ ./manage.py remove_expired_pending_group_memberships
```

`add_arguments(parser)`

Entry point for subclassed commands to add custom arguments.

`handle(*args, **options)`

The actual logic of the command. Subclasses must implement this method.

`help = 'Removes pending group memberships if they are older than 90 days.'`

enterprise.management.commands.revert_enrollment_objects module

Management command for reverting revoked enrollment related objects to a particular time.

```
class enterprise.management.commands.revert_enrollment_objects.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: `BaseCommand`

Management command for reverting revoked enrollment related objects to a particular time.

Example usage:

```
$ ./manage.py revert_enrollment_objects --year 2021 --month 11 --day 17 --enterprise-customer-name test-co
```

`add_arguments(parser)`

Entry point for subclassed commands to add custom arguments.

`handle(*args, **options)`

Entry point for management command execution.

`help = 'Reverts revoked enrollment related objects to a particular time.'`

`revert_enrollment_objects(options)`

Revert all `EnterpriseCourseEnrollment`, `LicensedEnterpriseCourseEnrollment`, and “student” `CourseEnrollment` objects to the date provided, using the history table IF `is_revoked = True` on `LicensedEnterpriseCourseEnrollment`

enterprise.management.commands.save_enterprise_customer_users module

Django management command for saving `EnterpriseCustomerUser` models.

```
class enterprise.management.commands.save_enterprise_customer_users.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: `BaseCommand`

Calls `save()` on `EnterpriseCustomerUser` models.

This is useful for triggering save-related signals causing the associated signal receiver functions to fire.

`add_arguments(parser)`

Entry point for subclassed commands to add custom arguments.

`handle(*args, **options)`

The actual logic of the command. Subclasses must implement this method.

`help = 'Save existing EnterpriseCustomerUser models.'`

`enterprise.management.commands.seed_enterprise_devstack_data` module

Management command for assigning enterprise roles to existing enterprise users.

```
class enterprise.management.commands.seed_enterprise_devstack_data.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: `BaseCommand`

Management command for populating Devstack with initial data for enterprise.

Example usage:

```
$ ./manage.py lms seed_enterprise_devstack_data
```

`add_arguments(parser)`

Entry point for subclassed commands to add custom arguments.

`handle(*args, **options)`

Entry point for management command execution.

**`help = '\n Seeds an enterprise customer, users of various roles and permissions
initial\n data in devstack for related Enterprise models.\n '`**

`enterprise.management.commands.unlink_enterprise_customer_learners` module

Django management command to unlink the learners, Delete the enrollments and remove the DSC.

```
class enterprise.management.commands.unlink_enterprise_customer_learners.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)
```

Bases: `BaseCommand`

Django management command to unlink the learners, Delete the enrollments and remove the DSC

Example usage: `./manage.py lms unlink_enterprise_customer_learners -e <enterprise-uuid> --data-csv /path/file.csv`
`./manage.py lms unlink_enterprise_customer_learners -e <enterprise-uuid> --data-csv /path/file.csv --skip-unlink`

`add_arguments(parser)`

Entry point for subclassed commands to add custom arguments.

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

enterprise.management.commands.update_config_last_errored_at module

Backfill missing audit record foreign keys.

```
class enterprise.management.commands.update_config_last_errored_at.Command(stdout=None,
                                                                              stderr=None,
                                                                              no_color=False,
                                                                              force_color=False)
```

Bases: `IntegratedChannelCommandMixin`, `BaseCommand`

Management command which backfills missing audit record foreign keys.

handle(*args, **options)

Set error state for configurations.

help = '\n Set error state for configurations.\n '

update_config_last_errored_at()

For each audit record kind (learner and content), find all the records in batch, and lookup if they've had recent sync errors in the last day. If not, clear out the last_content_sync_errored_at value associated with the configuration.

enterprise.management.commands.update_role_assignments_with_customers module

Management command for updating enterprise user role assignments with appropriate `enterprise_customer` and `applies_to_all_contexts` values.

```
class enterprise.management.commands.update_role_assignments_with_customers.Command(stdout=None,
                                                                              stderr=None,
                                                                              no_color=False,
                                                                              force_color=False)
```

Bases: `BaseCommand`

Management command for creating enterprise role assignments with a foreign key to an `EnterpriseCustomer`, or an explicit boolean flag indicating that the assignment grants the role to the user for every `EnterpriseCustomer`.

add_arguments(parser)

Entry point for subclassed commands to add custom arguments.

handle(*args, **options)

Entry point for management command execution.

```
help = '\n Applies explicit enterprise customer context for enterprise admin,
learner, and catalog system role assignments.\n It also sets
`applies_to_all_contexts` to true for assignments of the
`enterprise_openedx_operator` role.\n Example usage:\n # Do a dry run for admin
assignments of some customer\n ./manage.py lms
update_role_assignments_with_customers --role=enterprise_admin --dry-run
--enterprise-customer-uuid=00000000-1111-2222-3333-444444444444\n # Do a real run
for all operators\n ./manage.py lms update_role_assignments_with_customers
--role=enterprise_openedx_operator\n # Process everything for everyone\n ./manage.py
lms update_role_assignments_with_customers\n '
```

Module contents

Module contents

enterprise.settings package

Submodules

enterprise.settings.test module

These settings are here to use during tests, because django requires them.

In a real-world use case, apps in this project are installed into other Django applications, so these settings will not be used.

`enterprise.settings.test.here(*args)`

Return the absolute path to a directory from this file.

`enterprise.settings.test.root(*args)`

Return the absolute path to some file from the project's root.

Module contents

enterprise.templatetags package

Submodules

enterprise.templatetags.enterprise module

Template tags and filters for the Enterprise application.

`enterprise.templatetags.enterprise.alert_messages(messages)`

Django template tag that returns an alert message.

Usage:

```
{% alert_messages messages %}
```

`enterprise.templatetags.enterprise.course_modal(context, course=None)`

Django template tag that returns course information to display in a modal.

You may pass in a particular course if you like. Otherwise, the modal will look for course context within the parent context.

Usage:

```
{% course_modal %} {% course_modal course %}
```

`enterprise.templatetags.enterprise.expand_button(value, href)`

Django template tag that returns a button used to expand/collapse a container.

You may pass in the ID of the container that this button controls, and a button text value.

Usage:

```
{% expand_button 'Click me!' '#id' %}
```

`enterprise templatetags.enterprise fa_icon(message_type)`

Django template tag that returns font awesome icon depending upon message type.

Usage:

```
{% fa_icon "success" %}
```

`enterprise templatetags.enterprise.link_to_modal(link_text, index, autoescape=True)`

Django template filter that returns an anchor with attributes useful for course modal selection.

General Usage:

```
{{ link_text|link_to_modal:index }}
```

Examples

```
{{ course_title|link_to_modal:forloop.counter0 }} {{ course_title|link_to_modal:3 }} {{ view_details_text|link_to_modal:0 }}
```

`enterprise templatetags.enterprise.only_safe_html(html_text)`

Django template filter that strips all HTML tags excepts those defined in ALLOWED_TAGS.

General Usage:

```
{{ html_text|only_safe_html }}
```

Module contents

8.1.2 Submodules

8.1.3 enterprise.apps module

Enterprise Django application initialization.

class `enterprise.apps.EnterpriseConfig(app_name, app_module)`

Bases: `AppConfig`

Configuration for the enterprise Django application.

property `auth_user_model`

Return User model for `django.contrib.auth`.

`backend_service_edx_oauth2_key = 'test_backend_oauth2_key'`

`backend_service_edx_oauth2_provider_url = 'http://localhost:8000/oauth2'`

`backend_service_edx_oauth2_secret = 'test_backend_oauth2_secret'`

`customer_success_email = 'customersuccess@edx.org'`

`enterprise_integrations_email = 'enterprise-integrations@edx.org'`

`name = 'enterprise'`

`ready()`

Perform other one-time initialization steps.

`valid_image_extensions = ['.png']`

`valid_max_image_size = 512`

8.1.4 enterprise.constants module

Enterprise Django application constants.

class enterprise.constants.CourseModes

Bases: `object`

Class to group modes that a course might have.

AUDIT = 'audit'

CREDIT = 'credit'

HONOR = 'honor'

NO_ID_PROFESSIONAL = 'no-id-professional'

PROFESSIONAL = 'professional'

UNPAID_EXECUTIVE_EDUCATION = 'unpaid-executive-education'

VERIFIED = 'verified'

class enterprise.constants.DefaultColors

Bases: `object`

Class to group the default branding color codes. These color codes originated in the Enterprise Learner Portal.

PRIMARY = '#2D494E'

SECONDARY = '#F2F0EF'

TERTIARY = '#D23228'

class enterprise.constants.FulfillmentTypes

Bases: `object`

CHOICES = [('license', 'License'), ('learner_credit', 'Learner credit'), ('coupon_code', 'Coupon code')]

COUPON_CODE = 'coupon_code'

LEARNER_CREDIT = 'learner_credit'

LICENSE = 'license'

enterprise.constants.json_serialized_course_modes()

Returns

serialized course modes.

8.1.5 enterprise.decorators module

Decorators for enterprise app.

`enterprise.decorators.deprecated(extra)`

Flag a method as deprecated.

Parameters

extra – Extra text you’d like to display after the default text.

`enterprise.decorators.disable_for_loaddata(signal_handler)`

Use this decorator to turn off signal handlers when loading fixture data.

Django docs instruct to avoid further changes to the DB if `raw=True` as it might not be in a consistent state. See <https://docs.djangoproject.com/en/dev/ref/signals/#post-save>

`enterprise.decorators.enterprise_login_required(view)`

View decorator for allowing authenticated user with valid enterprise UUID.

This decorator requires enterprise identifier as a parameter *enterprise_uuid*.

This decorator will throw 404 if no kwarg *enterprise_uuid* is provided to the decorated view .

If there is no enterprise in database against the kwarg *enterprise_uuid* or if the user is not authenticated then it will redirect the user to the enterprise-linked SSO login page.

Usage:

```
@enterprise_login_required()
def my_view(request, enterprise_uuid):
    # Some functionality ...

OR

class MyView(View):
    ...
    @method_decorator(enterprise_login_required)
    def get(self, request, enterprise_uuid):
        # Some functionality ...
```

`enterprise.decorators.force_fresh_session(view)`

View decorator which terminates stale TPA sessions.

This decorator forces the user to obtain a new session the first time they access the decorated view. This prevents TPA-authenticated users from hijacking the session of another user who may have been previously logged in using the same browser window.

This decorator should be used in conjunction with the `enterprise_login_required` decorator.

Usage:

```
@enterprise_login_required
@force_fresh_session()
def my_view(request, enterprise_uuid):
    # Some functionality ...

OR

class MyView(View):
```

(continues on next page)

(continued from previous page)

```
...
@method_decorator(enterprise_login_required)
@method_decorator(force_fresh_session)
def get(self, request, enterprise_uuid):
    # Some functionality ...
```

`enterprise.decorators.ignore_warning(warning)`

Ignore any emitted warnings from a function.

Parameters

warning – The category of warning to ignore.

`enterprise.decorators.null_decorator(func)`

Use this decorator to stub out decorators for testing.

If we're unable to import `social_core.pipeline.partial`, which is the case in our CI platform, we need to be able to wrap the function with something.

8.1.6 enterprise.errors module

Errors thrown by the APIs in the Enterprise application.

exception `enterprise.errors.AdminNotificationAPIRequestError`

Bases: `Exception`

An exception that represents an error when creating admin notification read status via the `NotificationReadApiClient`.

exception `enterprise.errors.CodesAPIRequestError`

Bases: `Exception`

There was a problem with a request to the Codes application's APIs.

exception `enterprise.errors.EnrollmentModificationException`

Bases: `Exception`

An exception that represents an error when modifying the state of an enrollment via the `EnrollmentApiClient`.

exception `enterprise.errors.LinkUserToEnterpriseError`

Bases: `Exception`

An error occurred while linking a user to an enterprise.

exception `enterprise.errors.UnlinkUserFromEnterpriseError`

Bases: `Exception`

An error occurred while unlinking a user from an enterprise.

8.1.7 enterprise.forms module

User-facing forms for the Enterprise app.

```
class enterprise.forms.EnterpriseLoginForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                           initial=None, error_class=<class
                                           'django.forms.utils.ErrorList'>, label_suffix=None,
                                           empty_permitted=False, field_order=None,
                                           use_required_attribute=None, renderer=None)
```

Bases: Form

Enterprise Slug Login Form.

```
base_fields = {'enterprise_slug': <django.forms.fields.CharField object>}
```

```
clean()
```

Validate POST data.

```
declared_fields = {'enterprise_slug': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```
class enterprise.forms.EnterpriseSelectionForm(*args, **kwargs)
```

Bases: Form

Enterprise Selection Form.

```
base_fields = {'enterprise': <django.forms.fields.ChoiceField object>,
               'success_url': <django.forms.fields.CharField object>}
```

```
clean()
```

Validate POST data.

```
declared_fields = {'enterprise': <django.forms.fields.ChoiceField object>,
               'success_url': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

8.1.8 enterprise.logging module

Logging utilities for Enterprise

```
class enterprise.logging.EnterpriseRequestIdLoggerAdapter(logger, extra)
```

Bases: `LoggerAdapter`

A utility for logging X-Request-ID information <https://docs.python.org/3/howto/logging-cookbook.html#using-loggeradapters-to-impart-contextual-information>

```
process(msg, kwargs)
```

Process the logging message and keyword arguments passed in to a logging call to insert contextual information. You can either manipulate the message itself, the keyword args or both. Return the message and kwargs modified (or not) to suit your needs.

Normally, you'll only need to override this one method in a `LoggerAdapter` subclass for your specific needs.

`enterprise.logging.getEnterpriseLogger(name)`

Get an Enterprise-ready logger

`enterprise.logging.get_request_id()`

Helper to get the request id - usually set via an X-Request-ID header

8.1.9 enterprise.messages module

Utility functions for interfacing with the Django messages framework.

`enterprise.messages.add_consent_declined_message(request, enterprise_customer, item)`

Add a message to the Django messages store indicating that the user has declined data sharing consent.

Parameters

- **request** (*HttpRequest*) – The current request.
- **enterprise_customer** (*EnterpriseCustomer*) – The EnterpriseCustomer associated with this request.
- **item** (*str*) – A string containing information about the item for which consent was declined.

`enterprise.messages.add_generic_error_message_with_code(request, error_code)`

Add message to request indicating that there was an issue processing request.

Parameters

- **request** – The current request.
- **error_code** – A string error code to be used to point devs to the spot in the code where this error occurred.

`enterprise.messages.add_missing_price_information_message(request, item)`

Add a message to the Django messages store indicating that we failed to retrieve price information about an item.

Parameters

- **request** – The current request.
- **item** – The item for which price information is missing. Example: a program title, or a course.

`enterprise.messages.add_unenrollable_item_message(request, item)`

Add a message to the Django message store indicating that the item (i.e. course run, program) is unenrollable.

Parameters

- **request** – The current request.
- **item** – The item that is unenrollable (i.e. a course run).

8.1.10 enterprise.middleware module

Middleware for enterprise app.

class enterprise.middleware.**EnterpriseLanguagePreferenceMiddleware**(*get_response*)

Bases: `MiddlewareMixin`

Middleware for enterprise language preference.

Ensures that, once set, a user's preferences are reflected in the page whenever they are logged in.

process_request(*request*)

Perform the following checks

1. Check that the user is authenticated and belongs to an enterprise customer.
2. Check that the enterprise customer has a language set via the *default_language* column on EnterpriseCustomer model.
3. Check that user has not set a language via its account settings page.

If all the above checks are satisfied then set `request._anonymous_user_cookie_lang` to the *default_language* of EnterpriseCustomer model instance. This attribute will later be used by the *LanguagePreferenceMiddleware* middleware for setting the user preference. Since, this middleware relies on *LanguagePreferenceMiddleware* so it must always be followed by *LanguagePreferenceMiddleware*. Otherwise, it will not work.

8.1.11 enterprise.models module

Database models for enterprise.

class enterprise.models.**AdminNotification**(*args, **kwargs)

Bases: `TimeStampedModel`

Model for Admin Notification.

exception **DoesNotExist**

Bases: `ObjectDoesNotExist`

exception **MultipleObjectsReturned**

Bases: `MultipleObjectsReturned`

admin_notification_filter

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

adminnotificationread_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

expiration_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_expiration_date(**, field=<django.db.models.fields.DateField: expiration_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_next_by_start_date(**, field=<django.db.models.fields.DateField: start_date>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_expiration_date(**, field=<django.db.models.fields.DateField: expiration_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

get_previous_by_start_date(**, field=<django.db.models.fields.DateField: start_date>, is_next=False, **kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

start_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

text

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `enterprise.models.AdminNotificationFilter(*args, **kwargs)`

Bases: `TimeStampedModel`

Model for Admin Notification Filters.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

filter

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

notification_filter

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`objects = <django.db.models.manager.Manager object>`

class `enterprise.models.AdminNotificationRead(*args, **kwargs)`

Bases: `TimeStampedModel`

Model for Admin Notification Read Status.

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

admin_notification

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

admin_notification_id

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_user_id

get_next_by_created(`*`, `field=<model_utils.fields.AutoCreatedField: created>`, `is_next=True`, `**kwargs`)

get_next_by_modified(`*`, `field=<model_utils.fields.AutoLastModifiedField: modified>`, `is_next=True`, `**kwargs`)

get_previous_by_created(`*`, `field=<model_utils.fields.AutoCreatedField: created>`, `is_next=False`, `**kwargs`)

get_previous_by_modified(`*`, `field=<model_utils.fields.AutoLastModifiedField: modified>`, `is_next=False`, `**kwargs`)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_read

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class enterprise.models.BulkCatalogQueryUpdateCommandConfiguration(*args, **kwargs)

Bases: ConfigurationModel

Manages configuration for a run of the cert_generation management command.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

arguments

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

change_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

changed_by

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

changed_by_id**enabled**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_change_date(* , field=<django.db.models.fields.DateTimeField: change_date>, is_next=True, **kwargs)

get_previous_by_change_date(* , field=<django.db.models.fields.DateTimeField: change_date>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.


```
class enterprise.models.ChatGPTResponse(*args, **kwargs)
```

Bases: `TimeStampedModel`

Stores ChatGPT prompts and their responses for each enterprise customer.

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

`LEARNER_ENGAGEMENT` = 'learner_engagement'

`LEARNER_PROGRESS` = 'learner_progress'

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

`PROMPT_TYPES` = [('learner_progress', 'Learner progress'), ('learner_engagement', 'Learner engagement')]

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=True, **kwargs)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=True, **kwargs)

classmethod **get_or_create**(*prompt*, *role*, *enterprise_customer*, *prompt_type*)

Get or create ChatGPT response against given prompt.

This method will first check and return the entry against the given prompt exist in the current table, if no such entry exists, it will call OpenAI API and save and return the entry.

Parameters

- **prompt** (*str*) – OpenAI prompt.
- **role** (*str*) – ChatGPT role to assume for the prompt.
- **enterprise_customer** (`EnterpriseCustomer`) – Enterprise customer UUID making the request.
- **prompt_type** (*str*) – Prompt type, e.g. `learner_progress` or `learner_engagement` etc.

Returns

Response against the given prompt.

Return type

(*str*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

get_prompt_type_display(**field*=<*django.db.models.fields.CharField: prompt_type*>)

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager* object>

prompt

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prompt_hash

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prompt_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

response

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(**args*, ***kwargs*)

Set the value of `prompt_hash` before saving.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `enterprise.models.EnrollmentNotificationEmailTemplate`(**args*, ***kwargs*)

Bases: `TimeStampedModel`

Store optional templates to use when emailing users about course enrollment events.

`BODY_HELP_TEXT` = 'Fill in a standard Django template that, when rendered, produces the email you want sent to newly-enrolled Enterprise Customer learners. The following variables may be available:\n`user_name`: A human-readable name for the person being emailed. Be sure to handle the case where this is not defined, as it may be missing in some cases. It may also be a username, if the learner hasn\'t configured their "real" name in the system. `organization_name`: The name of the organization sponsoring the enrollment. `enrolled_in`: Details of the course or program that was enrolled in. It may contain: `name`: The name of the enrollable item (e.g., "Demo Course"). `url`: A link to the homepage of the enrolled-in item. `branding`: A custom branding name for the enrolled-in item. For example, the branding of a MicroMasters program would be "MicroMasters". `start`: The date the enrolled-in item becomes available. Render this to text using the Django ``date`` template filter (see the Django documentation).`type`: Whether the enrolled-in item is a course, a program, or something else.'

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

SUBJECT_HELP_TEXT = 'Enter a string that can be used to generate a dynamic subject line for notification emails. The placeholder {course_name} will be replaced with the name of the course or program that was enrolled in.'

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

enterprise_customer_id

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

get_template_type_display(**, field=<django.db.models.fields.CharField: template_type>*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

html_template

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plaintext_template

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

render_all_templates(*kwargs*)

Render both templates and return both.

render_html_template(*kwargs*)

Render just the HTML template and return it as a string.

render_plaintext_template(*kwargs*)

Render just the plaintext template and return it as a string.

render_template(*template_text*, *kwargs*)

Create a template from the DB-backed text and render it.

save_without_historical_record(**args*, ***kwargs*)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

subject_line

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

template_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
template_type_choices = [('SELF_ENROLL', 'Self Enrollment Template'),  
('ADMIN_ENROLL', 'Admin Enrollment Template')]
```

class enterprise.models.**EnterpriseCatalogQuery**(**args*, ***kwargs*)

Bases: `TimeStampedModel`

Stores a re-usable catalog query.

This stored catalog query used in *EnterpriseCustomerCatalog* objects to build catalog's `content_filter` field. This is a saved instance of *content_filter* that can be re-used across different catalogs.

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

content_filter

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

delete(**args*, ***kwargs*)

Deletes this `EnterpriseCatalogQuery`.

enterprise_customer_catalogs

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=True, ***kwargs*)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=True, ***kwargs*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

include_exec_ed_2u_courses

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager object*>

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `enterprise.models.EnterpriseCourseEnrollment(*args, **kwargs)`

Bases: `TimeStampedModel`

Store information about the enrollment of a user in a course.

This model is the central source of truth for information about whether a particular user, linked to a particular `EnterpriseCustomer`, has been enrolled in a course, and is the repository for any other relevant metadata about such an enrollment.

Do not delete records of this model - there are downstream business reporting processes that rely them, even if the underlying `student.CourseEnrollment` record has been marked inactive/un-enrolled. As a consequence, the only way to determine if a given `EnterpriseCourseEnrollment` is currently active is to examine the `is_active` field of the associated `student.CourseEnrollment`.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

property audit_reporting_disabled

Specify whether audit track data reporting is disabled for this enrollment.

- If the enterprise customer associated with this enrollment enables audit track data reporting, simply return False.
- If the enterprise customer associated with this enrollment does not enable audit track data reporting, return True if we are dealing with an audit enrollment, and False otherwise.

Returns

True if audit track data reporting is disabled, False otherwise.

course_enrollment

Returns the `student.CourseEnrollment` associated with this enterprise course enrollment record.

course_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_user_id**classmethod get_enterprise_course_enrollment_id**(*user, course_id, enterprise_customer*)

Return the `EnterpriseCourseEnrollment` object for a given user in given `course_id`.

classmethod get_enterprise_uuids_with_user_and_course(*user_id, course_run_id, is_customer_active=None*)

Returns a list of UUID(s) for `EnterpriseCustomer`(s) that this enrollment links together with the `user_id` and `course_run_id`

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)**get_next_by_modified**(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)**get_previous_by_created**(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**field=<model_utils.fields.AutoLastModifiedField: modified>*,
is_next=False, ***kwargs*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property is_active

Returns True iff this enrollment is currently active.

property is_audit_enrollment

Specify whether the course enrollment associated with this EnterpriseCourseEnrollment is in audit mode.

Returns

Whether the course enrollment mode is of an audit type.

learnercreditenterpriseourseenrollment_enrollment_fulfillment

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

property license

Returns the license associated with this enterprise course enrollment if one exists.

licensedenterpriseourseenrollment_enrollment_fulfillment

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

property mode

Returns the mode of the student.CourseEnrollment associated with this enterprise course enrollment record.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <enterprise.models.EnterpriseCourseEnrollmentManager object>

save_without_historical_record(**args*, ***kwargs*)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

saved_for_later

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

source

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

source_id**unenrolled**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

unenrolled_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

with_additional_fields =

<enterprise.models.EnterpriseCourseEnrollmentWithAdditionalFieldsManager object>

class `enterprise.models.EnterpriseCourseEnrollmentManager(*args, **kwargs)`

Bases: `Manager`

Model manager for *EnterpriseCourseEnrollment*.

get_queryset()

Override to return only those enrollment records for which learner is linked to an enterprise.

class `enterprise.models.EnterpriseCourseEnrollmentWithAdditionalFieldsManager(*args, **kwargs)`

Bases: `Manager`

Model manager for *EnterpriseCourseEnrollment*.

get_queryset()

Override to return only those enrollment records for which learner is linked to an enterprise.

class `enterprise.models.EnterpriseCourseEntitlement(*args, **kwargs)`

Bases: `TimeStampedModel`

Store the information about the entitlement of an enterprise user for a course

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

course_entitlement

Returns the `CourseEntitlement` associated with this enterprise course entitlement record.

course_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_user_id

get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

learnercreditenterpriseenrollment_entitlement_fulfillment

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

licensedenterpriseenrollment_entitlement_fulfillment

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <enterprise.models.EnterpriseCourseEntitlementManager object>

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.EnterpriseCourseEntitlementManager(*args, **kwargs)

Bases: Manager

Model manager for *EnterpriseCourseEntitlement*.

get_queryset()

Override to return only those entitlement records for which learner is linked to an enterprise.

class enterprise.models.EnterpriseCustomer(*args, **kwargs)

Bases: TimeStampedModel

Enterprise Customer is an organization or a group of people that “consumes” courses.

Users associated with an Enterprise Customer take courses on the edX platform.

Enterprise Customer might be providing certain benefits to their members, like discounts to paid course enrollments, and also might request (or require) sharing learner results with them.

Fields

- **uuid** (*UUIDField, PRIMARY KEY*) – Enterprise Customer code - used to reference this Enterprise Customer in other parts of the system (SSO, ecommerce, analytics etc.).
- **name** (*django.db.models.CharField*) – Enterprise Customer name.
- **active** (*django.db.models.BooleanField*) – used to mark inactive Enterprise Customers - implements “soft delete” pattern.

AT_ENROLLMENT = 'at_enrollment'

DATA_SHARING_CONSENT_CHOICES = (('at_enrollment', 'At Enrollment'), ('externally_managed', 'Managed externally'))

exception DoesNotExist

Bases: *ObjectDoesNotExist*

EXTERNALLY_MANAGED = 'externally_managed'

exception MultipleObjectsReturned

Bases: *MultipleObjectsReturned*

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

active_customers = <enterprise.models.EnterpriseCustomerManager object>

auth_org_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

blackboardenterpriseconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

branding_configuration

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

canvasenterpriseconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

career_engagement_network_message

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

catalog_contains_course(course_run_id)

Determine if the specified course run is contained in enterprise customer catalogs.

Parameters

course_run_id (*str*) – The string ID of the course or course run in question

Returns

Whether the enterprise catalog includes the given course run.

Return type

bool

chat_gpt_prompts

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

clear_pending_registration(*email*, **course_ids*)

Clear pending enrollments for the user in the given courses.

Parameters

- **email** – The email address which may have previously been used.
- **course_ids** – An iterable containing any number of course IDs.

contact_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

contentmetadataitemtransmission_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

cornerstoneenterpriseconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

country

A descriptor for country fields on a model instance. Returns a `Country` when accessed so you can do things like:

```
>>> from people import Person
>>> person = Person.object.get(name='Chris')

>>> person.country.name
'New Zealand'

>>> person.country.flag
'/static/flags/nz.gif'
```

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

customer_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

customer_type_id**data_sharing_consent_page**

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

default_contract_discount

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

default_language

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property default_provider_idp

Return default_provider if associated with this enterprise customer.

degreed2enterprisecustomerconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

degreedenterprisecustomerconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`enable_academies`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_analytics_screen`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_audit_data_reporting`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_audit_enrollment`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_autocohorting`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_browse_and_request`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_career_engagement_network_on_learner_portal`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_data_sharing_consent`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_demo_data_for_analytics_and_lpr`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_executive_education_2U_fulfillment`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_generation_of_api_credentials`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_integrated_customer_learner_portal_search`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enable_learner_portal`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_learner_portal_offers

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_one_academy

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_pathways

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_code_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_learner_credit_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_lms_configurations_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_reporting_config_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_saml_configuration_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_subscription_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_programs

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_slug_login

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_universal_link

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property enables_audit_data_reporting

Determine whether the enterprise customer has enabled the ability to report/pass-back audit track data.

enforce_data_sharing_consent

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enforces_data_sharing_consent(*enforcement_location*)

Determine whether the enterprise customer enforce data sharing consent at the given point.

Parameters

- **enforcement_location** (*str*) – the point where to see data sharing consent state.
- **'externally_managed'** (*argument can either be 'at_enrollment' or*) –

enroll_user_pending_registration(*email, course_mode, *course_ids, **kwargs*)

Create pending enrollments for the user in any number of courses, which will take effect on registration.

Parameters

- **email** – The email address for the pending link to be created
- **course_mode** – The mode with which the eventual enrollment should be created
- ***course_ids** – An iterable containing any number of course IDs to eventually enroll the user in.
- **cohort** (*optional*) – name of cohort to assign

Returns

The PendingEnterpriseCustomerUser attached to the email address

enroll_user_pending_registration_with_status(*email, course_mode, *course_ids, **kwargs*)

Create pending enrollments for the user in any number of courses, which will take effect on registration.
Return a dictionary representing status of submitted enrollments.

Parameters

- **email** – The email address for the pending link to be created
- **course_mode** – The mode with which the eventual enrollment should be created
- ***course_ids** – An iterable containing any number of course IDs to eventually enroll the user in.
- **cohort** (*optional*) – name of cohort to assign

Returns

The PendingEnterpriseCustomerUser attached to the email address new_enrollments
(Dict): course ID keys and new enrollment status values.

enterprise_customer_catalogs

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

enterprise_customer_consent

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```


`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

property `enterprise_customer_identity_provider`

Returns the first instance from `EnterpriseCustomerIdentityProvider` relation.

`enterprise_customer_identity_providers`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`enterprise_customer_users`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`enterprise_enrollment_template`

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

`genericenterprisecustomerpluginconfiguration_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`get_country_display(*, field=<django_countries.fields.CountryField: country>)`

get_course_enrollment_url(*course_key*)

Return enterprise landing page url for the given course.

Parameters

course_key (*str*) – The course key for the course to be displayed.

Returns

Enterprise landing page url.

Return type

(*str*)

get_course_run_enrollment_url(*course_run_key*)

Return enterprise landing page url for the given course.

Parameters

course_run_key (*str*) – The course run id for the course to be displayed.

Returns

Enterprise landing page url.

Return type

(*str*)

get_data_sharing_consent_text_overrides(*published_only=True*)

Return DataSharingConsentTextOverrides associated with this instance.

get_default_language_display(**, field=<django.db.models.fields.CharField: default_language>*)

get_enforce_data_sharing_consent_display(**, field=<django.db.models.fields.CharField: enforce_data_sharing_consent>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

get_program_enrollment_url(*program_uuid*)

Return enterprise landing page url for the given program.

Parameters

program_uuid (*str*) – The program UUID.

Returns

Enterprise program landing page url.

Return type

(*str*)

get_tpa_hint(*tpa_hint_param=None*)

Parameters

tpa_hint_param – query param passed in the URL.

Returns

tpa_hint to redirect

groups

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

property has_identity_providers

Return True if there are any identity providers associated with this enterprise customer.

property has_multiple_idps

Return True if there are multiple identity providers associated with this enterprise customer.

property has_single_idp

Return True if there are exactly one identity provider associated with this enterprise customer.

hide_course_original_price

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

hide_labor_market_data

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

property identity_provider

Return the first identity provider id associated with this enterprise customer.

property identity_provider_ids

Return the identity provider Ids associated with this enterprise customer.

property identity_providers

Return the identity providers associated with this enterprise customer.

integratedchannelapirequestlogs_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

invite_keys

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

moodleenterprisecustomerconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

notify_enrolled_learners(*catalog_api_user, course_id, users, admin_enrollment=False, activation_links=None*)

Notify learners about a course in which they've been enrolled.

Parameters

- **catalog_api_user** – The user for calling the Catalog API
- **course_id** – The specific course the learners were enrolled in
- **users** – An iterable of the users (or pending users) who were enrolled
- **admin_enrollment** – Default False. Set to true if using bulk enrollment, for example. When true, we use the admin enrollment template instead.
- **activation_links** (*dict*) – a dictionary map of unactivated license user emails to license activation links

objects = <django.db.models.manager.Manager object>

pendingenterprisecustomeradminuser_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

pendingenterprisecustomeruser_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

replace_sensitive_sso_username

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

reply_to

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

reporting_configurations

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

property requests_data_sharing_consent

Determine whether the enterprise customer has enabled the data sharing consent request.

property safe_branding_configuration

Return the associated EnterpriseCustomerBrandingConfiguration object OR default branding config

This function should always be used to access the customer's branding_configuration and prevent uncaught RelatedObjectDoesNotExist exceptions when accessed directly.

sapsuccessfactorsenterpriseconfiguration_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

sender_alias

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property serialized

Return a serialized version of this customer.

site

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

site_id**slug**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sso_orchestration_records

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

property sync_learner_profile_data

Return the sync_learner_profile data flag for the identity provider associated with this enterprise customer.

Returns False if enterprise customer does not have any identity provider.

system_wide_role_assignments

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

toggle_universal_link(enable_universal_link)

Sets enable_universal_link

If there is no change to be made, return.

When `enable_universal_link` changes to:

- True: a new `EnterpriseCustomerInviteKey` is created, if total count is less than 100
- False: all `EnterpriseCustomerInviteKey` are deactivated

Parameters

`enable_universal_link` – new value

`uuid`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`xapilrsconfiguration`

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

class `enterprise.models.EnterpriseCustomerBrandingConfiguration(*args, **kwargs)`

Bases: `TimeStampedModel`

Model that keeps track of enterprise branding configurations e.g. enterprise customer logo.

Fields

- **`enterprise_customer`** (*ForeignKey[EnterpriseCustomer]*) – enterprise customer
- **`logo`** (*ImageField*) – enterprise customer image

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

`created`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enterprise_customer`

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

`enterprise_customer_id`

`get_next_by_created`(***, *field*=<model_utils.fields.AutoCreatedField: created>, *is_next*=True, ***kwargs*)

`get_next_by_modified`(***, *field*=<model_utils.fields.AutoLastModifiedField: modified>, *is_next*=True, ***kwargs*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

logo

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager object*>

primary_color

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property safe_logo_url

Returns an absolute URL for the branding configuration logo OR the platform logo absolute URL

secondary_color

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tertiary_color

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.**EnterpriseCustomerCatalog**(**args*, ***kwargs*)

Bases: *TimeStampedModel*

Store catalog information from course discovery specifically for Enterprises.

We use this model to consolidate course catalog information, which includes information about catalogs, courses, programs, and possibly more in the future, as the course discovery service evolves.

exception DoesNotExist

Bases: *ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *MultipleObjectsReturned*

contains_courses(*content_ids*)

Return True if this catalog contains the given courses else False.

The content_ids parameter should be a list containing course keys and/or course run ids.

contains_programs(*program_uuids*)

Return true if this catalog contains the given programs.

content_filter

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

content_filter_ids

Return the list of any content IDs specified in the catalog's content filter.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enabled_course_modes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_catalog_query

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_catalog_query_id**enterprise_customer**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id**enterprisecustomerreportingconfiguration_set**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_content_filter()

Return content filter of the linked catalog query otherwise content filter of catalog itself.

get_course(*course_key*)

Get all of the metadata for the given course.

Parameters

course_key (*str*) – The course key which identifies the course.

Returns

The course metadata.

Return type

dict

get_course_and_course_run(*course_run_id*)

Get course data and all of the metadata for the given course run.

Parameters

course_run_id (*str*) – The course run key which identifies the course run.

Returns

The course and course run metadata.

Return type

tuple(course, course_run)

Raises

ImproperlyConfigured – Missing or invalid catalog integration.

get_course_enrollment_url(*course_key*)

Return enterprise course enrollment page url with the catalog information for the given course.

Parameters

course_key (*str*) – The course key for the course to be displayed.

Returns

Enterprise landing page url.

Return type

(*str*)

get_course_run(*course_run_id*)

Get all of the metadata for the given course run.

Parameters

course_run_id (*str*) – The course run key which identifies the course run.

Returns

The course run metadata.

Return type

dict

get_course_run_enrollment_url(*course_run_key*)

Return enterprise course enrollment page url with the catalog information for the given course.

Parameters

course_run_key (*str*) – The course run id for the course to be displayed.

Returns

Enterprise landing page url.

Return type

(*str*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_paginated_content(*query_parameters*)

Return paginated discovery service search results without expired course runs.

Parameters

query_parameters (*dict*) – Additional query parameters to add to the search API call, e.g. page.

Returns

The paginated discovery service search results.

Return type

dict

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

get_program(*program_uuid*)

Get all of the metadata for the given program.

Parameters

program_uuid (*str*) – The program UUID which identifies the program.

Returns

The program metadata.

Return type

dict

get_program_enrollment_url(*program_uuid*)

Return enterprise program enrollment page url with the catalog information for the given program.

Parameters

program_uuid (*str*) – The program UUID.

Returns

Enterprise program landing page url.

Return type

(*str*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

publish_audit_enrollment_urls

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Saves this EnterpriseCatalogQuery.

Copies the content_filter of a related CatalogQuery into this instance's content_filter if syncing is allowed.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.EnterpriseCustomerIdentityProvider(*args, **kwargs)

Bases: `TimeStampedModel`

EnterpriseCustomerIdentityProvider is a One to Many relationship between Enterprise Customer and Identity Provider.

There should be a link between an enterprise customer and its Identity Provider. This relationship has following constraints:

1. An enterprise customer may or may not have an identity provider.
2. An enterprise customer can have more than one identity providers.
3. Enterprise customer site should match with identity provider's site. (i.e. same domain names)

Fields

- **enterprise_customer** (*ForeignKey[EnterpriseCustomer]*) – enterprise customer
- **provider_id** (*django.db.models.SlugField*) – The provider_id string of the identity provider.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

default_provider

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property identity_provider

Associated identity provider instance.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

provider_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property provider_name

Readable name for the identity provider.

property sync_learner_profile_data

Return bool indicating if data received from the identity provider should be synced to the edX profile.

class enterprise.models.EnterpriseCustomerInviteKey(*args, **kwargs)

Bases: TimeStampedModel, SoftDeletableModel

Stores an invite key used to link a learner to an enterprise.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id**expiration_date**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

classmethod `from_db(db, field_names, values)`

get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_expiration_date(* , field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=True, **kwargs)

get_next_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_expiration_date(* , field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=False, **kwargs)

get_previous_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property is_valid

Returns whether the key is still valid (non-expired and usage limit has not been reached).

linked_enterprise_customer_users

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Saves this `EnterpriseCustomerInviteKey`.

Prevents `is_active` from being updated once it's set to `False`.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

property usage_count

usage_limit

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.EnterpriseCustomerManager(*args, **kwargs)

Bases: `Manager`

Model manager for `EnterpriseCustomer` model.

Filters out inactive Enterprise Customers, otherwise works the same as default model manager.

get_queryset()

Return a new `QuerySet` object. Filters out inactive Enterprise Customers.

use_for_related_fields = False

class enterprise.models.EnterpriseCustomerReportingConfiguration(*args, **kwargs)

Bases: `TimeStampedModel`

The Enterprise's configuration for sending automated data reports securely via email to the Enterprise Admin.

`ALLOWED_NON_COMPRESSION_DATA_TYPES = ('catalog',)`

`DATA_TYPE_CATALOG = 'catalog'`

`DATA_TYPE_CHOICES = (('progress_v3', 'progress_v3'), ('catalog', 'catalog'), ('engagement', 'engagement'), ('grade', 'grade'), ('completion', 'completion'), ('course_structure', 'course_structure'))`

`DATA_TYPE_COMPLETION = 'completion'`

`DATA_TYPE_COURSE_STRUCTURE = 'course_structure'`

`DATA_TYPE_ENGAGEMENT = 'engagement'`

`DATA_TYPE_GRADE = 'grade'`

```
DATA_TYPE_PROGRESS_V3 = 'progress_v3'

DAYS_OF_WEEK = ((0, 'Monday'), (1, 'Tuesday'), (2, 'Wednesday'), (3, 'Thursday'),
(4, 'Friday'), (5, 'Saturday'), (6, 'Sunday'))

DELIVERY_METHOD_CHOICES = (('email', 'email'), ('sftp', 'sftp'))

DELIVERY_METHOD_EMAIL = 'email'

DELIVERY_METHOD_SFTP = 'sftp'

exception DoesNotExist
    Bases: ObjectDoesNotExist

FREQUENCY_CHOICES = (('daily', 'daily'), ('monthly', 'monthly'), ('weekly',
'weekly'))

FREQUENCY_TYPE_DAILY = 'daily'

FREQUENCY_TYPE_MONTHLY = 'monthly'

FREQUENCY_TYPE_WEEKLY = 'weekly'

MANUAL_REPORTS = ('grade', 'completion', 'course_structure')

exception MultipleObjectsReturned
    Bases: MultipleObjectsReturned

REPORT_TYPE_CHOICES = (('csv', 'csv'), ('json', 'json'))

REPORT_TYPE_CSV = 'csv'

REPORT_TYPE_JSON = 'json'
```

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

clean()

Override of clean method to perform additional validation on frequency, day_of_month/day_of week and compression.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

data_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

day_of_month

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

day_of_week

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

decrypted_password

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

decrypted_sftp_password

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

delivery_method

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_compression

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property encrypted_password

Return encrypted password as a string.

The data is encrypted in the DB at rest, but is unencrypted in the app when retrieved through the `decrypted_password` field. This method will encrypt the password again before sending.

property encrypted_sftp_password

Return encrypted SFTP password as a string.

The data is encrypted in the DB at rest, but is unencrypted in the app when retrieved through the `decrypted_password` field. This method will encrypt the password again before sending.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_catalogs

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

enterprise_customer_id

frequency

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_data_type_display(**,field=<django.db.models.fields.CharField: data_type>*)

get_day_of_week_display(**,field=<django.db.models.fields.SmallIntegerField: day_of_week>*)

get_delivery_method_display(**,field=<django.db.models.fields.CharField: delivery_method>*)

get_frequency_display(**,field=<django.db.models.fields.CharField: frequency>*)

get_next_by_created(**,field=<model_utils.fields.AutoCreatedField: created>,is_next=True,**kwargs*)

get_next_by_modified(**,field=<model_utils.fields.AutoLastModifiedField: modified>,is_next=True,**kwargs*)

get_previous_by_created(**,field=<model_utils.fields.AutoCreatedField: created>,is_next=False,**kwargs*)

get_previous_by_modified(**,field=<model_utils.fields.AutoLastModifiedField: modified>,is_next=False,**kwargs*)

get_report_type_display(**,field=<django.db.models.fields.CharField: report_type>*)

hour_of_day

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

include_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = *<django.db.models.manager.Manager object>*

pgp_encryption_key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

report_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sftp_file_path

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sftp_hostname

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sftp_port

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sftp_username

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

classmethod validate_compression(*enable_compression, data_type, delivery_method*)

Check enable_compression flag is set as expected

Parameters

- **enable_compression** (*bool*) – file compression flag
- **data_type** (*str*) – report type
- **delivery_method** (*str*) – delivery method for sending files

Returns

Validation Error

Return type

(dict)

validate_delivery_method(*create_report, delivery_method*)

Check delivery_method is changed or not while updating report

Parameters

- **create_report** (*str*) – report uuid
- **delivery_method** (*str*) – selected delivery method

Returns

Validation Error

Return type

(dict)

class enterprise.models.EnterpriseCustomerSsoConfiguration(**args, **kwargs*)

Bases: TimeStampedModel, SoftDeletableModel

Stores records of individual customer integrations with the SSO orchestration api.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

SAP_SUCCESS_FACTORS = 'sap_success_factors'

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

all_objects = <django.db.models.manager.Manager object>

```
base_saml_config_fields = ('uuid', 'metadata_url', 'metadata_xml', 'entity_id',
    'user_id_attribute', 'full_name_attribute', 'first_name_attribute',
    'last_name_attribute', 'email_attribute', 'username_attribute', 'country_attribute',
    'active', 'update_from_metadata')
```

configured_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

country_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

display_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-ToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id**entity_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

errored_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
fields_locked_while_configuring = ('metadata_url', 'metadata_xml', 'entity_id',
    'user_id_attribute', 'full_name_attribute', 'last_name_attribute',
    'email_attribute', 'username_attribute', 'country_attribute', 'active',
    'update_from_metadata', 'odata_api_timeout_interval', 'odata_api_root_url',
    'odata_company_id', 'sapsf_oauth_root_url', 'odata_api_request_timeout',
    'sapsf_private_key', 'odata_client_id', 'oauth_user_id')
```

first_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

full_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(**,field=<model_utils.fields.AutoCreatedField: created>,is_next=True,**kwargs*)

get_next_by_modified(**,field=<model_utils.fields.AutoLastModifiedField: modified>,is_next=True,**kwargs*)

get_previous_by_created(**,field=<model_utils.fields.AutoCreatedField: created>,is_next=False,**kwargs*)

get_previous_by_modified(**,field=<model_utils.fields.AutoLastModifiedField: modified>,is_next=False,**kwargs*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

identity_provider

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_pending_configuration()

Returns True if the configuration has been submitted but not completed configuration.

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

last_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

marked_authorized

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

metadata_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

metadata_xml

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

oauth_user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_api_request_timeout

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_api_root_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_api_timeout_interval

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_client_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_company_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
sap_config_fields = ('oauth_user_id', 'odata_api_request_timeout',  
'odata_api_root_url', 'odata_api_timeout_interval', 'odata_client_id',  
'odata_company_id', 'sapsf_oauth_root_url', 'sapsf_private_key')
```

sapsf_oauth_root_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sapsf_private_key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Override the save method to ensure that the configuration is locked once submitted and not completed by the SSO orchestration api.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

submit_for_configuration(updating_existing_record=False)

Submit the configuration to the SSO orchestration api.

submitted_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

update_from_metadata

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user_id_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

username_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

validated_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `enterprise.models.EnterpriseCustomerType(*args, **kwargs)`

Bases: `TimeStampedModel`

Enterprise Customer Types are used to differentiate Enterprise learners.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprisecustomer_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=True, **kwargs)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=True, **kwargs)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, **kwargs)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class enterprise.models.EnterpriseCustomerUser(*args, **kwargs)

Bases: TimeStampedModel

Model that keeps track of user - enterprise customer affinity.

Fields

- **enterprise_customer** (ForeignKey[EnterpriseCustomer]) – enterprise customer
- **user_id** (django.db.models.IntegerField) – user identifier

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

adminnotificationread_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

all_objects = <enterprise.models.EnterpriseCustomerUserManager object>

create_order_for_enrollment(course_run_id, discount_percentage, mode, sales_force_id)

Create an order on the Ecommerce side for tracking the course enrollment of a enterprise customer user.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property data_sharing_consent_records

Return the DataSharingConsent records associated with this EnterpriseCustomerUser.

Returns

The filtered DataSharingConsent QuerySet.

Return type

QuerySet (DataSharingConsent)

enroll(course_run_id, mode, cohort=None, source_slug=None, discount_percentage=0.0, sales_force_id=None)

Enroll a user into a course track, and register an enterprise course enrollment.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id**enterprise_enrollments**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

enterprise_entitlements

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

classmethod get_active_enterprise_users(user_id, enterprise_customer_uuids=None)

Return a queryset of all active enterprise users to which the given user is related. Or, if enterprise_customer_uuids is non-null, only the enterprise users related to the list of given enterprise_customer_uuids.

get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)**get_next_by_modified**(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)**get_previous_by_created**(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)**get_previous_by_modified**(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)**get_remote_id**(idp_id=None)

Retrieve the SSO provider's identifier for this user from the LMS Third Party API. In absence of idp_id, returns id from default idp

Parameters

idp_id (*str*) (*optional*) – If provided, idp resolution skipped and specified idp used to locate remote id.

Returns None if: * the user doesn't exist, or * the associated EnterpriseCustomer has no identity_provider, or * the remote identity is not found.

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

classmethod **inactivate_other_customers**(*user_id*, *enterprise_customer*)

Mark as inactive all the enterprise customers of given user except the given enterprise_customer.

invite_key

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

invite_key_id**is_relinkable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

linked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

memberships

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property name

Return linked user's name.

objects = <enterprise.models.EnterpriseCustomerUserManager object>

save(*args, **kwargs)

Override to handle creation of EnterpriseCustomerUser records.

This is needed because of soft deletion of EnterpriseCustomerUser records. This will handle all of get_or_create/update_or_create/create methods.

By default, when an EnterpriseCustomerUser record is created/updated with *active=True*, all other linked records for the user will be marked as *active=False*. To disable this side effect, update set *should_inactivate_other_customers=False* on an EnterpriseCustomerUser instance.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

should_inactivate_other_customers

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

unenroll(course_run_id)

Unenroll a user from a course track.

update_session(request)

Update the session of a request for this learner.

property user

Return User associated with this instance.

Return `django.contrib.auth.models.User` instance associated with this `EnterpriseCustomerUser` instance via email.

property user_email

Return linked user email.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property username

Return linked user's username.

class enterprise.models.EnterpriseCustomerUserManager(*args, **kwargs)

Bases: Manager

Model manager for `EnterpriseCustomerUser` entity.

This class should contain methods that create, modify or query `EnterpriseCustomerUser` entities.

get(kwargs)**

Overridden get method to return the first element in case of learner with multiple enterprises.

Raises EnterpriseCustomerUser.DoesNotExist if no records are found.

get_link_by_email(user_email, enterprise_customer)

Return link by email and enterprise_customer

get_queryset()

Return linked or unlinked learners based on how the manager is created.

link_user(*enterprise_customer, user_email*)

Link user email to Enterprise Customer.

If `django.contrib.auth.models.User` instance with specified email does not exist, `PendingEnterpriseCustomerUser` instance is created instead.

unlink_user(*enterprise_customer, user_email, is_relinkable=True*)

Unlink user email from Enterprise Customer.

If `django.contrib.auth.models.User` instance with specified email does not exist, `PendingEnterpriseCustomerUser` instance is deleted instead.

Raises `EnterpriseCustomerUser.DoesNotExist` if instance of `django.contrib.auth.models.User` with specified email exists and corresponding `EnterpriseCustomerUser` instance does not.

Raises `PendingEnterpriseCustomerUser.DoesNotExist` exception if instance of `django.contrib.auth.models.User` with specified email exists and corresponding `PendingEnterpriseCustomerUser` instance does not.

class `enterprise.models.EnterpriseEnrollmentSource(*args, **kwargs)`

Bases: `TimeStampedModel`

Define a Name and Source for all Enterprise Enrollment Sources.

API = 'enterprise_api'

CUSTOMER_ADMIN = 'customer_admin'

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

ENROLLMENT_TASK = 'enrollment_task'

ENROLLMENT_URL = 'enrollment_url'

MANAGEMENT_COMMAND = 'management_command'

MANUAL = 'manual'

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

OFFER_REDEMPTION = 'offer_redemption'

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprisecourseenrollment_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)
```

```
get_next_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True,
    **kwargs)
```

```
get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False,
    **kwargs)
```

```
get_previous_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>,
    is_next=False, **kwargs)
```

```
classmethod get_source(source_slug)
```

Retrieve the source based on the Slug provided.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

pendingenrollment_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

slug

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.EnterpriseFeatureRole(*args, **kwargs)
```

Bases: UserRole

Enterprise-specific feature role definitions.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprisefeatureuserroleassignment_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=True, ***kwargs*)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=True, ***kwargs*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager* object>

class `enterprise.models.EnterpriseFeatureUserRoleAssignment(*args, **kwargs)`

Bases: `UserRoleAssignment`

Model to map users to an `EnterpriseFeatureRole`.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

applies_to_all_contexts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property enterprise_customer_uuids

Get the enterprise customer uuids linked to the user.

get_context()

Returns a non-empty list of enterprise customer uuid strings to which `self.user` is linked, or `None` if the user is not linked to any `EnterpriseCustomer`.

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=*True*, ***kwargs*)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=*True*, ***kwargs*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=*False*, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=*False*, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager object*>

role

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

role_class

alias of `EnterpriseFeatureRole`

role_id**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

user_id

class enterprise.models.EnterpriseFulfillmentSource(*args, **kwargs)

Bases: `TimeStampedModel`

Base class for enterprise subsidy fulfillments

class Meta

Bases: `object`

abstract = `False`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

classmethod enrollments_for_user(*enterprise_customer_user*)

Returns a QuerySet of subsidy based enrollment records for a particular user, along with their associated (hydrated) user, enterprise enrollments, and customer object.

enterprise_course_enrollment

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

enterprise_course_enrollment_id

enterprise_course_entitlement

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

enterprise_course_entitlement_id

property enterprise_customer_user

fulfillment_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_fulfillment_type_display(**, field=<django.db.models.fields.CharField: fulfillment_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

is_revoked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

reactivate(**kwargs)

Idempotently reactivates this enterprise fulfillment source.

revoke()

Marks this object as revoked and marks the associated EnterpriseCourseEnrollment as “saved for later”. This object and the associated EnterpriseCourseEnrollment are both saved.

TODO: revoke entitlements as well?

save(*args, **kwargs)

Overriding the save method in order to make sure that modified field is updated even if it is not given as a parameter to the update field argument.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you’re doing before you use this method.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.EnterpriseGroup(*args, **kwargs)

Bases: `TimeStampedModel`, `SoftDeletableModel`

Enterprise Group model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

applies_to_all_contexts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id

get_all_learners(*user_query=None, sort_by=None, desc_order=False, fetch_removed=False, pending_users_only=False*)

Returns all users associated with the group, whether the group specifies the entire org else all associated membership records.

Params:

q (optional): filter the returned members list by user email and name with a provided sub-string
 sort_by (optional): specify how the list of returned members should be ordered. Supported sorting values are *memberDetails*, *memberStatus*, and *recentAction*. Ordering can be reversed by supplying a - at the beginning of the sorting value ie *-memberStatus*.

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

members

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.**EnterpriseGroupMembership**(*args, **kwargs)

Bases: `TimeStampedModel`, `SoftDeletableModel`

Enterprise Group Membership model

exception **DoesNotExist**

Bases: `ObjectDoesNotExist`

exception **MultipleObjectsReturned**

Bases: `MultipleObjectsReturned`

activated_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

clean(*args, **kwargs)

Ensure that records added via Django Admin have matching customer records between learner and group.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_user_id

get_next_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(*args, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(*args, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

get_status_display(*args, field=<django.db.models.fields.CharField: status>)

group

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

group_id

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

member_email

Return the email associated with the member

membership_user

Return the user record associated with the membership, defaulting to enterprise_customer_user and falling back on obj.pending_enterprise_customer_user

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

pending_enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

pending_enterprise_customer_user_id**recent_action**

Return the timestamp of the most recent action relating to the membership

removed_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

status

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `enterprise.models.EnterpriseRoleAssignmentContextMixin`

Bases: `object`

Mixin for RoleAssignment models related to enterprises.

DEPRECATED: Not removing since it's referenced in a migration (0001_squashed_0092_auto_20200312_1650).

class `enterprise.models.HistoricalEnrollmentNotificationEmailTemplate`(*id, created, modified, plaintext_template, html_template, subject_line, template_type, enterprise_customer, history_id, history_date, history_change_reason, history_type, history_user*)

Bases: `HistoricalChanges, Model`

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id

static `get_default_history_user(instance)`

Returns the user specified by `get_user` method for manually creating historical objects

`get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)`

`get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)`

`get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)`

`get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)`

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_history_date(**field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

get_template_type_display(**field*=<*django.db.models.fields.CharField: template_type*>)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

history_user_id

html_template

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance

instance_type

alias of `EnrollmentNotificationEmailTemplate`

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

plaintext_template

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

subject_line

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

template_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCourseEnrollment(id, created, modified, course_id,
                                                         saved_for_later, unenrolled,
                                                         unenrolled_at,
                                                         enterprise_customer_user, source,
                                                         history_id, history_date,
                                                         history_change_reason,
                                                         history_type, history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

course_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_user_id

static get_default_history_user(*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display(***, *field*=<*django.db.models.fields.CharField: history_type*>)

get_next_by_created(***, *field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=True, ***kwargs*)

get_next_by_history_date(***, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next*=True, ***kwargs*)

get_next_by_modified(***, *field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=True, ***kwargs*)

get_previous_by_created(***, *field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next*=False, ***kwargs*)

get_previous_by_history_date(***, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next*=False, ***kwargs*)

get_previous_by_modified(***, *field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next*=False, ***kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of `EnterpriseCourseEnrollment`

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

saved_for_later

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

source

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

source_id**unenrolled**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

unenrolled_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCourseEntitlement(id, created, modified, uuid,
                                                            course_uuid,
                                                            enterprise_customer_user,
                                                            history_id, history_date,
                                                            history_change_reason,
                                                            history_type, history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

course_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_user_id**static get_default_history_user(instance)**

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)**get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)****get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)****get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)****get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)****get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)****get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)****history_change_reason**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of *EnterpriseCourseEntitlement*

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCustomer(created, modified, uuid, name, slug,  
                                                    auth_org_id, active, country,  
                                                    hide_course_original_price,  
                                                    enable_data_sharing_consent,  
                                                    enforce_data_sharing_consent,  
                                                    enable_audit_enrollment,  
                                                    enable_audit_data_reporting,  
                                                    replace_sensitive_sso_username,  
                                                    enable_autocohorting,  
                                                    enable_portal_code_management_screen,  
                                                    enable_portal_reporting_config_screen,  
                                                    enable_portal_subscription_management_screen,  
                                                    enable_portal_saml_configuration_screen,  
                                                    enable_universal_link,  
                                                    enable_browse_and_request,  
                                                    enable_learner_portal,  
                                                    enable_learner_portal_offers,  
                                                    enable_portal_learner_credit_management_screen,  
                                                    hide_labor_market_data,  
                                                    enable_integrated_customer_learner_portal_search,  
                                                    en-  
                                                    able_career_engagement_network_on_learner_portal,  
                                                    enable_pathways,  
                                                    enable_programs,  
                                                    enable_academies,  
                                                    enable_one_academy,  
                                                    enable_analytics_screen,  
                                                    enable_portal_lms_configurations_screen,  
                                                    enable_slug_login,  
                                                    enable_executive_education_2U_fulfillment,  
                                                    enable_demo_data_for_analytics_and_lpr,  
                                                    contact_email,  
                                                    default_contract_discount,  
                                                    default_language,  
                                                    sender_alias,  
                                                    reply_to,  
                                                    enable_generation_of_api_credentials,  
                                                    career_engagement_network_message,  
                                                    site,  
                                                    customer_type,  
                                                    history_id,  
                                                    history_date,  
                                                    history_change_reason,  
                                                    history_type,  
                                                    history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

auth_org_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

career_engagement_network_message

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

contact_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

country

A descriptor for country fields on a model instance. Returns a Country when accessed so you can do things like:

```
>>> from people import Person
>>> person = Person.object.get(name='Chris')

>>> person.country.name
'New Zealand'

>>> person.country.flag
'/static/flags/nz.gif'
```

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

customer_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

customer_type_id**default_contract_discount**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

default_language

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_academies

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_analytics_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_audit_data_reporting

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_audit_enrollment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_autocohorting

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_browse_and_request

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_career_engagement_network_on_learner_portal

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_data_sharing_consent

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_demo_data_for_analytics_and_lpr

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_executive_education_2U_fulfillment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_generation_of_api_credentials

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_integrated_customer_learner_portal_search

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_learner_portal

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_learner_portal_offers

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_one_academy

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_pathways

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_code_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_learner_credit_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_lms_configurations_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_reporting_config_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_saml_configuration_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_portal_subscription_management_screen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_programs

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_slug_login

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enable_universal_link

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enforce_data_sharing_consent

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_country_display(**, field=<django_countries.fields.CountryField: country>*)

static get_default_history_user(*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_default_language_display(**, field=<django.db.models.fields.CharField: default_language>*)

get_enforce_data_sharing_consent_display(**, field=<django.db.models.fields.CharField: enforce_data_sharing_consent>*)

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

hide_course_original_price

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

hide_labor_market_data

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

property instance

instance_type

alias of *EnterpriseCustomer*

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

replace_sensitive_sso_username

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

reply_to

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

revert_url()

URL for this change in the default admin site.

sender_alias

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

site

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

site_id

slug

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCustomerCatalog(created, modified, uuid, title,
                                                            content_filter, enabled_course_modes,
                                                            publish_audit_enrollment_urls,
                                                            enterprise_customer,
                                                            enterprise_catalog_query, history_id,
                                                            history_date, history_change_reason,
                                                            history_type, history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

content_filter

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enabled_course_modes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_catalog_query

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_catalog_query_id**enterprise_customer**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id**static get_default_history_user(instance)**

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)**get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)****get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)****get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)****get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)****get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)**

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

property instance

instance_type

alias of *EnterpriseCustomerCatalog*

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

publish_audit_enrollment_urls

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

revert_url()

URL for this change in the default admin site.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCustomerInviteKey(created, modified, is_removed, uuid,  
                                                         usage_limit, expiration_date,  
                                                         is_active, enterprise_customer,  
                                                         history_id, history_date,  
                                                         history_change_reason,  
                                                         history_type, history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id**expiration_date**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static get_default_history_user(*instance*)

Returns the user specified by `get_user` method for manually creating historical objects

```
get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)
```

```
get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)
```

```
get_next_by_expiration_date(*, field=<django.db.models.fields.DateTimeField: expiration_date>,  
                             is_next=True, **kwargs)
```

```
get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>,  
                          is_next=True, **kwargs)
```

```
get_next_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True,
                      **kwargs)
```

```
get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False,
                        **kwargs)
```

```
get_previous_by_expiration_date(* , field=<django.db.models.fields.DateTimeField: expiration_date>,
                                is_next=False, **kwargs)
```

```
get_previous_by_history_date(* , field=<django.db.models.fields.DateTimeField: history_date>,
                             is_next=False, **kwargs)
```

```
get_previous_by_modified(* , field=<model_utils.fields.AutoLastModifiedField: modified>,
                          is_next=False, **kwargs)
```

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

property instance

instance_type

alias of *EnterpriseCustomerInviteKey*

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

usage_limit

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration(created, modified,
                                                                    is_removed,
                                                                    display_name, uuid,
                                                                    active, identity_provider,
                                                                    metadata_url,
                                                                    metadata_xml, entity_id,
                                                                    update_from_metadata,
                                                                    user_id_attribute,
                                                                    full_name_attribute,
                                                                    first_name_attribute,
                                                                    last_name_attribute,
                                                                    email_attribute,
                                                                    username_attribute,
                                                                    country_attribute,
                                                                    submitted_at,
                                                                    configured_at,
                                                                    validated_at, errored_at,
                                                                    marked_authorized,
                                                                    odata_api_timeout_interval,
                                                                    odata_api_root_url,
                                                                    odata_company_id,
                                                                    sapsf_oauth_root_url,
                                                                    odata_api_request_timeout,
                                                                    sapsf_private_key,
                                                                    odata_client_id,
                                                                    oauth_user_id,
                                                                    enterprise_customer,
                                                                    history_id, history_date,
                                                                    history_change_reason,
                                                                    history_type,
                                                                    history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

configured_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

country_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

display_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id**entity_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

errored_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

first_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

full_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static `get_default_history_user(instance)`

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

history_user_id

identity_provider

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance

instance_type

alias of *EnterpriseCustomerSsoConfiguration*

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

last_name_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

marked_authorized

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

metadata_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

metadata_xml

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

oauth_user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

odata_api_request_timeout

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_api_root_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_api_timeout_interval

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_client_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

odata_company_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

sapsf_oauth_root_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sapsf_private_key

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

submitted_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

update_from_metadata

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user_id_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

username_attribute

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

validated_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseCustomerUser(id, created, modified, user_id, active,  
                                                         linked, is_relinkable,  
                                                         should_inactivate_other_customers,  
                                                         enterprise_customer, invite_key,  
                                                         history_id, history_date,  
                                                         history_change_reason, history_type,  
                                                         history_user)
```

Bases: `HistoricalChanges, Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id**static get_default_history_user**(*instance*)

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)**get_next_by_created**(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)**get_next_by_history_date**(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)**get_next_by_modified**(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)**get_previous_by_created**(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)**get_previous_by_history_date**(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)**get_previous_by_modified**(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)**history_change_reason**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object**history_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of *EnterpriseCustomerUser*

invite_key

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

invite_key_id**is_relinkable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

linked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

should_inactivate_other_customers

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseGroup(created, modified, is_removed, uuid, name,  
                                              applies_to_all_contexts, enterprise_customer,  
                                              history_id, history_date, history_change_reason,  
                                              history_type, history_user)
```

Bases: `HistoricalChanges, Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

applies_to_all_contexts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id**static get_default_history_user(*instance*)**

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(*, *field*=<django.db.models.fields.CharField: history_type>)**get_next_by_created(*, *field*=<model_utils.fields.AutoCreatedField: created>, *is_next*=True, **kwargs)****get_next_by_history_date(*, *field*=<django.db.models.fields.DateTimeField: history_date>, *is_next*=True, **kwargs)****get_next_by_modified(*, *field*=<model_utils.fields.AutoLastModifiedField: modified>, *is_next*=True, **kwargs)****get_previous_by_created(*, *field*=<model_utils.fields.AutoCreatedField: created>, *is_next*=False, **kwargs)**

get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)

get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

property instance

instance_type

alias of *EnterpriseGroup*

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalEnterpriseGroupMembership(created, modified, is_removed, uuid,
    activated_at, status, removed_at,
    group, enterprise_customer_user,
    pending_enterprise_customer_user,
    history_id, history_date,
    history_change_reason, history_type,
    history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

activated_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_user_id**static get_default_history_user(instance)**

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)**get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)****get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)**

```
get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False,
                      **kwargs)
```

```
get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>,
                           is_next=False, **kwargs)
```

```
get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>,
                        is_next=False, **kwargs)
```

```
get_status_display(*, field=<django.db.models.fields.CharField: status>)
```

group

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-ToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

group_id

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-ToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

property instance**instance_type**

alias of [EnterpriseGroupMembership](#)

is_removed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

pending_enterprise_customer_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

pending_enterprise_customer_user_id**property prev_record**

Get the previous history record for the instance. *None* if first.

removed_at

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

revert_url()

URL for this change in the default admin site.

status

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment(id, created,  
                                                                           modified, uuid,  
                                                                           fulfillment_type,  
                                                                           is_revoked,  
                                                                           transaction_id,  
                                                                           enter-  
                                                                           prise_course_entitlement,  
                                                                           enter-  
                                                                           prise_course_enrollment,  
                                                                           history_id,  
                                                                           history_date, his-  
                                                                           tory_change_reason,  
                                                                           history_type,  
                                                                           history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_course_enrollment

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_course_enrollment_id

enterprise_course_entitlement

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_course_entitlement_id

fulfillment_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static get_default_history_user(*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_fulfillment_type_display(**, field=<django.db.models.fields.CharField: fulfillment_type>*)

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance

instance_type

alias of `LearnerCreditEnterpriseCourseEnrollment`

is_revoked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

transaction_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment(id, created, modified, uuid,
                                                                    fulfillment_type,
                                                                    is_revoked, license_uuid,
                                                                    enter-
                                                                    prise_course_entitlement,
                                                                    enter-
                                                                    prise_course_enrollment,
                                                                    history_id, history_date,
                                                                    history_change_reason,
                                                                    history_type,
                                                                    history_user)
```

Bases: `HistoricalChanges, Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_course_enrollment

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_course_enrollment_id**enterprise_course_entitlement**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_course_entitlement_id**fulfillment_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static get_default_history_user(*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_fulfillment_type_display(**, field=<django.db.models.fields.CharField: fulfillment_type>*)

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object**history_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

history_user_id**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of `LicensedEnterpriseCourseEnrollment`

is_revoked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

license_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = `<django.db.models.manager.Manager object>`

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalPendingEnrollment(id, created, modified, course_id, course_mode,  
                                                    cohort_name, discount_percentage,  
                                                    sales_force_id, license_uuid, user, source,  
                                                    history_id, history_date, history_change_reason,  
                                                    history_type, history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

cohort_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

course_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

course_mode

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

discount_percentage

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static get_default_history_user(*instance*)

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(*, *field*=<django.db.models.fields.CharField: history_type>)**get_next_by_created(*, *field*=<model_utils.fields.AutoCreatedField: created>, *is_next*=True, **kwargs)****get_next_by_history_date(*, *field*=<django.db.models.fields.DateTimeField: history_date>, *is_next*=True, **kwargs)****get_next_by_modified(*, *field*=<model_utils.fields.AutoLastModifiedField: modified>, *is_next*=True, **kwargs)****get_previous_by_created(*, *field*=<model_utils.fields.AutoCreatedField: created>, *is_next*=False, **kwargs)****get_previous_by_history_date(*, *field*=<django.db.models.fields.DateTimeField: history_date>, *is_next*=False, **kwargs)**

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance

instance_type

alias of *PendingEnrollment*

license_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

sales_force_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

source

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

source_id**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

```
class enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser(id, created, modified,
                                                                    user_email,
                                                                    enterprise_customer,
                                                                    history_id, history_date,
                                                                    history_change_reason,
                                                                    history_type,
                                                                    history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id

static get_default_history_user(instance)

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)

get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)

get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)

get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance

instance_type

alias of `PendingEnterpriseCustomerAdminUser`

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

user_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalPendingEnterpriseCustomerUser(id, created, modified, user_email,
                                                                enterprise_customer, history_id,
                                                                history_date,
                                                                history_change_reason,
                                                                history_type, history_user)
```

Bases: `HistoricalChanges`, `Model`

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id

static get_default_history_user(*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of *PendingEnterpriseCustomerUser*

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

user_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment(id,
                                                                    created, modified, ap-
                                                                    plies_to_all_contexts,
                                                                    user, role,
                                                                    enterprise_customer,
                                                                    history_id,
                                                                    history_date, his-
                                                                    tory_change_reason,
                                                                    history_type,
                                                                    history_user)
```

Bases: HistoricalChanges, Model

exception DoesNotExist

Bases: *ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *MultipleObjectsReturned*

applies_to_all_contexts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id

static get_default_history_user(*instance*)

Returns the user specified by `get_user` method for manually creating historical objects

get_history_type_display(**, field=<django.db.models.fields.CharField: history_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_history_date(**, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history_change_reason

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_object

history_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

history_user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

history_user_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property instance**instance_type**

alias of `SystemWideEnterpriseUserRoleAssignment`

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property next_record

Get the next history record for the instance. *None* if last.

objects = <django.db.models.manager.Manager object>

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

role

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

role_id**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

class enterprise.models.LearnerCreditEnterpriseCourseEnrollment(*args, **kwargs)

Bases: `EnterpriseFulfillmentSource`

An Enterprise Course Enrollment that is enrolled via a transaction ID.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_course_enrollment

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

enterprise_course_enrollment_id**enterprise_course_entitlement**

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

enterprise_course_entitlement_id**fulfillment_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`get_fulfillment_type_display(*, field=<django.db.models.fields.CharField: fulfillment_type>)`

`get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)`

`get_next_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)`

`get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)`

`get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)`

`history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_revoked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

reactivate(*transaction_id=None, **kwargs*)

Idmpotently reactivates this LearnerCreditEnterpriseCourseEnrollment.

Parameters

transaction_id (*str*) – New ledgered transaction UUID to associate with this learner credit fulfillment.

transaction_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.LicensedEnterpriseCourseEnrollment(**args, **kwargs*)

Bases: [EnterpriseFulfillmentSource](#)

An Enterprise Course Enrollment that is enrolled via a license.

exception DoesNotExist

Bases: [ObjectDoesNotExist](#)

exception MultipleObjectsReturned

Bases: [MultipleObjectsReturned](#)

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_course_enrollment

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Restaurant.place is a ForwardOneToOneDescriptor instance.

enterprise_course_enrollment_id**enterprise_course_entitlement**

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Restaurant.place is a ForwardOneToOneDescriptor instance.

enterprise_course_entitlement_id

fulfillment_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_fulfillment_type_display(**, field=<django.db.models.fields.CharField: fulfillment_type>*)

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_revoked

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

license_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.PendingEnrollment(*args, **kwargs)

Bases: `TimeStampedModel`

Track future enrollments for PendingEnterpriseCustomerUser.

Store a course ID, an intended enrollment mode, and a link to a PendingEnterpriseCustomerUser; when the PendingEnterpriseCustomerUser is converted to a full EnterpriseCustomerUser, API calls will be made to enroll the newly-created user in whatever courses have been added.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

cohort_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

course_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

course_mode

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

discount_percentage

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next=True*, ***kwargs*)

get_next_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next=True*, ***kwargs*)

get_previous_by_created(**field*=<*model_utils.fields.AutoCreatedField: created*>, *is_next=False*, ***kwargs*)

get_previous_by_modified(**field*=<*model_utils.fields.AutoLastModifiedField: modified*>, *is_next=False*, ***kwargs*)

history = <*django.db.models.manager.HistoryManagerFromHistoricalQuerySet object*>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

license_uuid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager object*>

sales_force_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save_without_historical_record(**args*, ***kwargs*)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

source

Accessor to the related object on the forward side of a many-to-one or one-to-one (via *ForwardOneToOneDescriptor* subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

source_id

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

class enterprise.models.PendingEnterpriseCustomerAdminUser(*args, **kwargs)

Bases: TimeStampedModel

Model for pending enterprise admin users.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

admin_registration_url

Returns a URL to be used by a pending enterprise admin user to register their account.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id

get_next_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(*args, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

```
get_previous_by_modified(*, field=<model_utils.fields.AutoLastModifiedField: modified>,
                           is_next=False, **kwargs)
```

```
history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
objects = <django.db.models.manager.Manager object>
```

```
save_without_historical_record(*args, **kwargs)
```

Save model without saving a historical record

Make sure you know what you’re doing before you use this method.

user_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class enterprise.models.PendingEnterpriseCustomerUser(*args, **kwargs)
```

Bases: `TimeStampedModel`

Model that stores “future members” of enterprise customer.

Fields

- **enterprise_customer** (`ForeignKey[EnterpriseCustomer]`) – enterprise customer
- **user_email** (`django.db.models.EmailField`) – user email

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enterprise_customer_id

fulfill_pending_course_enrollments(*enterprise_customer_user*)

Enrolls a newly created EnterpriseCustomerUser in any courses attached to their PendingEnterpriseCustomerUser record.

Parameters

enterprise_customer_user – a EnterpriseCustomerUser instance

fulfill_pending_group_memberships(*enterprise_customer_user*)

Updates any membership records associated with a new created enterprise customer user object.

Parameters

enterprise_customer_user – a EnterpriseCustomerUser instance

get_next_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs*)

get_next_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs*)

get_previous_by_modified(**, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs*)

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

link_pending_enterprise_user(*user, is_user_created*)

Link a PendingEnterpriseCustomerUser to the appropriate EnterpriseCustomer by creating or updating an EnterpriseCustomerUser record.

Parameters

- **is_user_created** – a boolean whether the User instance was created or updated
- **user** – a User instance

Returns: an EnterpriseCustomerUser instance

memberships

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

pendingenrollment_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

user_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class enterprise.models.SystemWideEnterpriseRole(*args, **kwargs)

Bases: `UserRole`

System wide user role definitions specific to Enterprise.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(*args, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(*args, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(*args, field=<model_utils.fields.AutoLastModifiedField: modified>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

system_wide_role_assignments

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

class enterprise.models.SystemWideEnterpriseUserRoleAssignment(*args, **kwargs)

Bases: UserRoleAssignment

Model to map users to a SystemWideEnterpriseRole.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

applies_to_all_contexts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enterprise_customer

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

enterprise_customer_id

classmethod get_assignments(user, role_names=None)

Return an iterator of (rolename, [enterprise customer uuids]) for the given user (and maybe role_names).

Differs from super().get_assignments(...) in that it yields (role name, customer uuid list) pairs such that the first item in the customer uuid list for each role corresponds to the currently *active* EnterpriseCustomerUser for the user.

The resulting generated pairs are sorted by role name, and within role_name, by (active, customer uuid). For example:


```
(‘enterprise_admin’, [‘active-enterprise-uuid’, ‘inactive-enterprise-uuid’, ‘other-inactive-enterprise-uuid’]) (‘enterprise_learner’, [‘active-enterprise-uuid’, ‘inactive-enterprise-uuid’]), (‘enterprise_openedx_operator’, [‘*’])
```

get_context()

Return a non-empty list of contexts for which `self.user` is assigned `self.role`.

classmethod get_distinct_assignments_by_role_name(*user*, *role_names=None*)

Returns a mapping of role names to sets of enterprise customer uuids for which the user is assigned that role.

get_next_by_created(***, *field=<model_utils.fields.AutoCreatedField: created>*, *is_next=True*, ***kwargs*)**get_next_by_modified**(***, *field=<model_utils.fields.AutoLastModifiedField: modified>*, *is_next=True*, ***kwargs*)**get_previous_by_created**(***, *field=<model_utils.fields.AutoCreatedField: created>*, *is_next=False*, ***kwargs*)**get_previous_by_modified**(***, *field=<model_utils.fields.AutoLastModifiedField: modified>*, *is_next=False*, ***kwargs*)**has_access_to_all_contexts()**

Returns `true` if the role for this assignment is `ENTERPRISE_OPERATOR_ROLE`, or if `applies_to_all_contexts` is `true`; returns `false` otherwise.

history = <django.db.models.manager.HistoryManagerFromHistoricalQuerySet object>**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modified

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>**role**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

role_id**save**(**args*, ***kwargs*)

Overriding the save method in order to make sure that modified field is updated even if it is not given as a parameter to the update field argument.

save_without_historical_record(**args*, ***kwargs*)

Save model without saving a historical record

Make sure you know what you’re doing before you use this method.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

`enterprise.models.get_default_customer_type()`

Get default enterprise customer type id to use when creating a new EnterpriseCustomer model.

`enterprise.models.get_default_site()`

Get default site id to use when creating a new EnterpriseCustomer model. The default value depending on what environment the person is in. In production, it should be 'courses.edx.org'. In stage it should be 'courses.stage.edx.org'.

8.1.12 enterprise.roles_api module

Python API for doing CRUD operations on roles and user role assignments.

`enterprise.roles_api.admin_role()`

Return the enterprise admin role.

`enterprise.roles_api.assign_admin_role(user, enterprise_customer=None, applies_to_all_contexts=False)`

Assigns the given user the *enterprise_admin* role in the given customer.

`enterprise.roles_api.assign_learner_role(user, enterprise_customer=None, applies_to_all_contexts=False)`

Assigns the given user the *enterprise_learner* role in the given customer.

`enterprise.roles_api.catalog_admin_role()`

Returns the enterprise catalog admin role.

`enterprise.roles_api.delete_admin_role_assignment(user, enterprise_customer=None)`

Deletes the admin role assignment for the given user in the given enterprise customer. If *enterprise_customer* is null, will delete *every* admin role assignment for this user.

`enterprise.roles_api.delete_learner_role_assignment(user, enterprise_customer=None)`

Deletes the learner role assignment for the given user in the given enterprise customer. If *enterprise_customer* is null, will delete *every* learner role assignment for this user.

`enterprise.roles_api.delete_role_assignment(user, role, enterprise_customer=None)`

Deletes the given role assignment for the given user in the given enterprise customer. If *enterprise_customer* is null, will delete *every* role assignment for this user.

`enterprise.roles_api.get_or_create_system_wide_role(role_name)`

Gets or creates the *SystemWideEnterpriseRole* with the given name. Caches the result for 60 minutes in the Django cache. Returns the role object.

`enterprise.roles_api.learner_role()`

Returns the enterprise learner role.

`enterprise.roles_api.openedx_operator_role()`

Returns the enterprise openedx operator role.

`enterprise.roles_api.roles_by_name()`

Returns a mapping of system wide roles by name.

8.1.13 enterprise.rules module

Rules needed to restrict access to the enterprise data api.

8.1.14 enterprise.signals module

Django signal handlers.

`enterprise.signals.assign_or_delete_enterprise_learner_role(sender, instance, **kwargs)`

Assign or delete enterprise_learner role for EnterpriseCustomerUser when created or updated.

The enterprise_learner role is assigned when a new EnterpriseCustomerUser record is initially created and removed when a EnterpriseCustomerUser record is updated and unlinked (i.e., soft delete - see ENT-2538).

`enterprise.signals.create_enterprise_enrollment_receiver(sender, instance, **kwargs)`

Watches for post_save signal for creates on the CourseEnrollment table.

Spin off an async task to generate an EnterpriseCourseEnrollment if appropriate.

`enterprise.signals.create_pending_enterprise_admin_user(sender, instance, **kwargs)`

Creates a PendingEnterpriseCustomerUser when a PendingEnterpriseCustomerAdminUser is created.

`enterprise.signals.default_content_filter(sender, instance, **kwargs)`

Set default value for *EnterpriseCustomerCatalog.content_filter* if not already set.

`enterprise.signals.delete_enterprise_admin_role_assignment(sender, instance, **kwargs)`

Delete the associated enterprise admin role assignment record when deleting an EnterpriseCustomerUser record.

`enterprise.signals.delete_enterprise_catalog_data(sender, instance, **kwargs)`

Send deletions of Enterprise Catalogs to the Enterprise Catalog Service.

`enterprise.signals.delete_enterprise_learner_role_assignment(sender, instance, **kwargs)`

Delete the associated enterprise learner role assignment record when deleting an EnterpriseCustomerUser record.

`enterprise.signals.delete_pending_enterprise_admin_user(sender, instance, **kwargs)`

Deletes a PendingEnterpriseCustomerUser when its associated PendingEnterpriseCustomerAdminUser is removed.

`enterprise.signals.enterprise_unenrollment_receiver(sender, **kwargs)`

Mark the EnterpriseCourseEnrollment object as unenrolled when a user unenrolls from a course.

`enterprise.signals.generate_default_orchestration_record_display_name(sender, instance, **kwargs)`

Ensure that the display_name field is populated with a default value if it is not provided while creating.

`enterprise.signals.handle_user_post_save(sender, **kwargs)`

Handle User model changes. Context: This signal runs any time a user logs in, including b2c users.

Steps:

1. Check for existing PendingEnterpriseCustomerUser(s) for user's email. If one or more exists, create an EnterpriseCustomerUser record for each which will ensure the user has the "enterprise_learner" role.
2. When we get a new EnterpriseCustomerUser record (or an existing record if one existed), check if the PendingEnterpriseCustomerUser has any pending course enrollments. If so, enroll the user in these courses.
3. Delete the PendingEnterpriseCustomerUser record as its no longer needed.
4. Using the newly created EnterpriseCustomerUser (or an existing record if one existed), check if there is a PendingEnterpriseCustomerAdminUser. If so, ensure the user has the "enterprise_admin" role.

`enterprise.signals.save_logo_file(sender, instance, **kwargs)`

Now that the object is instantiated and instance.id exists, save the image at correct path and re-save the model.

`enterprise.signals.skip_saving_logo_file(sender, instance, **kwargs)`

To avoid saving the logo image at an incorrect path, skip saving it.

`enterprise.signals.update_enterprise_catalog_data(sender, instance, **kwargs)`

Send data changes to Enterprise Catalogs to the Enterprise Catalog Service.

Additionally sends a request to update the catalog's metadata from discovery, and index any relevant content for Algolia.

`enterprise.signals.update_enterprise_catalog_query(sender, instance, **kwargs)`

Sync data changes from Enterprise Catalog Query to the Enterprise Customer Catalog.

`enterprise.signals.update_lang_pref_of_all_learners(sender, instance, **kwargs)`

Update the language preference of all the learners belonging to the enterprise customer. Set the language preference to the value enterprise customer has used as the *default_language*.

`enterprise.signals.update_learner_language_preference(sender, instance, created, **kwargs)`

Update the language preference of the learner. Set the language preference to the value enterprise customer has used as the *default_language*.

8.1.15 enterprise.tasks module

Django tasks.

`enterprise.tasks.enterprise_course_enrollment_model()`

Returns the EnterpriseCourseEnrollment class. This function is needed to avoid circular ref issues when model classes call tasks in this module.

`enterprise.tasks.enterprise_customer_user_model()`

Returns the EnterpriseCustomerUser class. This function is needed to avoid circular ref issues when model classes call tasks in this module.

`enterprise.tasks.enterprise_enrollment_source_model()`

Returns the EnterpriseEnrollmentSource class. This function is needed to avoid circular ref issues when model classes call tasks in this module.

`enterprise.tasks.enterprise_group_membership_model()`

Returns the EnterpriseGroupMembership class. This function is needed to avoid circular ref issues when model classes call tasks in this module.

`enterprise.tasks.pending_enterprise_customer_user_model()`

Returns the PendingEnterpriseCustomerUser class. This function is needed to avoid circular ref issues when model classes call tasks in this module.

8.1.16 enterprise.toggles module

Waffle toggles for enterprise features within the LMS.

`enterprise.toggles.enterprise_features()`

Returns a dict of enterprise Waffle-based feature flags.

`enterprise.toggles.enterprise_groups_v1()`

Returns whether the enterprise groups feature flag is enabled.

`enterprise.toggles.feature_prequery_search_suggestions()`

Returns whether the prequery search suggestion feature flag is enabled.

`enterprise.toggles.top_down_assignment_real_time_lcm()`

Returns whether top-down assignment and real time LCM feature flag is enabled.

8.1.17 enterprise.tpa_pipeline module

Module provides elements to be used in third-party auth pipeline.

`enterprise.tpa_pipeline.get_default_idp_user_social_auth(enterprise_customer, user=None, user_idp_id=None)`

Return social auth entry of user for given enterprise default IDP.

Parameters

- **user** (*User*) – user object
- **enterprise_customer** (*EnterpriseCustomer*) – Instance of the enterprise customer.
- **user_idp_id** (*str*) – User id of user in third party LMS

`enterprise.tpa_pipeline.get_enterprise_customer_for_running_pipeline(request, pipeline)`

Get the EnterpriseCustomer associated with a running pipeline.

`enterprise.tpa_pipeline.get_enterprise_customer_for_sso(sso_provider_id)`

Get the EnterpriseCustomer object tied to an identity provider.

`enterprise.tpa_pipeline.get_sso_provider(request, pipeline)`

Helper method to retrieve the sso provider ID from either a user's SSO login request

`enterprise.tpa_pipeline.get_user_from_social_auth(tpa_providers, user_id, enterprise_customer)`

Find the LMS user from the LMS model *UserSocialAuth*.

Parameters

- **tpa_providers** (*third_party_auth.provider*) – list of third party auth provider objects
- **user_id** (*str*) – User id of user in third party LMS
- **enterprise_customer** (*EnterpriseCustomer*) – Instance of the enterprise customer.

`enterprise.tpa_pipeline.get_user_social_auth(user, enterprise_customer)`

Return social auth entry of user for given enterprise.

Parameters

- **user** (*User*) – user object
- **enterprise_customer** (*EnterpriseCustomer*) – Instance of the enterprise customer.

`enterprise.tpa_pipeline.handle_enterprise_logistration(backend, user, **kwargs)`

Perform the linking of user in the process of logging to the Enterprise Customer.

Parameters

- **backend** – The class handling the SSO interaction (SAML, OAuth, etc)
- **user** – The user object in the process of being logged in with
- ****kwargs** – Any remaining pipeline variables

`enterprise.tpa_pipeline.handle_redirect_after_social_auth_login(backend, user)`

Change the redirect url if user has more than 1 EnterpriseCustomer associations.

Parameters

- **backend** (*User*) – social auth backend object
- **user** (*User*) – user object

`enterprise.tpa_pipeline.select_enterprise_page_as_redirect_url(strategy)`

Change the redirect url for the user to enterprise selection page.

`enterprise.tpa_pipeline.validate_provider_config(enterprise_customer, sso_provider_id)`

Helper method to ensure that a customer's provider config is validated

8.1.18 enterprise.urls module

URLs for enterprise.

8.1.19 enterprise.utils module

Utility functions for enterprise app.

exception `enterprise.utils.CourseCatalogApiError`

Bases: `Exception`

Exception to raise when we we received data from Course Catalog but it contained an error.

exception `enterprise.utils.CourseEnrollmentDowngradeError`

Bases: `Exception`

Exception to raise when an enrollment attempts to enroll the user in an unpaid mode when they are in a paid mode.

exception `enterprise.utils.CourseEnrollmentPermissionError`

Bases: `Exception`

Exception to raise when an enterprise attempts to use enrollment features it's not configured to use.

exception `enterprise.utils.NotConnectedToOpenEdX(*args, **kwargs)`

Bases: `Exception`

Exception to raise when not connected to OpenEdX.

In general, this exception shouldn't be raised, because this package is designed to be installed directly inside an existing OpenEdX platform.

```
class enterprise.utils.ValidationMessages
```

Bases: `object`

Namespace class for validation messages.

```
BOTH_COURSE_FIELDS_SPECIFIED = 'Either "CSV bulk upload" or a singular course ID may
be used for manual enrollments, but not both together.'
```

```
BOTH_FIELDS_SPECIFIED = 'Either "Email or Username" or "CSV bulk upload" must be
specified, but both were.'
```

```
BULK_LINK_FAILED = 'Error: Learners could not be added. Correct the following
errors.'
```

```
COURSE_MODE_INVALID_FOR_COURSE = 'Enrollment track {course_mode} is not available
for course {course_id}.'
```

```
COURSE_NOT_EXIST_IN_CATALOG = "Course ID {course_id} doesn't exist in Enterprise
Customer's Catalog"
```

```
COURSE_WITHOUT_COURSE_MODE = 'Select a course enrollment track for the given
course(s).'
```

```
INVALID_CHANNEL_WORKER = 'Enterprise channel worker user with the username
"{channel_worker_username}" was not found.'
```

```
INVALID_COURSE_ID = 'Could not retrieve details for the course ID {course_id}.
Specify a valid ID.'
```

```
INVALID_DISCOUNT = 'Discount percentage should be from 0 to 100.'
```

```
INVALID_EMAIL = '{argument} does not appear to be a valid email address.'
```

```
INVALID_EMAIL_OR_USERNAME = '{argument} does not appear to be a valid email address
or known username'
```

```
INVALID_ENCODING = "Unable to parse CSV file. Please make sure it is a CSV 'utf-8'
encoded file."
```

```
MISSING_EXPECTED_COLUMNS = 'Expected a CSV file with [{expected_columns}] columns,
but found [{actual_columns}] columns instead.'
```

```
MISSING_REASON = 'Reason field is required but was not filled.'
```

```
NO_FIELDS_SPECIFIED = 'Either "Email or Username" or "CSV bulk upload" must be
specified, but neither were.'
```

```
USER_ALREADY_REGISTERED = 'User with email address {email} is already registered
with Enterprise Customer {ec_name}'
```

```
USER_NOT_EXIST = "User with email address {email} doesn't exist."
```

```
USER_NOT_LINKED = 'User is not linked with Enterprise Customer'
```

```
enterprise.utils.batch(iterable, batch_size=1)
```

Break up an iterable into equal-sized batches.

Parameters

- **iterable** (e.g. *list*) – an iterable to batch
- **batch_size** (*int*) – the size of each batch. Defaults to 1.

Returns

iterates through each batch of an iterable

Return type

generator

`enterprise.utils.camelCase(string)`

Convert a string to camelCase.

`enterprise.utils.clean_html_for_template_rendering(text)`

Given html text that will be rendered as a variable in a template, strip out characters that impact rendering.

Parameters

text (*str*) – The text to clean.

Returns

The cleaned text.

Return type

(*str*)

`enterprise.utils.convert_to_snake(string)`

Helper method to convert strings to snake case.

`enterprise.utils.customer_admin_enroll_user(enterprise_customer, user, course_mode, course_id, enrollment_source=None)`

For use with bulk enrollment, or any use case of admin enrolling a user

Enroll a single user in a course using a particular course mode, indicating it's a customer_admin enrolling a user (such as bulk enrollment)

Parameters

- **enterprise_customer** – The EnterpriseCustomer model object which is sponsoring the enrollment
- **user** – The user model object who needs to be enrolled in the course
- **course_mode** – The string representation of the mode with which the enrollment should be created
- **course_id** – An opaque course_id to enroll in

Returns

Whether or not enrollment succeeded for the course specified

Return type

succeeded (Boolean)

`enterprise.utils.customer_admin_enroll_user_with_status(enterprise_customer, user, course_mode, course_id, enrollment_source=None, license_uuid=None, transaction_id=None, force_enrollment=False)`

For use with bulk enrollment, or any use case of admin enrolling a user

Enroll a single user in a course using a particular course mode, indicating it's a customer_admin enrolling a user (such as bulk enrollment). Return a status based on whether the enrollment existed before attempting to enroll.

TODO: The `enroll_user` function above, used by Django admin, for example, should also be rewired to use this new ability, but for now it still uses enrollment client.

Parameters

- **enterprise_customer** – The EnterpriseCustomer model object which is sponsoring the enrollment
- **user** – The user model object who needs to be enrolled in the course
- **course_mode** – The string representation of the mode with which the enrollment should be created
- **course_id** – An opaque course_id to enroll in
- **enrollment_source** – Source of enrollment, used for tracking enrollments
- **license_uuid** – UUID of associated license with the enrollment, used to create a mapping between licenses and enrollments
- **transaction_id** – UUID of associated ledgered transaction if this enrollment is subsidized via learner credit 2.

Returns

Whether or not the enrollment succeeded for the course specified created (Boolean): Whether or not the enrollment existed prior to calling method

Return type

succeeded (Boolean)

`enterprise.utils.delete_data_sharing_consent(course_id, customer_uuid, user_email)`

Delete the DSC records from the DB for given learner, course and customer, also its cache.

`enterprise.utils.discovery_query_url(content_filter, html_format=True)`

Return discovery url for preview.

`enterprise.utils.enroll_subsidy_users_in_courses(enterprise_customer, subsidy_users_info, discount=100.0)`

Takes a list of licensed learner data and enrolls each learner in the requested courses.

Parameters

- **enterprise_customer** – The EnterpriseCustomer (object) which is sponsoring the enrollment
- **subsidy_users_info** (*list of dict*) – Each element contains information necessary to create a enterprise enrollment from a subsidy for a specific learner in a specified course run. Required fields:
 - `'user_id'` OR `'email'`: Either unique identifier describing the user to enroll.
 - `'course_run_key'`: The course to enroll into.
 - `'course_mode'`: The course mode.
 - `'license_uuid'` OR `'transaction_id'`: ID of either accepted form of subsidy.
 - `'force_enrollment'` (bool, optional): Enroll user even enrollment deadline is expired (default False).

Example:

```

licensed_users_info: [
  {
    'email': 'newuser@test.com',
    'course_run_key': 'course-v1:edX+DemoX+Demo_Course',
    'course_mode': 'verified',
    'license_uuid': '5b77bdbade7b4fcb838f8111b68e18ae'
  },
  {
    'email': 'newuser2@test.com',
    'course_run_key': 'course-v2:edX+FunX+Fun_Course',
    'course_mode': 'unpaid-executive-education',
    'transaction_id': '84kdbdbade7b4fcb838f8asjke8e18ae',
  },
  {
    'user_id': 1234,
    'course_run_key': 'course-v1:edX+SadX+Sad_Course',
    'course_mode': 'unpaid-executive-education',
    'transaction_id': '3a5312d722564db0a16e3d81f53a3718',
  },
]

```

- **discount** – (int) the discount offered to the learner for their enrollment. Subscription based enrollments default to 100

Expected Return Values:

```

Results: {
  successes:
    [{ 'user_id': <lms_user_id>, 'email': <email>, 'course_run_key': <key>,
    ↪ 'user': <user object> } ... ],
  pending:
    [{ 'user_id': <lms_user_id>, 'email': <email>, 'course_run_key': <key>,
    ↪ 'user': <user object> } ... ],
  failures:
    [{ 'user_id': <lms_user_id>, 'email': <email>, 'course_run_key': <key> } ...
    ↪ ]
}

```

`enterprise.utils.enroll_user(enterprise_customer, user, course_mode, *course_ids, **kwargs)`

Enroll a single user in any number of courses using a particular course mode.

Parameters

- **enterprise_customer** – The EnterpriseCustomer model object which is sponsoring the enrollment
- **user** – The user model object who needs to be enrolled in the course
- **course_mode** – The string representation of the mode with which the enrollment should be created
- ***course_ids** – An iterable containing any number of course IDs to eventually enroll the user in.
- **kwargs** – Should contain `enrollment_client` if it's already been instantiated and should be passed in.

Returns

Whether or not enrollment succeeded for all courses specified

Return type

Boolean

```
enterprise.utils.enroll_users_in_course(enterprise_customer, course_id, course_mode, emails,
                                       enrollment_requester=None, enrollment_reason=None,
                                       discount=0.0, sales_force_id=None)
```

Enroll existing users in a course, and create a pending enrollment for nonexisting users.

Parameters

- **enterprise_customer** – The EnterpriseCustomer which is sponsoring the enrollment
- **course_id** (*str*) – The unique identifier of the course in which we’re enrolling
- **course_mode** (*str*) – The mode with which we’re enrolling in the course
- **emails** – An iterable of email addresses which need to be enrolled
- **enrollment_requester** (*User*) – Admin user who is requesting the enrollment.
- **enrollment_reason** (*str*) – A reason for enrollment.
- **discount** (*Decimal*) – Percentage discount for enrollment.
- **sales_force_id** (*str*) – Salesforce opportunity id.

Returns

A list of users who were successfully enrolled in the course.

pending: A list of PendingEnterpriseCustomerUsers who were successfully linked and had pending enrollments created for them in the database.

failures: A list of users who could not be enrolled in the course.

Return type

successes

```
enterprise.utils.enterprise_course_enrollment_model()
```

Returns the EnterpriseCourseEnrollment class.

```
enterprise.utils.enterprise_customer_invite_key_model()
```

Returns the EnterpriseCustomerInviteKey class.

```
enterprise.utils.enterprise_customer_model()
```

Returns the EnterpriseCustomer class.

```
enterprise.utils.enterprise_customer_user_model()
```

Returns the EnterpriseCustomerUser class.

```
enterprise.utils.enterprise_enrollment_source_model()
```

Returns the EnterpriseEnrollmentSource class.

```
enterprise.utils.filter_audit_course_modes(enterprise_customer, course_modes)
```

Filter audit course modes out if the enterprise customer has not enabled the ‘Enable audit enrollment’ flag.

Parameters

- **enterprise_customer** – The EnterpriseCustomer that the enrollment was created using.
- **course_modes** – iterable with dictionaries containing a required ‘mode’ key

`enterprise.utils.find_enroll_email_template(enterprise_customer, template_type)`

Find email template from the template database represented by EnrollmentNotificationEmailTemplate model.

Parameters

- **enterprise_customer** (-) – the customer model
- **template_type** (-) – type of template to fetch, must be one of: `enterprise.utils.SELF_ENROLL_EMAIL_TEMPLATE_TYPE`, or `enterprise.utils.ADMIN_ENROLL_EMAIL_TEMPLATE_TYPE`

Returns

Customer specific template if found. Default template for the given type if found. None if neither default template, nor per customer template found.

`enterprise.utils.format_price(price, currency='$')`

Format the price to have the appropriate currency and digits..

Parameters

- **price** – The price amount.
- **currency** – The currency for the price.

Returns

A formatted price string, i.e. '\$10', '\$10.52'.

`enterprise.utils.get_active_course_runs(course, users_all_enrolled_courses)`

Return active course runs (user is enrolled in) of the given course.

This function will return the `course_runs` of 'course' which have active enrollment by looking into 'users_all_enrolled_courses'

`enterprise.utils.get_advertised_course_run(course)`

Find the advertised course run for a given course :param course: course dict :type course: dict

Returns

a `course_run` or None

Return type

`dict`

`enterprise.utils.get_advertised_or_closest_course_run(content_metadata_item)`

Returns advertised course run of a course. If the advertised run does not exist, it looks for the closest run.

`enterprise.utils.get_all_field_names(model, excluded=None)`

Return all fields' names from a model. Filter out the field names present in *excluded*.

According to [Django documentation](#), `get_all_field_names` should become some monstrosity with chained iterable ternary nested in a list comprehension. For now, a simpler version of iterating over fields and getting their names work, but we might have to switch to full version in future.

`enterprise.utils.get_best_mode_from_course_key(course_key)`

Helper method to retrieve a list of enrollments for a given course and select the one most applicable to enroll an enterprise learner in.

`enterprise.utils.get_cache_key(**kwargs)`

Wrapper method on `edx_django_utils.get_cache_key` utility.

`enterprise.utils.get_catalog_admin_url(catalog_id)`

Get url to catalog details admin page.

Parameters

catalog_id (*int*) – Catalog id for which to return catalog details url.

Returns

URL pointing to catalog details admin page for the give catalog id.

Example

```
>>> get_catalog_admin_url_template(2)
"http://localhost:18381/admin/catalogs/catalog/2/change/"
```

`enterprise.utils.get_catalog_admin_url_template(mode='change')`

Get template of catalog admin url.

URL template will contain a placeholder '{catalog_id}' for catalog id. :param mode e.g. change/add.:

Returns

A string containing template for catalog url.

Example

```
>>> get_catalog_admin_url_template('change')
"http://localhost:18381/admin/catalogs/catalog/{catalog_id}/change/"
```

`enterprise.utils.get_closest_course_run(course_runs)`

Return course run with start date closest to now.

`enterprise.utils.get_configuration_value(val_name, default=None, **kwargs)`

Get a configuration value, or fall back to default if it doesn't exist.

Also takes a *type* argument to guide which particular upstream method to use when trying to retrieve a value. Current types include:

- *url* to specifically get a URL.

`enterprise.utils.get_configuration_value_for_site(site, key, default=None)`

Get the site configuration value for a key, unless a site configuration does not exist for that site.

Useful for testing when no Site Configuration exists in edx-enterprise or if a site in LMS doesn't have a configuration tied to it.

Parameters

- **site** – A Site model object
- **key** – The name of the value to retrieve
- **default** – The default response if there's no key in site config or settings

Returns

The value located at that key in the site configuration or settings file.

`enterprise.utils.get_content_metadata_item_id(content_metadata_item)`

Return the unique identifier given a content metadata item dictionary.

`enterprise.utils.get_course_run_duration_info(course_run)`

Return course run's duration(str) info.

`enterprise.utils.get_course_run_start(course_run, default=None)`

Return the given course run's start date as a datetime.

`enterprise.utils.get_course_track_selection_url(course_run, query_parameters)`

Return track selection url for the given course.

Parameters

- **course_run** (*dict*) – A dictionary containing course run metadata.
- **query_parameters** (*dict*) – A dictionary containing query parameters to be added to course selection url.

Raises

(**KeyError**) – Raised when course run dict does not have 'key' key.

Returns

Course track selection url.

Return type

(str)

`enterprise.utils.get_create_ent_enrollment(course_id, enterprise_customer_user,
enterprise_enrollment_source, license_uuid=None)`

Get or Create the Enterprise Course Enrollment.

If license_uuid present, will also create a LicensedEnterpriseCourseEnrollment record.

`enterprise.utils.get_current_course_run(course, users_active_course_runs)`

Return the current course run on the following conditions:

- If user has active course runs (already enrolled) then return course run with closest start date

Otherwise it will check the following logic:

- Course run is enrollable (see `is_course_run_enrollable`)
- Course run has a verified seat and the upgrade deadline has not expired.
- If no enrollable/upgradeable course runs, then select all the course runs.
- After filtering the course runs checks whether the filtered course run is about to close or not if yes then return the next course run or the current one.

`enterprise.utils.get_default_catalog_content_filter()`

Return default enterprise customer catalog content filter.

`enterprise.utils.get_default_invite_key_expiration_date()`

Returns the default expiration date for an invite key.

The default expiration date is 365 days from the current date.

`enterprise.utils.get_duration_of_course_or_courserun(content_metadata_item)`

Returns duration start, end dates given a piece of content_metadata item If course item, extracts start, end dates of closest course run based on current timestamp

Returns

in days or 0 start: start field of closest course run item, or None end: end field of closest course run item, or None

Return type

duration

`enterprise.utils.get_ecommerce_worker_user()`

Return the user object of ecommerce worker user.

`enterprise.utils.get_enterprise_customer(uuid)`

Get the EnterpriseCustomer instance associated with uuid.

Parameters**uuid** – The universally unique ID of the enterprise customer.**Returns**

The EnterpriseCustomer instance, or None if it doesn't exist.

`enterprise.utils.get_enterprise_customer_by_invite_key_or_404(invite_key_uuid)`

Given an EnterpriseCustomerInviteKey UUID, return the corresponding EnterpriseCustomer or raise a 404.

Parameters**invite_key_uuid** (*str*) – The UUID identifying an EnterpriseCustomerInviteKey.**Returns**

The EnterpriseCustomer given the EnterpriseCustomerInviteKey UUID.

Return type*(EnterpriseCustomer)*`enterprise.utils.get_enterprise_customer_by_slug_or_404(slug)`

Given an EnterpriseCustomer slug, return the corresponding EnterpriseCustomer or raise a 404.

Parameters**slug** (*str*) – The unique slug (a string) identifying the customer.**Returns**

The EnterpriseCustomer given the slug.

Return type*(EnterpriseCustomer)*`enterprise.utils.get_enterprise_customer_for_user(auth_user)`

Return first found enterprise customer instance for given user.

Some users are associated with an enterprise customer via *EnterpriseCustomerUser* model,

1. if given user is associated with any enterprise customer, return first enterprise customer.
2. otherwise return *None*.

Parameters**auth_user** (*contrib.auth.User*) – Django User**Returns**

enterprise customer associated with the current user.

Return type*(EnterpriseCustomer)*`enterprise.utils.get_enterprise_customer_or_404(enterprise_uuid)`

Given an EnterpriseCustomer UUID, return the corresponding EnterpriseCustomer or raise a 404.

Parameters**enterprise_uuid** (*str*) – The UUID (in string form) of the EnterpriseCustomer to fetch.

Returns

The EnterpriseCustomer given the UUID.

Return type

(*EnterpriseCustomer*)

`enterprise.utils.get_enterprise_customer_user(user_id, enterprise_uuid)`

Return the object for EnterpriseCustomerUser.

Parameters

- **user_id** (*str*) – user identifier
- **enterprise_uuid** (*UUID*) – Universally unique identifier for the enterprise customer.

Returns

enterprise customer user record

Return type

(*EnterpriseCustomerUser*)

`enterprise.utils.get_enterprise_utm_context(enterprise_customer)`

Get the UTM context for the enterprise.

`enterprise.utils.get_enterprise_uuids_for_user_and_course(auth_user, course_run_id,
is_customer_active=None)`

Get the EnterpriseCustomer UUID(s) associated with a user and a course id`.

Some users are associated with an enterprise customer via *EnterpriseCustomerUser* model,

1. if given user is enrolled in a specific course via an enterprise customer enrollment, return related enterprise customers as a list.
2. otherwise return empty list.

Parameters

- **auth_user** (*contrib.auth.User*) – Django User
- **course_run_id** (*str*) – Course Run to lookup an enrollment against.
- **active** – (boolean or None): Filter flag for returning active, inactive, or all uuids

Returns

enterprise customer uuids associated with the current user and course run or None

Return type

(list of *str*)

`enterprise.utils.get_enterprise_worker_user()`

Return the user object of enterprise worker user.

`enterprise.utils.get_identity_provider(provider_id)`

Get Identity Provider with given id.

Returns

Instance of ProviderConfig or None.

`enterprise.utils.get_idiff_list(list_a, list_b)`

Returns a list containing lower case difference of list_b and list_a after case insensitive comparison.

Parameters

- **list_a** – list of strings
- **list_b** – list of string

Returns

List of unique lower case strings computed by subtracting list_b from list_a.

`enterprise.utils.get_idp_choices()`

Get a list of identity providers choices for enterprise customer.

Returns

A list of choices of all identity providers, None if it can not get any available identity provider.

`enterprise.utils.get_integrated_channel_choices()`

Helper method to return channel code for each integrated channel.

`enterprise.utils.get_integrations_for_customers(customer_uuid)`

Helper method to return channel code for each enterprise customer with active integrations.

Parameters

customer_uuid (*UUID*) – uuid of an enterprise customer

Returns

a list of integration channel codes.

Return type

list

`enterprise.utils.get_language_code(language)`

Return IETF language tag of given language name.

- if language is not found in the language map then *en-US* is returned.

Parameters

language (*str*) – string language name

Returns: a language tag (two-letter language code - two letter country code if applicable)

`enterprise.utils.get_last_course_run_end_date(course_runs)`

Returns the end date of the course run that falls at the end.

`enterprise.utils.get_learner_portal_url(enterprise_customer)`

Learner portal url for enterprise_customer

`enterprise.utils.get_md5_hash(content)`

Get the MD5 hash digest of the given content.

Parameters

content (*str*) – Content in string format for calculating MD5 hash digest.

Returns

MD5 hash digest.

Return type

(*str*)

`enterprise.utils.get_notification_subject_line(course_name, template_configuration=None)`

Get a subject line for a notification email.

The method is designed to fail in a “smart” way; if we can’t render a database-backed subject line template, then we’ll fall back to a template saved in the Django settings; if we can’t render *_that_* one, then we’ll fall through to a friendly string written into the code.

One example of a failure case in which we want to fall back to a stock template would be if a site admin entered a subject line string that contained a template tag that wasn't available, causing a `KeyError` to be raised.

Parameters

- **course_name** (*str*) – Course name to be rendered into the string
- **template_configuration** – A database-backed object with a stored subject line template

`enterprise.utils.get_oauth2authentication_class()`

Return oauth2 authentication class to authenticate REST APIs with Bearer token.

`enterprise.utils.get_platform_logo_url()`

Return an absolute URL of the platform logo using the branding api.

`enterprise.utils.get_program_type_description(program_type)`

Get the pre-set description associated with this program type.

Parameters

program_type – The type of the program. Should be one of:

- “MicroMasters Certificate”
- “Professional Certificate”
- “XSeries Certificate”

Returns

The description associated with the program type. If none exists, then the empty string.

`enterprise.utils.get_request_value(request, key, default=None)`

Get the value in the request, either through query parameters or posted data, from a key.

Parameters

- **request** – The request from which the value should be gotten.
- **key** – The key to use to get the desired value.
- **default** – The backup value to use in case the input key cannot help us get the value.

Returns

The value we're looking for.

`enterprise.utils.get_social_auth_from_idp(idp, user=None, user_idp_id=None)`

Return social auth entry of user for given enterprise IDP.

`idp` (`EnterpriseCustomerIdentityProvider`): `EnterpriseCustomerIdentityProvider` Object
`user` (`User`): `User` Object
`user_idp_id` (`str`): User id of user in third party LMS

`enterprise.utils.get_sso_orchestrator_api_base_url()`

Return the SSO orchestrator api base url settings value.

`enterprise.utils.get_sso_orchestrator_basic_auth_password()`

Return the SSO orchestrator auth password settings value.

`enterprise.utils.get_sso_orchestrator_basic_auth_username()`

Return the SSO orchestrator auth username settings value.

`enterprise.utils.get_sso_orchestrator_configure_edx_oauth_path()`

Return the SSO orchestrator configure-edx-oauth endpoint path, or None if it is not defined.

`enterprise.utils.get_sso_orchestrator_configure_path()`

Return the SSO orchestrator configure path settings value.

`enterprise.utils.get_user_valid_idp(user, enterprise_customer)`

Return the default idp if it has user social auth record else it will return any idp with valid user social auth record

user (User): user object enterprise_customer (EnterpriseCustomer): EnterpriseCustomer object

`enterprise.utils.get_users_by_email(emails)`

Accept a list of emails, and separate them into users that exist on OpenEdX and users who don't.

Parameters

emails – An iterable of email addresses to split between existing and nonexisting

Returns

Queryset of users who exist in the OpenEdX platform and who were in the list of email addresses
unregistered_emails: List of unique emails which were in the original list, but do not yet exist as users

Return type

users

`enterprise.utils.has_course_run_available_for_enrollment(course_runs)`

Iterates over all course runs to check if there any course run that is available for enrollment.

Argument:

course_runs: list of course runs

Returns

True if found else false

`enterprise.utils.is_course_run_about_to_end(current_course_run)`

Return False if end - now > course run's weeks to complete otherwise True.

`enterprise.utils.is_course_run_active(course_run)`

Checks whether a course run is active. That is, whether the course run is published, enrollable, and marketable.

:param course_run: The metadata about a course run. :type course_run: dict

Returns

True if course run is "active"

Return type

bool

`enterprise.utils.is_course_run_available_for_enrollment(course_run)`

Check if a course run is available for enrollment.

`enterprise.utils.is_course_run_enrollable(course_run)`

Return true if the course run is enrollable, false otherwise.

We look for the following criteria:

1. **end date is greater than a reasonably-defined enrollment window, or undefined.**

A reasonably-defined enrollment window is 1 day before course run end date.

2. enrollment_start is less than now, or undefined.

3. `enrollment_end` is greater than now, or undefined.

`enterprise.utils.is_course_run_published(course_run)`

Return True if the course run's status value is "published".

`enterprise.utils.is_course_run_upgradeable(course_run)`

Return true if the course run has a verified seat with an unexpired upgrade deadline, false otherwise.

`enterprise.utils.is_pending_user(user)`

Returns true if `pending_user` attributes are detected

`enterprise.utils.is_user_enrolled(user, course_id, course_mode, enrollment_client=None)`

Query the enrollment API and determine if a learner is enrolled in a given course run track.

Parameters

- **user** – The user whose enrollment needs to be checked
- **course_mode** – The mode with which the enrollment should be checked
- **course_id** – course id of the course where enrollment should be checked.
- **enrollment_client** – An optional `EnrollmentAPIClient` if it's already been instantiated and should be passed in.

Returns

Whether or not enrollment exists

Return type

Boolean

`enterprise.utils.is_valid_url(url)`

Return where the specified URL is a valid absolute url.

`enterprise.utils.licensed_enterprise_course_enrollment_model()`

returns the `LicensedEnterpriseCourseEnrollment` class.

`enterprise.utils.localized_utcnow()`

Helper function to return localized `utcnow()`.

`enterprise.utils.logo_path(instance, filename)`

Returns the enterprise customer logo image path.

Parameters

- **instance** (*`EnterpriseCustomerBrandingConfiguration`*) – `EnterpriseCustomerBrandingConfiguration` object
- **filename** (*`str`*) – file to upload

Returns

path of image file e.g. `enterprise/branding/<enterprise_uuid>/logo_<uuid>.<ext>.lower()`

Return type

path

`enterprise.utils.parse_datetime_handle_invalid(datetime_value)`

Return the parsed version of a datetime string. If the string is invalid, return `None`.

`enterprise.utils.parse_lms_api_datetime(datetime_string,
datetime_format='%Y-%m-%dT%H:%M:%SZ')`

Parse a received datetime into a timezone-aware, Python datetime object.

Parameters

- **datetime_string** – A string to be parsed.
- **datetime_format** – A datetime format string to be used for parsing

```
enterprise.utils.send_email_notification_message(user, enrolled_in, dashboard_url,
                                                enterprise_customer_uuid, email_connection=None,
                                                admin_enrollment=False)
```

Send an email notifying a user about their enrollment in a course.

Parameters

- **user** – a dict with either of the following forms: - 1: { 'first_name': name, 'username': user_name, 'email': email } (similar to a User object) - 2: { 'user_email': user_email } (similar to a PendingEnterpriseCustomerUser object)
- **enrolled_in** (*dict*) – The dictionary contains details of the enrollable object (either course or program) that the user enrolled in. This **MUST** contain a *name* key, and **MAY** contain the other following keys:

```
- url: A human-friendly link to the enrollable's home page
- type: Either `course` or `program` at present
- branding: A special name for what the enrollable "is"; for
  ↳ example,
    "MicroMasters" would be the branding for a "MicroMasters
  ↳ Program"
- start: A datetime object indicating when the enrollable will be
  ↳ available.
```

- **dashboard_url** – link to enterprise customer's unique homepage for user
- **enterprise_customer_uuid** – The EnterpriseCustomer uuid that the enrollment was created using.
- **email_connection** – An existing Django email connection that can be used without creating a new connection for each individual message
- **admin_enrollment** – If true, uses admin enrollment template instead of default ones.

```
enterprise.utils.serialize_notification_content(enterprise_customer, course_details, course_id, users,
                                                admin_enrollment=False, activation_links=None)
```

Prepare serializable contents to send emails with (if using tasks to send emails)

Parameters

- **enterprise_customer** (*enterprise.models.EnterpriseCustomer*) –
- **course_details** (*dict*) – With at least 'title', 'start' and 'course' keys (usually obtained via CourseCatalogApiClient)
- **course_id** (*str*) –
- **users** (*list*) – list of users to enroll (each user should be a User or PendingEnterpriseCustomerUser)
- **activation_links** (*dict*) – a dictionary map of unactivated license user emails to license activation links

Returns: A list of dictionary objects that are of the form:

```
{
  "user": user
  "enrolled_in": {
    'name': course_name,
    'url': destination_url,
    'type': 'course',
    'start': course_start,
  },
  "dashboard_url": dashboard_url,
  "enterprise_customer_uuid": self.uuid,
  "admin_enrollment": admin_enrollment,
}
```

where **user** is one of

- 1: { 'first_name': name, 'username': user_name, 'email': email } (dict of User object)
- 2: { 'user_email' : user_email } (dict of PendingEnterpriseCustomerUser object)

`enterprise.utils.strip_html_tags(text, allowed_tags=None)`

Strip all tags from a string except those tags provided in *allowed_tags* parameter.

Parameters

- **text** (*str*) – string to strip html tags from
- **allowed_tags** (*list*) – allowed list of html tags

Returns: a string without html tags

`enterprise.utils.subsidized_enterprise_course_enrollment_model()`

returns the `LearnerCreditEnterpriseCourseEnrollment` class.

`enterprise.utils.track_enrollment(pathway, user_id, course_run_id, url_path=None)`

Emit a track event for enterprise course enrollment.

`enterprise.utils.track_enterprise_user_linked(user_id, enterprise_customer_key,
 enterprise_customer_id, created_new_ent_user)`

Emit a track event when user is linked to an enterprise

`enterprise.utils.track_event(user_id, event_name, properties)`

Emit a track event to segment (and forwarded to GA) for some parts of the Enterprise workflows.

`enterprise.utils.traverse_pagination(response, client, api_url)`

Traverse a paginated API response.

Extracts and concatenates “results” (list of dict) returned by DRF-powered APIs.

Parameters

- **response** (*Dict*) – Current response dict from service API;
- **client** (*requests.Session*) – either the `OAuthAPIClient` (from `edx_rest_api_client`) or `requests.Session` object;
- **api_url** (*str*) – API endpoint URL to call.

Returns

list of dict.

`enterprise.utils.truncate_string(string, max_length=16000000)`

Truncate a string to the specified max length. If max length is not specified, it will be set to MAX_ALLOWED_TEXT_LENGTH.

Returns

(truncated_string, was_truncated)

Return type

(tuple)

`enterprise.utils.ungettext_min_max(singular, plural, range_text, min_val, max_val)`

Return grammatically correct, translated text based off of a minimum and maximum value.

Example

min = 1, max = 1, singular = '{ } hour required for this course', plural = '{ } hours required for this course' output = '1 hour required for this course'

min = 2, max = 2, singular = '{ } hour required for this course', plural = '{ } hours required for this course' output = '2 hours required for this course'

min = 2, max = 4, range_text = '{ }-{ } hours required for this course' output = '2-4 hours required for this course'

min = None, max = 2, plural = '{ } hours required for this course' output = '2 hours required for this course'

Expects range_text to already have a translation function called on it.

Returns

None if both of the input values are None. `singular` formatted if both are equal or one of the inputs, but not both, are None, and the value is 1. `plural` formatted if both are equal or one of its inputs, but not both, are None, and the value is > 1. `range_text` formatted if min != max and both are valid values.

`enterprise.utils.unset_enterprise_learner_language(enterprise_customer_user)`

Unset the language preference of the given enterprise learners.

Parameters

enterprise_customer_user (`EnterpriseCustomerUser`) – Instance of the enterprise customer user.

`enterprise.utils.unset_language_of_all_enterprise_learners(enterprise_customer_uuid)`

Unset the language preference of all the learners belonging to the given enterprise customer.

Parameters

enterprise_customer_uuid (`UUID`) – uuid of an enterprise customer

`enterprise.utils.update_query_parameters(url, query_parameters)`

Return url with updated query parameters.

Parameters

- **url** (`str`) – Original url whose query parameters need to be updated.
- **query_parameters** (`dict`) – A dictionary containing query parameters to be added to course selection url.

Returns

slug identifier for the identity provider that can be used for identity verification of
users associated the enterprise customer of the given user.

Return type
(slug)

`enterprise.utils.validate_course_exists_for_enterprise(enterprise_customer, course_id, **kwargs)`

Validates that a specified course id exists within the LMS and within the enterprise_customer's catalog(s).

Parameters

- **enterprise_customer** (`EnterpriseCustomer`) – Instance of the enterprise customer.
- **course_id** (`string`) – The unique identifier of a course.
- **kwargs** – Should contain enrollment_client if it's already been instantiated and is passed in.

`enterprise.utils.validate_email_to_link(email, enterprise_customer, raw_email=None, message_template=None, raise_exception=True)`

Validate email to be linked to Enterprise Customer.

Performs two checks:

- Checks that email is valid
- Checks that it is not already linked to the provided Enterprise Customer

Parameters

- **email** (`str`) – user email to link
- **enterprise_customer** (`EnterpriseCustomer`) – the enterprise customer to link the email to
- **raw_email** (`str`) – raw value as it was passed by user - used in error message.
- **message_template** (`str`) – Validation error template string.
- **raise_exception** (`bool`) – whether to raise an exception when an email is invalidated

Raises

ValidationError – if email is invalid or already linked to Enterprise Customer.

Returns

Whether or not there is an existing record with the same email address.

Return type
`bool`

8.1.20 enterprise.validators module

Database models field validators.

`enterprise.validators.get_app_config()`
:return Application configuration.

`enterprise.validators.validate_content_filter_fields(content_filter)`
Validate particular fields (if present) passed in through content_filter are certain types.

`enterprise.validators.validate_hex_color(value)`
Validate value is suitable for a color hex value.

`enterprise.validators.validate_image_extension(value)`
Validate that a particular image extension.

`enterprise.validators.validate_image_size(image)`

Validate that a particular image size.

`enterprise.validators.validate_pgp_key(pgp_key)`

Validate that given PGP key is valid.

Raises

(**ValidationError**) – Raised if given `pgp_key` is not valid.

8.1.21 enterprise.views module

User-facing views for the Enterprise app.

class `enterprise.views.CourseEnrollmentView(**kwargs)`

Bases: `NonAtomicView`

Enterprise landing page view.

This view will display the course mode selection with related enterprise information.

PACING_FORMAT = {'instructor_paced': 'Instructor-Paced', 'self_paced': 'Self-Paced'}

get(*request, enterprise_uuid, course_id*)

Show course track selection page for the enterprise.

Based on *enterprise_uuid* in URL, the view will decide which enterprise customer's course enrollment page is to use.

Unauthenticated learners will be redirected to enterprise-linked SSO.

A 404 will be raised if any of the following conditions are met:

- No enterprise customer uuid kwarg *enterprise_uuid* in request.
- **No enterprise customer found against the enterprise customer uuid *enterprise_uuid* in the request kwargs.**
- No course is found in database against the provided *course_id*.

get_available_course_modes(*request, course_run_id, enterprise_catalog*)

Return the available course modes for the course run.

The provided `EnterpriseCustomerCatalog` is used to filter and order the course modes returned using the `EnterpriseCustomerCatalog`'s field "enabled_course_modes".

get_base_details(*request, enterprise_uuid, course_run_id*)

Retrieve fundamental details used by both POST and GET versions of this view.

Specifically, take an `EnterpriseCustomer` UUID and a course run ID, and transform those into an actual `EnterpriseCustomer`, a set of details about the course, and a list of the available course modes for that course run.

get_enterprise_course_enrollment_page(*request, enterprise_customer, course, course_run, course_modes, enterprise_course_enrollment, data_sharing_consent*)

Render enterprise-specific course track selection page.

post(*request, enterprise_uuid, course_id*)

Process a submitted track selection form for the enterprise.

set_final_prices(*modes, request*)

Set the final discounted price on each premium mode.

class enterprise.views.**EnterpriseLoginView**(**kwargs)

Bases: [FormView](#)

Allow an enterprise learner to login by enterprise customer's slug login.

form_class

alias of [EnterpriseLoginForm](#)

form_invalid(*form*)

If the form is invalid then return the errors.

form_valid(*form*)

If the form is valid, redirect to the third party auth login page.

get_context_data(**kwargs)

Return the context data needed to render the view.

template_name = 'enterprise/enterprise_customer_login_page.html'

class enterprise.views.**EnterpriseProxyLoginView**(**kwargs)

Bases: [View](#)

Allows an enterprise learner to login via existing flow from the learner portal.

get(*request*)

Redirects a learner through login with their enterprise's third party auth if it uses tpa.

class enterprise.views.**EnterpriseSelectionView**(**kwargs)

Bases: [FormView](#)

Allow an enterprise learner to activate one of learner's linked enterprises.

dispatch(*request, *args, **kwargs*)

form_class

alias of [EnterpriseSelectionForm](#)

form_invalid(*form*)

If the form is invalid then return the errors.

form_valid(*form*)

If the form is valid, activate the selected enterprise.

get_context_data(**kwargs)

Return the context data needed to render the view.

get_initial()

Return the initial data to use for forms on this view.

template_name = 'enterprise/enterprise_customer_select_form.html'

class enterprise.views.**FailedEnrollmentReason**(*enrollment_client_error, failure_reason_message*)

Bases: [tuple](#)

enrollment_client_error

Alias for field number 0

failure_reason_message

Alias for field number 1

class `enterprise.views.GrantDataSharingPermissions(**kwargs)`

Bases: `View`

Provide a form and form handler for data sharing consent.

View handles the case in which we get to the “verify consent” step, but consent hasn’t yet been provided - this view contains a GET view that provides a form for consent to be provided, and a POST view that consumes said form.

course_or_program_exist(*course_id*, *program_uuid*)

Return whether the input course or program exist.

static create_enterprise_course_enrollment(*request*, *enterprise_customer*, *course_id*,
license_uuid=None)

Create EnterpriseCustomerUser and EnterpriseCourseEnrollment record if not already exists.

get(*request*)

Render a form to collect user input about data sharing consent.

get_context_from_db(*consent_page*, *platform_name*, *item*, *context*)

Make set of variables(populated from db) that will be used in data sharing consent page.

get_course_or_program_context(*enterprise_customer*, *course_id=None*, *program_uuid=None*)

Return a dict having course or program specific keys for data sharing consent page.

get_default_context(*enterprise_customer*, *platform_name*)

Get the set of variables that will populate the template by default.

get_page_language_context_data(*course_id*, *enterprise_customer*, *success_url*, *failure_url*,
license_uuid, *request*, *platform_name*)

Return a dict of data for the language on the page.

is_course_run_id(*course_id*)

Returns True if the *course_id* is in the correct format of a *course_run_id*, false otherwise.

Parameters

course_id (*str*) – The *course_key* or *course run id*

Returns

True or False

Return type

(Boolean)

post(*request*)

Process the above form.

preview_mode = False

class `enterprise.views.HandleConsentEnrollment(**kwargs)`

Bases: `View`

Handle enterprise course enrollment at providing data sharing consent.

View handles the case for enterprise course enrollment after successful consent.

get(*request, enterprise_uuid, course_id*)

Handle the enrollment of enterprise learner in the provided course.

Based on *enterprise_uuid* in URL, the view will decide which enterprise customer's course enrollment record should be created.

Depending on the value of query parameter *course_mode* then learner will be either redirected to LMS dashboard for audit modes or redirected to ecommerce basket flow for payment of premium modes.

class `enterprise.views.NonAtomicView`(**kwargs)

Bases: `View`

A base class view for views that disable atomicity in requests.

dispatch(*request, *args, **kwargs*)

Disable atomicity for the view.

Since we have settings.ATOMIC_REQUESTS enabled, Django wraps all view functions in an atomic transaction, so they can be rolled back if anything fails.

However, we need to be able to save data in the middle of get/post(), so that it's available for calls to external APIs. To allow this, we need to disable atomicity at the top dispatch level.

class `enterprise.views.ProgramEnrollmentView`(**kwargs)

Bases: `NonAtomicView`

Enterprise Program Enrollment landing page view.

This view will display information pertaining to program enrollment, including the Enterprise offering the program, its (reduced) price, the courses within it, and whether one is already enrolled in them, and other several pieces of Enterprise context.

static extend_course(*course, enterprise_customer, request*)

Extend a course with more details needed for the program landing page.

In particular, we add the following:

- *course_image_uri*
- *course_title*
- *course_level_type*
- *course_short_description*
- *course_full_description*
- *course_effort*
- *expected_learning_items*
- *staff*

get(*request, enterprise_uuid, program_uuid*)

Show Program Landing page for the Enterprise's Program.

Render the Enterprise's Program Enrollment page for a specific program. The Enterprise and Program are both selected by their respective UUIDs.

Unauthenticated learners will be redirected to enterprise-linked SSO.

A 404 will be raised if any of the following conditions are met:

- No enterprise customer UUID query parameter *enterprise_uuid* found in request.

- No enterprise customer found against the enterprise customer uuid `enterprise_uuid` in the request kwargs.
- No Program can be found given `program_uuid` either at all or associated with the Enterprise..

get_enterprise_program_enrollment_page(*request, enterprise_customer, program_details*)

Render Enterprise-specific program enrollment page.

get_program_details(*request, program_uuid, enterprise_customer*)

Retrieve fundamental details used by both POST and GET versions of this view.

Specifically:

- Take the program UUID and get specific details about the program.
- Determine whether the learner is enrolled in the program.
- Determine whether the learner is certificate eligible for the program.

post(*request, enterprise_uuid, program_uuid*)

Process a submitted track selection form for the enterprise.

class `enterprise.views.RouterView`(***kwargs*)

Bases: `NonAtomicView`

A router or gateway view for managing Enterprise workflows.

`COURSE_ENROLLMENT_VIEW_URL = '/enterprise/{}/course/{}/enroll/'`

`HANDLE_CONSENT_ENROLLMENT_VIEW_URL =
'/enterprise/handle_consent_enrollment/{}/course/{}/'`

`PROGRAM_ENROLLMENT_VIEW_URL = '/enterprise/{}/program/{}/enroll/'`

`VIEWS = {'/enterprise/handle_consent_enrollment/{}/course/{}/': <class
'enterprise.views.HandleConsentEnrollment'>, '/enterprise/{}/course/{}/enroll/':
<class 'enterprise.views.CourseEnrollmentView'>,
'/enterprise/{}/program/{}/enroll/': <class
'enterprise.views.ProgramEnrollmentView'>}`

eligible_for_direct_audit_enrollment(*request, enterprise_customer, resource_id, course_key=None*)

Return whether a request is eligible for direct audit enrollment for a particular enterprise customer.

‘resource_id’ can be either `course_run_id` or `program_uuid`. We check for the following criteria: - The *audit* query parameter. - The user’s being routed to the course enrollment landing page. - The customer’s catalog contains the course in question. - The audit track is an available mode for the course.

get(*request, *args, **kwargs*)

Run some custom GET logic for Enterprise workflows before routing the user through existing views.

In particular, before routing to existing views:

- If the requested resource is a course, find the current course run for that course, and make that course run the requested resource instead.
- Look to see whether a request is eligible for direct audit enrollment, and if so, directly enroll the user.

static get_course_run_id(*user, enterprise_customer, course_key*)

User is requesting a course, we need to translate that into the current course run.

Parameters

- **user** –
- **enterprise_customer** –
- **course_key** –

Returns

course_run_id

static `get_path_variables(**kwargs)`

Get the base variables for any view to route to.

Currently gets: - *enterprise_uuid* - the UUID of the enterprise customer. - *course_run_id* - the ID of the course, if applicable. - *program_uuid* - the UUID of the program, if applicable.

post(*request*, *args, **kwargs)

Run some custom POST logic for Enterprise workflows before routing the user through existing views.

redirect(*request*, *args, **kwargs)

Redirects to the appropriate view depending on where the user came from.

exception `enterprise.views.VerifiedModeUnavailableException`

Bases: `Exception`

Exception that indicates the verified enrollment mode has expired or is otherwise unavailable for a course run.

`enterprise.views.add_reason_to_failure_url(base_failure_url, failure_reason)`

Adds a query param to the given *base_failure_url* indicating why an enrollment has failed.

`enterprise.views.get_global_context(request, enterprise_customer=None)`

Get the set of variables that are needed by default across views.

`enterprise.views.get_price_text(price, request)`

Return the localized converted price as string (ex. '\$150 USD').

If the *local_currency* switch is enabled and the users location has been determined this will convert the given price based on conversion rate from the Catalog service and return a localized string

`enterprise.views.get_safe_redirect_url(url, requires_https=None)`

Ensure that the URL which is going to be used for redirection exists in whitelist.

`enterprise.views.render_page_with_error_code_message(request, context_data, error_code, exception=None, **kwargs)`

Return a 404 page with specified *error_code* after logging error and adding message to django messages.

`enterprise.views.should_upgrade_to_licensed_enrollment(consent_record, license_uuid)`

In the event that a learner did enroll into audit from B2C and they consented on the data sharing consent page, we want to upgrade their audit enrollment into verified when they view the course in the learner portal and hit the DSC GET endpoint again.

`enterprise.views.verify_edx_resources()`

Ensure that all necessary resources to render the view are present.

8.1.22 Module contents

Your project description goes here.

CHANGE LOG

9.1 Unreleased

- nothing unreleased

9.2 [4.17.0]

- feat: limit the number of resulting learners in Django admin manage learners view

9.3 [4.16.5]

- feat: splitting out group membership serializer learner id into lms user ID and ecu ID

9.4 [4.16.4]

- revert: fix: set default language for all learners linked with an enterprise customer

9.5 [4.16.3]

- fix: fixing the removal logic for EnterpriseGroupMemberships, adding optional flag

9.6 [4.16.2]

- fix: fix the PATCH method of EnterpriseCourseEnrollmentView.

9.7 [4.16.1]

- fix: add enterprise audit reporting status in plotly auth token

9.8 [4.16.0]

- feat: Adding python3.11 support. Dropped django32 support.

9.9 [4.15.12]

- feat: update enterprise customer serializer to include active integration codes

9.10 [4.15.11]

- fix: support *force_enrollment* in serializers used by bulk enrollment (ENT-8788)

9.11 [4.15.10]

- fix: set default language for all learners linked with an enterprise customer

9.12 [4.15.9]

- fix: return a 404 response for inactive CSOD customers while fetching courses

9.13 [4.15.8]

- fix: SSO self-serve tool invalid entityId parsing

9.14 [4.15.7]

- feat: add send group membership invite and removal braze emails

9.15 [4.15.6]

- perf: update user preferences inside an async task to void request timeout

9.16 [4.15.5]

- fix: Improved the query to only fetch active configs for CSOD customers.

9.17 [4.15.4]

- fix: allowing for existing pecus to be added to enterprise groups

9.18 [4.15.3]

- feat: replacing non encrypted fields of degreed config model with encrypted ones

9.19 [4.15.2]

- feat: save cornerstone learner's information received from frontend.

9.20 [4.15.1]

- feat: allowing for sorting and filtering of the enterprise group learner endpoints

9.21 [4.15.0]

- feat: Add new languages in enterprise customer admin

9.22 [4.14.0]

- Added the ability for enterprise customers to enable One Academy for its learners.

9.23 [4.13.14]

- feat: handle Degreed 409 case (mark as active instead of recreating course)

9.24 [4.13.13]

- fix: adding additional info to the enterprise group serializer

9.25 [4.13.12]

- feat: adding additional info to the enterprise group membership serializer

9.26 [4.13.11]

- feat: pass force_enrollment when bulk enrolling learners

9.27 [4.13.10]

- fix: remove filter to debug failing transmissions

9.28 [4.13.9]

- fix: add missing filter to disable failing transmissions for 24hrs

9.29 [4.13.8]

- feat: adding an activated_at value to group membership records

9.30 [4.13.7]

- fix: adding get_queryset for fix of integrated channel api logs loading

9.31 [4.13.6]

- feat: disable failing transmissions for 24hrs

9.32 [4.13.5]

- feat: added autocomplete for enterprise customer in EnterpriseCustomerCatalogAdmin

9.33 [4.13.4]

- feat: admin pages for enterprise groups and enterprise group memberships

9.34 [4.13.3]

- feat: adding management command to remove expired pending group memberships

9.35 [4.13.2]

- feat: add a waffle flag for enterprise groups feature

9.36 [4.13.1]

- feat: adding soft delete functionality for groups and group memberships

9.37 [4.13.0]

- feat: add Waffle-based *enterprise_features* to the *EnterpriseCustomerUserViewSet*.

9.38 [4.12.6]

- fix: Proximus learner transmission failures

9.39 [4.12.5]

- feat: adding a group membership to the EnterpriseCustomerUserReadOnlySerializer

9.40 [4.12.4]

- feat: adding assign_learners and remove_learners api endpoints for enterprise groups

9.41 [4.12.3]

- feat: management command to clear out excessive records for API log table

9.42 [4.12.2]

- feat: add api /learners/ endpoint to the enterprise group viewset

9.43 [4.12.1]

- feat: unlink canvas user if decommissioned on canvas side

9.44 [4.12.0]

- feat: Remove history tables for integrated channels customers configurations.

9.45 [4.11.15]

- feat: CRUD api endpoints for the enterprise group table

9.46 [4.11.14]

- feat: added channel_name for api call logs records

9.47 [4.11.13]

- feat: Update IntegratedChannelAPIRequestLogs list view

9.48 [4.11.12]

- feat: new enterprise models relating to enterprise groups

9.49 [4.11.11]

- feat: record degreed API calls

9.50 [4.11.10]

- feat: add unique constraint on learner data transmission audit models

9.51 [4.11.9]

- feat: register admin view for IntegratedChannelAPIRequestLogs

9.52 [4.11.8]

- chore: Update requirements

9.53 [4.11.7]

- feat: update canvas client to store API calls in DB

9.54 [4.11.6]

- feat: Added a flag for prequery search suggestions

9.55 [4.11.5]

- fix: Added logs and remove cornerstone from management command

9.56 [4.11.4]

- feat: update blackboard client to store API calls in DB

9.57 [4.11.3]

- feat: update cornerstone client to store API calls in DB

9.58 [4.11.2]

- feat: added caching for fetching degreed course id

9.59 [4.11.1]

- Added management command to fix *LearnerDataTransmissionAudit* table records.

9.60 [4.11.0]

- Added the ability for enterprise customers to enable/disable academies.

9.61 [4.10.11]

- fix: add missing comma to catalog query fields list.

9.62 [4.10.10]

feat: remove ability to edit catalog *include_exec_ed_2u_courses*

9.63 [4.10.9]

feat: added logs temporarily for ENT-8276

9.64 [4.10.8]

fix: guard against null content_filters

9.65 [4.10.7]

feat: mgmt command to add exec-ed exclusions to catalogs

9.66 [4.10.6]

fix: tweak catalog compare mgmt command

9.67 [4.10.5]

fix: tweak catalog compare mgmt command

9.68 [4.10.4]

revert: 4.10.1

9.69 [4.10.3]

feat: management command to test query migration

9.70 [4.10.2]

feat: removed unencrypted user credentials data columns

9.71 [4.10.1]

feat: added json field in learner transmission audit to record 3 most latest response statuses

9.72 [4.10.0]

feat: enrollment API enhancements

- Allows Enrollment API Admin to see all enrollments.
- Makes the endpoint return more fields, such as: enrollment_track, enrollment_date, user_email, course_start and course_end.
- Changes EnterpriseCourseEnrollment's default ordering from 'created' to 'id', which equivalent, but faster in some cases (due to the existing index on 'id').

9.73 [4.9.5]

feat: replacing non encrypted fields of moodle config model with encrypted ones

9.74 [4.9.4]

feat: Add model for integrated channel API request log table (ENT-8018)

9.75 [4.9.3]

fix: Remove SAP debug logs

9.76 [4.9.2]

refactor: learner data transmission audit record to utilize the existing records (ENT-8005)

9.77 [4.9.1]

fix: enable printing of error messages from the SSO Orchestrator API (ENT-8169)

9.78 [4.9.0]

feat: add “Setup Auth org id” action for Enterprise Customers (ENT-8169)

9.79 [4.8.18]

feat: added logs to debug ENT-8130

9.80 [4.8.17]

fix: remove logs from Degreed2 client

9.81 [4.8.16]

fix: removed course completion check from is_already_transmitted utility (ENT 7837)

9.82 [4.8.15]

fix: restructured response from moodle request wrapper feat: replacing non encrypted fields of moodle config model with encrypted ones (ENT 5613)

9.83 [4.8.14]

feat: Modified existing content transmission job to post skills metadata to Degreed2

9.84 [4.8.13]

fix: fixed create_course_completion request’s response handling in case return body is 0

9.85 [4.8.12]

- chore: update paragon npm dependency to move to the new @openedx scope.

9.86 [4.8.11]

feat: allow incomplete course learner transmissions till completion refactor: adding log inside moodle request wrapper

9.87 [4.8.10]

feat: not submitting sso orchestrator records if no changes occur

9.88 [4.8.9]

feat: adding timeouts to sso orchestrator configurations and api cleanup

9.89 [4.8.8]

fix: added more logs and handled edge cases in Degreed assign skills job

9.90 [4.8.7]

refactor: Override the default save method of `EnterpriseCustomerPluginConfiguration` to update only changed fields

9.91 [4.8.6]

feat: add `marked_authorized` flag to SSO config

9.92 [4.8.5]

feat: Added a management command to assign skills to Degreed courses

9.93 [4.8.4]

fix: changed relative `resumeCourseRunUrl` to an absolute URL

9.94 [4.8.3]

refactor: adding log for learner data transmission

9.95 [4.8.1]

feat: integrated resumeCourseRunUrl into enrollments API

9.96 [4.7.6]

chore: remove unnecessary logs from the integrated channels

9.97 [4.7.5]

feat: added flag to allow in progress course learner data transmission

9.98 [4.7.4]

feat: added fields for holding encrypted data in database

9.99 [4.7.3]

feat: added management command to re-encrypt enterprise customer reporting configs

9.100 [4.7.2]

feat: stop learner data transmissions for course runs

9.101 [4.7.1]

chore: retire Degreed v1 code from the set of channels

9.102 [4.7.0]

feat: Add an `api-docs` page that lives at `[LMS_ROOT_URL]/enterprise/api-docs/`

9.103 [4.6.12]

feat: unlink `degreed2` inactive user

9.104 [4.6.11]

chore: Aligning SAP naming conventions

9.105 [4.6.10]

chore: Update requirements

9.106 [4.6.9]

chore: returning SP metadata url from the sso orchestrator to the API caller

9.107 [4.6.8]

feat: truncate API Response before writing to the `APIResponseRecord` fix: initiate filtering the API Response only when a successful response is received

9.108 [4.6.7]

feat: filter courses from API Response of SAPSF to store in the `APIResponseRecord` table

9.109 [4.6.6]

chore: orchestrator exception handling and submission refinements

9.110 [4.6.5]

feat: Added logs for Degreed2 client

9.111 [4.6.4]

chore: updating sso orchestrator self service api endpoints

9.112 [4.6.3]

fix: Remove not required fields

9.113 [4.6.2]

fix: clarify contact email helper text for enterprise customer

9.114 [4.6.1]

feat: Added the `disable_subject_metadata_transmission` flag to `CornerstoneEnterpriseCustomerConfiguration`.

9.115 [4.6.0]

feat: Added `enable_source_demo_data_for_analytics_and_lpr` field to `EnterpriseCustomer`.

9.116 [4.5.7]

fix: Fixed ChatGPT prompt and a few model modifications for better readability for admins.

9.117 [4.5.6]

feat: Added logs for learner completion data post request[moodle]

9.118 [4.5.5]

chore: sso orchestrator configs should start inactive and be activated upon successful configuration

9.119 [4.5.4]

feat: inactive moodle course instead of true delete

9.120 [4.5.3]

feat: added dry run mode for content metadata transmission

9.121 [4.5.2]

chore: adding a more flexible way of fetching api request data

9.122 [4.5.1]

fix: fix how we determine the value of active flag within schedule for SAP

9.123 [4.5.0]

feat: Add the `enable_programs` field for EnterpriseCustomer

9.124 [4.4.0]

feat: Add the `enable_pathways` field for EnterpriseCustomer

9.125 [4.3.4]

feat: add migrations for Django 4.2

9.126 [4.3.3]

chore: adding display name to the sso orchestrator table

9.127 [4.3.2]

fix: enterprise sso orchestrator api cleanup

9.128 [4.3.1]

chore: use `lms_update_or_create_enrollment` without feature flag

9.129 [4.3.0]

feat: Added the `enable_career_engagement_network_on_learner_portal` field for `EnterpriseCustomer`

9.130 [4.2.0]

feat: create generic `PaginationWithFeatureFlags` to add a `features` property to DRF's default pagination response containing Waffle-based feature flags. feat: integrate `PaginationWithFeatureFlags` with `EnterpriseCustomerViewSet`.

9.131 [4.1.15]

feat: enterprise sso orchestrator api client implementation

9.132 [4.1.14]

feat: adding read only field to `EnterpriseCustomerCatalogSerializer` model

9.133 [4.1.13]

feat: adding braze email task to sso orchestration endpoint

9.134 [4.1.12]

feat: enterprise sso orchestration record table api implementation

9.135 [4.1.11]

fix: adding api credential boolean into django

9.136 [4.1.10]

chore: adding more logging to the `_sanitize_and_set_item_metadata` flow

9.137 [4.1.9]

Fixing typos

9.138 [4.1.8]

Added support for Django 4.2

9.139 [4.1.7]

fix: enterprise api credentials endpoint cleanup.

9.140 [4.1.6]

fix: putting `api_credentials` bool in api response to access in admin portal

9.141 [4.1.5]

feat: implementing sso enterprise sso orchestrator config table

9.142 [4.1.4]

feat: enterprise API Credentials generation endpoints

9.143 [4.1.3]

fix: bringing changelog and version number back in sync (re-release of 4.1.2).

9.144 [4.1.2]

feat: implementing sso enterprise sso orchestrator config table

9.145 [4.1.1]

feat: Added patch endpoint to update an enterprise customer catalog.

9.146 [4.1.0]

feat: Added the ability to get AI generated learner engagement and learner progress summary.

9.147 [4.0.18]

fix: enrolled course card grouping behavior for exec-ed courses

9.148 [4.0.17]

chore: restoring licensed enrollment table if it does not exist

9.149 [4.0.16]

feat: Replace deprecated *django-fernet-fields* with its forked *django-fernet-fields-v2*.

9.150 [4.0.15]

fix: make api-record in lms admin read only

9.151 [4.0.14]

bug: swapping grades api grade_percent return value type from string to float

9.152 [4.0.13]

fix: more flexible default site

9.153 [4.0.12]

fix: allow sub directories in moodle base URLs

9.154 [4.0.11]

feat: upgrade django-simple-history to 3.1.1

9.155 [4.0.10]

chore: fixing import path error in management command

9.156 [4.0.9]

feat: customizable management command to generate local enterprise data

9.157 [4.0.8]

fix: reorganizing enterprise customer django admin portal

9.158 [4.0.7]

feat: add grade value to learner transmission for degreed v2

9.159 [4.0.6]

fix: variable name conflicts in monthly_impact_report command

9.160 [4.0.5]

feat: incorporate additional attributes to degreed v2

9.161 [4.0.4]

feat: remove content transmission audits without a catalog uuid

9.162 [4.0.3]

fix: changing sap transmit metadata flow to account for rate limiting

9.163 [4.0.2]

fix: removing items to delete dependency on the catalog service

9.164 [4.0.1]

chore: upgrade course_enrollment from audit to verified

9.165 [4.0.0]

chore: upgrade node from version 16 to version 18

9.166 [3.69.1]

fix: content metadata exporter sanitizing content to delete

9.167 [3.69.0]

refactor: Replaced the deprecated *NullBooleanField* with *BooleanField(null=True)*

9.168 [3.68.1]

fix: pick first object from CourseDetails

9.169 [3.68.0]

feat: add more metadata into *EnterpriseCourseEnrollmentView*

9.170 [3.67.7]

feat: marking orphaned content audits when catalogs are deleted

9.171 [3.67.6]

chore: fixing doc string linter errors

9.172 [3.67.5]

chore: better formatting of the enterprise api views

9.173 [3.67.4]

feat: add button to update customer modified time

9.174 [3.67.3]

feat: adding management command to clear error state

9.175 [3.67.2]

fix: fixing name of table used by model fetching method

9.176 [3.67.1]

chore: more orphaned content transmission logging

9.177 [4.0.0]

feat!: upgraded to Node v18, added .nvmrc and updated workflows.

9.178 [3.67.0]

fix: compare lowered input email address to lowered auth user record email address during bulk enrollment.

9.179 [3.66.8]

fix: optional x-request-id logging

9.180 [3.66.7]

feat: optional x-request-id logging

9.181 [3.66.6]

- feat: increase the throttle limit of service users for EnterpriseCustomerViewSet
- Switch from `edx-sphinx-theme` to `sphinx-book-theme` since the former is deprecated

9.182 [3.66.5]

chore: set default expiration_date for EnterpriseCustomerInviteKey

9.183 [3.66.4]

feat: add more logging in the bulk enrollment flow

9.184 [3.66.3]

fix: prevent org_id param addition to exec-ed course home url when auth_org_id is not present

9.185 [3.66.2]

chore: unique constraint on transmission audits to prevent duplicates

9.186 [3.66.1]

fix: Fixed url encoding issue with org_id in executive education course url

9.187 [3.66.0]

feat: Added org_id for executive education courses landing page

9.188 [3.65.4]

fix: swapping enterprise unenrolled bool to timestamp and minor code cleanup

9.189 [3.65.2]

feat: extending the enterprise fulfillment api serializer to contain more info

9.190 [3.65.1]

fix: making sure unenrollment is saved while revoking fulfillment

9.191 [3.65.0]

feat: new enterprise endpoint to surface filterable unenrolled subsidized enrollments

9.192 [3.64.1]

fix: Reverted course_run_url for Executive Education courses

9.193 [3.64.0]

feat: Updated course_run_url for Executive Education courses

9.194 [3.63.0]

feat: Hooking enterprise enrollments up to platform signals to write unenrollment records.
New field *unenrolled* on enterprise enrollments to track enrollment status, defaults to *None*.

9.195 [3.62.7]

feat: include obsolete attribute in degreed content transmissions

9.196 [3.62.6]

chore: adding better logging to the remove dup audit management command

9.197 [3.62.5]

feat: include auth_org_id in enterprise customer api

9.198 [3.62.4]

fix: duplicate records reading queryset length then splicing

9.199 [3.62.3]

feat: Add idempotent catalog creation endpoint

9.200 [3.62.2]

fix: management command fix- reading queryset length in a mysql supported way

9.201 [3.62.1]

feat: new management command to remove duplicate transmission audits

9.202 [3.62.0]

feat: Add 'auth_org_id' field to EnterpriseCustomer for Auth0 integration

9.203 [3.61.16]

feat: Q filtering to enterprise customer basic list

9.204 [3.61.15]

feat: Add customer catalog write only viewset

9.205 [3.61.14]

fix: Integrated channels metadata exporter now considers failed update audits as existing records

9.206 [3.61.13]

feat: expands pagination page size from enterprise catalog query view

9.207 [3.61.12]

chore: remove codecov from requirements as it removed from PyPi and is breaking CI when trying to install the package. Pinned pip-tools and tox in `common_constraints.txt` due to dependency resolution issues with click and filelock.

9.208 [3.61.11]

feat: include owners and longer descriptions for degreed2 content metadata transmissions

9.209 [3.61.10]

feat: new tagging orphaned content test for integrated channels

9.210 [3.61.9]

feat: Serialize and create a viewset for `enterpriseCatalogQuery` as readonly

9.211 [3.61.8]

feat: added enterprise fulfillment viewset

9.212 [3.61.7]

feat: surfacing subsidized enrollment uuid value in the bulk enrollment endpoint

9.213 [3.61.6]

feat: Add `user_id` support to `enroll_learners_in_courses` endpoint

9.214 [3.61.5]

fix: Ensure *EnterpriseCustomerBrandingConfigurationSerializer* returns correct logo URL on stage/production after *settings.DEFAULT_FILE_STORAGE* changed to use *storages.backends.s3boto3.S3Boto3Storage* instead of *storages.backends.s3boto.S3BotoStorage*.

9.215 [3.61.4]

fix: improved admin screen for system wide enterprise role assignments

9.216 [3.61.3]

chore: cleanup squashed migration files

9.217 [3.61.2]

fix: migrating uuid and updating column constraints

9.218 [3.61.1]

fix: allowing for uuid column migration to be non-unique

9.219 [3.61.0]

feat: enterprise entitlements and subsidy based fulfillment models hierarchy rework

9.220 [3.60.21]

feat: django admin bulk action to clear *remote_deleted_at* on *ContentMetadataItemTransmission* records

9.221 [3.60.20]

fix: changing api client to parse grades api response body

9.222 [3.60.19]

fix: applied validations on report delivery_method

9.223 [3.60.18]

fix: checking for response attr in http exception handling

9.224 [3.60.17]

fix: making backend generated fields read only on django admin

9.225 [3.60.16]

fix: adding backoff and retry logic to the grades api client

9.226 [3.60.15]

feat: allowing manage learners form enroll learners using exec ed modes

9.227 [3.60.14]

feat: adding exec ed course mode to enterprise

9.228 [3.60.13]

fix: handle file compression check and test

9.229 [3.60.12]

feat: adding last modified timestamp to single LMS config endpoints

9.230 [3.60.11]

fix: integrations log TypeErrors

9.231 [3.60.10]

fix: making moodle client return value match other channels

9.232 [3.60.9]

fix: memoize synced_at datetimes on config record

9.233 [3.60.8]

fix: failed content deletes and creates are never retried

9.234 [3.60.7]

fix: cleaning up serializer field inheritance

9.235 [3.60.6]

fix: password is required when pgp key is provided

9.236 [3.60.5]

fix: adding an index for performance, non-blocking syntax

9.237 [3.60.4]

fix: add back squashed migrations (fixes devstack provisioning)

9.238 [3.60.3]

feat: exec ed content is tagged and distinguishable from edx content

9.239 [3.60.2]

fix: sorting error reporting audits by all datetimes at once instead of three separate order bys

9.240 [3.60.1]

fix: adding an index for performance, testing syntax for no-lock

9.241 [3.60.0]

build: write an updated pylintrc and quiet a noisy logging statement

9.242 [3.59.2]

chore: allowing for page size specification in error reporting api

9.243 [3.59.1]

fix: cleaning up integrated channels inheritance column types

9.244 [3.59.0]

refactor: Removed unused code related to tableau.

9.245 [3.58.18]

fix: internal service error fix for enterprise branding

9.246 [3.58.17]

fix: fixed a bug that didn't allow admins to create an account or update password

9.247 [3.58.16]

fix: fix bug that didn't allow admins to customize branding

9.248 [3.58.15]

fix: catch update transmissions that failed and need to be resent

9.249 [3.58.14]

feat: Add health check for canvas integrated channels

9.250 [3.58.13]

feat: Add in learner and content sync time records to integrated channel configs

9.251 [3.58.12]

fix: make integrated channels api response record field readonly in admin view

9.252 [3.58.11]

fix: Added model field validation for `pgp_encryption_key`

9.253 [3.58.10]

refactor: update enterprise manual reporting to make it generic

9.254 [3.58.9]

fix: Added validation for PGP encryption key entered by the user.

9.255 [3.58.8]

feat: integration sync reporting viewset improvements

9.256 [3.58.7]

feat: populate friendly_status_message in content and learner metadata api

9.257 [3.58.6]

chore: smarter canvas client update course behavior (course deleted)

9.258 [3.58.5]

feat: integrated channels customer configs soft delete support

9.259 [3.58.4]

chore: fixing api view that included unneeded values

9.260 [3.58.3]

feat: transmission audit admin view and api improvements

9.261 [3.58.2]

fix: integrated channels not picking up courses to update

9.262 [3.58.1]

feat: adding new variables to the learner transmission audit model

9.263 [3.58.0]

feat: Add a new endpoint to generate a signed token for plotly analytics.

9.264 [3.57.3]

feat: remove debug logging used to investigate missing CSOD deletes

9.265 [3.57.2]

feat: Added POST support for catalog query preview

9.266 [3.57.1]

fix: improving transmission records by moving response body to new table

9.267 [3.57.0]

- feat: adds an `EnterpriseCatalogQuery.include_exec_ed_2u_courses` field. When True for an `EnterpriseCatalogQuery` record, this field indicates that course content with type `executive-education-2u` should not be excluded from being associated with that record. See related enterprise-catalog PR, which includes an ADR: <https://github.com/openedx/enterprise-catalog/pull/507>

9.268 [3.56.16]

feat: integration sync status self service endpoints

9.269 [3.56.15]

feat: tagging exported content records for transmission

9.270 [3.56.14]

chore: add plugin_configuration_id to standard integration logging

9.271 [3.56.13]

chore: smarter canvas client update course behavior

9.272 [3.56.12]

chore: adding http status response code to content record django admin table

9.273 [3.56.11]

chore: cleaning up Canvas client url construction

9.274 [3.56.10]

fix: adhering to urljoin patterns in integrated channels API views

9.275 [3.56.9]

fix: properly truncate payload to resolve missing CSOD deletes

9.276 [3.56.8]

feat: add debug logging to investigate missing CSOD deletes

9.277 [3.56.7]

feat: add debug logging to investigate missing CSOD deletes

9.278 [3.56.6]

feat: expand utility of CSOD deleted_at reset job

9.279 [3.56.5]

fix: properly pass SAP client status back to content transmission records

9.280 [3.56.4]

fix: open redirect url whitelisting for data sharing consent and change enterprise page

9.281 [3.56.3]

fix: replace id with uuid in branding logo file path

9.282 [3.56.2]

fix: refactor the way we send cornerstone content metadata deletes

9.283 [3.56.1]

fix: accounting for integrated Canvas instances that have no root account Ids.

9.284 [3.56.0]

feat: refactor content metadata jobs to save api call status

9.285 [3.55.3]

fix: accurately selecting content key values when filtering for existing content metadata transmission audits.

9.286 [3.55.2]

fix: integrated channels properly handling customers with multiple catalogs that have overlapping content.

9.287 [3.55.1]

fix: properly removing update transmission payloads from SAP transmissions before saving completed records.

9.288 [3.55.0]

feat: add *enable_executive_education_2U_fulfillment* to *EnterpriseCustomer*

9.289 [3.54.2]

fix: follow-on to cornerstone learner records foreign keys job

9.290 [3.54.1]

fix: create cornerstone learner audit records with new foreign keys

9.291 [3.54.0]

feat: Remove progress, progress_v2 option from reporting configs and move all v1, v2 to v3.

9.292 [3.53.4]

fix: update course run selection logic for SAP content exporter

9.293 [3.53.3]

feat: mark Cornerstone session token fields read-only in admin

9.294 [3.53.2]

feat: update data sharing consent request language

9.295 [3.53.1]

feat: Django Admin tweaks for integrations

9.296 [3.53.0]

feat: Added management command for weekly nudge to dormant enterprise learners

9.297 [3.52.0]

feat: add *enable_portal_learner_credit_management_screen* to *EnterpriseCustomer*

9.298 [3.51.1]

feat: basic integrated channels task concurrency control

9.299 [3.51.0]

feat: Added command for monthly impact report for enterprise administrators

9.300 [3.50.0]

feat: add *enable_learner_portal_offers* to *EnterpriseCustomer*

9.301 [3.49.10]

fix: append backslash to ecommerce url and change *get* to *get_or_create* in *fulfill_pending_course_enrollments*

9.302 [3.49.9]

feat: add source query param in data sharing consent url

9.303 [3.49.8]

chore: add logs for dsc

9.304 [3.49.7]

fix: parsing improvements to incorrect learner completion completed-at dates

9.305 [3.49.6]

fix: incorrect learner completion completed-at dates

9.306 [3.49.5]

feat: add lms_user_id to serialized admin users

9.307 [3.49.4]

feat: add dry-run mode to integrated channels

9.308 [3.49.3]

fix: don't transmit schedule data to SAP if start or end date is empty

9.309 [3.49.2]

feat: improved channel logging

9.310 [3.49.1]

fix: add stricter validation on system wide role assignments

9.311 [3.49.0]

fix: Return None for context if a `SystemWideEnterpriseUserRoleAssignment` has no `enterprise_customer` and does not apply to all contexts.

We'll no longer fall back on granting context based on enterprise membership when there is no explicit context.

9.312 [3.48.0]

chore: add migration to remove `is_active` from role assignment model schema

9.313 [3.47.2]

chore: remove `is_active` field from role assignment model

9.314 [3.47.1]

chore: remove data-cleaning management commands. prepare for column-removal migration

9.315 [3.47.0]

temp: adding system wide role assignment field and management commands to clean data

9.316 [3.46.6]

fix: correctly handle multiple canvas and blackboard oauth configs

9.317 [3.46.5]

fix: `degreed2` improperly tracking completion status

9.318 [3.46.4]

fix: Degreed2 estimated time to complete in hours

9.319 [3.46.3]

fix: update logic for parsing course price for SAP

9.320 [3.46.2]

fix: Degreed2 estimated time to complete in days

9.321 [3.46.1]

feat: admin view improvements

9.322 [3.46.0]

fix: modify signature of `EmbargoApiClient.redirect_if_blocked` Make this signature match and use the same signature that `openedx.core.djangoapps.embargo.api.redirect_if_blocked()` now uses.

9.323 [3.44.4]

fix: implement back-off and retry for degreed2 fix: drop `tpa_hint` param in redirects when no SSO

9.324 [3.44.3]

fix: Undoes revert of 3.44.0, while also ensuring that `SystemWideEnterpriseUserRoleAssignment.get_assignments()` can handle and respect any null values returned from `get_context()`.

9.325 [3.44.2]

- feat: configure django admin for degreed2 audit records
- fix: Moodle client should accept treat duplicate course id on create as a success

9.326 [3.44.1]

fix: no-op version bump (skipping 3.44.0) to account for a revert: <https://github.com/openedx/edx-enterprise/pull/1534>

9.327 [3.44.0]

fix: [REVERTED] override get_assignments() so that active enterprise uuids come first.

Overrides the SystemWideEnterpriseUserRoleAssignment.get_assignments() method to return a list of (role, context) assignments, where the first item in the list corresponds to the currently active enterprise for the user.

9.328 [3.43.1]

chore: replace enterprise customer drop-downs in django admin

9.329 [3.43.0]

feat: allow admins to remove learners from org

9.330 [3.42.5]

fix: improve guards on fk data backfill job

9.331 [3.42.4]

feat: updated logic for completions in integrated channels

9.332 [3.42.3]

feat: additional fk data backfill performance improvements

9.333 [3.42.2]

feat: speed up fk data backfill

9.334 [3.42.1]

feat: use new foreign keys on integrated channels audit models

9.335 [3.42.0]

feat: add admin_users to EnterpriseCustomerSerializer

9.336 [3.41.13]

fix: remove backfill management command arguments

9.337 [3.41.12]

fix: Use enterprise customer uuid coming in request data

9.338 [3.41.11]

fix: Add unique_together constraint in SystemWideEnterpriseUserRoleAssignment

9.339 [3.41.10]

fix: Add management command to backfill missing audit record foreign keys.

9.340 [3.41.9]

fix: Squash SAP Success Factors migrations to remove reference to PositiveIntegerField.

9.341 [3.41.8]

fix: Alter *enterprise_course_enrollment_id* field from *PositiveIntegerField* to *IntegerField* in *BlackboardLearnerAssessmentDataTransmissionAudit* and *SapSuccessFactorsLearnerDataTransmissionAudit*. This change require to run migrations on mysql8.

9.342 [3.41.7]

fix: add foreign keys to integrated channels audit models

9.343 [3.41.6]

fix: making making degreed token base url optional

9.344 [3.41.5]

feat: add missing logging for grades api results in integrated channels exporter

9.345 [3.41.4]

feat: added html pages for oauth authorization success/failure

9.346 [3.41.3]

fix: allow null completed_timestamp field for integrated channels learner audit models

9.347 [3.41.2]

feat: setting customer identity provider config is_valid on first SSO login

9.348 [3.41.0]

feat: Allow partial_update on *EnterpriseCustomerViewSet*

9.349 [3.40.16]

fix: CSOD Learner Audit Django Admin Timeouts

9.350 [3.40.15]

fix: Use correct completions URL for Degreed2

9.351 [3.40.14]

fix: CornerstoneLearnerDataTransmissionAudit admin view timeout

9.352 [3.40.13]

fix: Degreed2 Missing Learner Data Audit Records

9.353 [3.40.12]

fix: Degreed2 Missing Learner Data Audit Records

9.354 [3.40.11]

feat: New integrated channels Blackboard api endpoint to fetch global config creds

9.355 [3.40.10]

feat: Add drafting functionality to save incorrect fields

9.356 [3.40.9]

feat: new integrated channels customer configs list view, new integrated channels config serializer *is_valid* field

9.357 [3.40.8]

feat: add *enable_browse_and_request* field to *EnterpriseCustomer*

9.358 [3.40.7]

fix: Broken Canvas oauth authorization url

9.359 [3.40.6]

feat: SAPSF content metadata transmission now also sends course schedule

9.360 [3.40.5]

feat: adding CornerstoneLearnerDataTransmissionAudit admin view feat: log if-modified-since + content metadata for CSOD

9.361 [3.40.4]

feat: support filtering by a list of user ids for *EnterpriseCustomerUserViewSet*

9.362 [3.40.3]

feat: show field show_course_price in SAPSF Django admin form

9.363 [3.40.2]

feat: override chunk size default to match channel capability

9.364 [3.40.1]

chore: squash migrations for blackboard and sap_success_factor apps.

9.365 [3.40.0]

fix: Alter *enterprise_course_enrollment_id* field from *PositiveIntegerField* to *IntegerField* in *BlackboardLearnerAssessmentDataTransmissionAudit* and *SapSuccessFactorsLearnerDataTransmissionAudit*. This change require to run migrations on mysql8.

9.366 [3.39.1]

fix: switching blackboard integrated channels from client based auth credentials to global creds

9.367 [3.39.0]

fix: ensure *active* field on EnterpriseCustomerUser objects are set to *False* appropriately feat: add management command to clean up *active* fields on EnterpriseCustomerUser objects

9.368 [3.38.7]

feat: customer configs draft saving that makes all variables optional feat: feat: Add field for Display name for LMS configs

[3.39.0] chore: dropped Django22, 30 and 31 support

9.369 [3.38.6]

feat: add created to enterprise course enrollment serializer fields

9.370 [3.38.5]

fix: update link_learners action to respond with error when payload is empty.

9.371 [3.38.4]

fix: bugfix for Cornerstone missing completion records

9.372 [3.38.3]

fix: more logging to debug missing completion records

9.373 [3.38.2]

fix: Django Admin bugfix

9.374 [3.38.1]

feat: New crud viewset for IC degreed2 configurations

9.375 [3.38.0]

feat: Adds toggle_universal_link endpoint

9.376 [3.37.0]

feat: Dependency upgrades

9.377 [3.36.13]

fix: check if instance is an iterable rather than a list in `EnterpriseCustomerUserReadOnlySerializer`

9.378 [3.36.12]

feat: add enterprise role assignments to `EnterpriseCustomerUserReadOnlySerializer`

9.379 [3.36.11]

fix: Integrated channels Degreed2 exporter now handles invalid start/end date in content metadata item

9.380 [3.36.10]

fix: add *basic_list* action to `EnterpriseCustomerInviteKeyViewSet` to return unpaginated set of invite keys.

9.381 [3.36.9]

feat: new oauth state for multi-lms-configuration

9.382 [3.36.8]

feat: allow more than 1 lms configuration per lms-kind

9.383 [3.36.7]

feat: update *enterprise_customer_invite_key* filter and serializer

9.384 [3.36.6]

feat: Show OAuth Auth link for Blackboard Admin

9.385 [3.36.5]

fix: add support for an *enterprise_customer_invite_key* UUID query parameter to be passed and handled by the *EnterpriseProxyLoginView*

9.386 [3.36.4]

feat: OAuth Auth link for Blackboard Admin

9.387 [3.36.3]

feat: Integrated channels, grade send logic only logs instead of raising when *enterprise_customer_user* record is inactive

9.388 [3.36.2]

feat: add *is_active* on enterprise customer invite key

9.389 [3.36.1]

feat: improved integrated channel log consistency

9.390 [3.36.0]

feat: added view to link learners from a enterprise customer key

9.391 [3.35.3]

fix: require expiry date on EnterpriseCustomerInviteKey model

9.392 [3.35.2]

feat: reformat integrated channels logging to be more splunk friendly

9.393 [3.35.1]

docs: Updating help_text for universal_link field on EnterpriseCustomer model

9.394 [3.35.0]

feat: Adding universal_link field to EnterpriseCustomer model

9.395 [3.34.2]

feat: add try catch block to skip unfound courses

9.396 [3.34.1]

feat: add enterprise customer invite key model and viewset

9.397 [3.34.0]

feat: New management command to revert enrollment data

9.398 [3.33.12]

feat: SAPSF integrated no longer considers grade change as a reason to retransmit completions.

9.399 [3.33.11]

feat: New management command to backfill end dates on Canvas

9.400 [3.33.10]

fix: incorrectly skipping completion transmissions

9.401 [3.33.9]

feat: allow filtering enterprise learners by enterprise uuid and enterprise role

9.402 [3.33.8]

fix: Moodle duplicate content metadata records detected

9.403 [3.33.7]

fix: allow for records to be saved for integrated channels' content across catalogs

9.404 [3.33.6]

fix: CSOD API session tokens bugfix

9.405 [3.33.5]

fix: CSOD API session tokens are now saved to the customer's configuration instead of individual transmission audits

9.406 [3.33.4]

feat: integrated channels only requests content metadata for courses that need updating

9.407 [3.33.3]

feat: Change Bulk Enrollment Assignment Logic for Pending learners

9.408 [3.33.2]

fix: no longer notify learners of already existing enrollments

9.409 [3.33.1]

fix: Rename model field from key to client_id: Degreed2

9.410 [3.33.0]

feat: New Integrated channel Degreed v2.

9.411 [3.32.0]

feat: Added management command to fix DSC records having spaces instead of +.

9.412 [3.31.1]

fix: pip-tools upgrade

9.413 [3.31.0]

feat: new integrated channels content metadata transmitter flow

9.414 [3.30.14]

fix: blackboard logging function was not returning desired string

9.415 [3.30.13]

fix: properly weight blackboard grades

9.416 [3.30.12]

- chore: update course enrollments through lms

9.417 [3.30.11]

- docs: added adr for zero state browsing with universal link

9.418 [3.30.10]

- fix: refactor moodle _post to use body params

9.419 [3.30.9]

- chore: Don't expire courses that have been modified after given date

9.420 [3.30.8]

- feat: Added a boolean in EnterpriseCustomer to specify whether labor market data should be available in learner portal

9.421 [3.30.7]

- update admin banner notification text field with following changes
- increase max_length from 255 to 512
- update help text
- use textarea widget in django admin

9.422 [3.30.6]

- maint: Integrated channels detection system of catalog changes needed is now disabled via override.

9.423 [3.30.5]

- fix: Integrated channels data transforming generates json serializable fields.

9.424 [3.30.4]

- fix: Blackboard integrated channel now correctly synchronizes the one-and-only valid refresh_token

9.425 [3.30.3]

- fix: content_filter in django admin was broken after the jsonfield upgrade, so this contains fix for that

9.426 [3.30.2]

- fix: switch is_revoked to True on LicensedEnterpriseCourseEnrollment after license expiration

9.427 [3.30.1]

- Allowing management commands to optionally run on inactive Integrated Channel configurations

9.428 [3.30.0]

- Switched back to jsonfield from jsonfield2

9.429 [3.29.0]

- Added api for fetching field choices from EnterpriseCustomerReportingConfiguration

9.430 [3.28.24]

- Integrated channels Canvas: now fills in Start/end dates in description, and uses Course participation type

9.431 [3.28.23]

- Fix cornerstone character limit bug with dict database table

9.432 [3.28.22]

- fix: Adding error handling for role assignment backfill management command

9.433 [3.28.21]

- bug: The exporter now properly handles instances when enterprise customer catalogs do not need updates.

9.434 [3.28.20]

- feat: Added enterprise_learner role assignment backfill management command

9.435 [3.28.19]

- fix: additional Moodle field changes

9.436 [3.28.18]

- Added customer config based learner data transmission feature flag

9.437 [3.28.17]

- Improve error logging in the Moodle integration

9.438 [3.28.16]

- Fix import error used by bulk enrollment in utils

9.439 [3.28.15]

- integrated channels: single learner assessment exporter logging is not helpful right now so improve it.

9.440 [3.28.14]

- logging improvement when calling integrated channels `extract_integration_id`

9.441 [3.28.13]

- fixes the way moodle queries for courses ENT-4806

9.442 [3.28.12]

- Integrated channels automatically fill in current date for audit completions if date not available.

9.443 [3.28.11]

- Create “enterprise_learner” role when `EnterpriseCustomerUser` records are re-linked.
- When `EnterpriseCustomerUser` records get deleted, also delete the “enterprise_admin” role specific to the relevant enterprise customer.

9.444 [3.28.10]

- Integrated channel transmitter completions routine now logs as error, any status codes greater than or equal to 400

9.445 [3.28.9]

- Include a `failure_reason=dsc_denied` to the DSC failure url when learner denies the DSC terms.

9.446 [3.28.8]

- SAP integrated channel django form gets missing idp id field

9.447 [3.28.7]

- Degreed integrated channel now uses idp_id explicitly when calling get_remote_id()

9.448 [3.28.6]

- SAP integrated channel now uses idp_id explicitly when calling get_remote_id()

9.449 [3.28.5]

- Fixed datetime issue in email_drip_for_missing_dsc_records.

9.450 [3.28.4]

- Integrated channels: audit track completion status now based on incomplete non-gated content

9.451 [3.28.3]

- Integrated channels: log response code and message if SAP post fails

9.452 [3.28.2]

- Add *progress_v3* report type for enterprise reporting.

9.453 [3.28.1]

- Inject a failure reason into the `failure_url` query params when a verified course mode is not available for DSC-based enrollments.

9.454 [3.28.0]

- Added support for Django 3.0, 3.1 and 3.2

9.455 [3.27.27]

- Adds enterprise catalog query title as an optional attribute to create/update post requests on the catalog service.

9.456 [3.27.26]

- Refactor data-sharing consent GET and POST handlers to not have too many statements, because readability matters.

9.457 [3.27.25]

- Blackboard Integrated channel oauth2 refresh token handling fixes.

9.458 [3.27.24]

- Adding a new EnterpriseEnrollmentSource to be used for bulk enrollment.

9.459 [3.27.23]

- Add logging of user id for troubleshooting in a couple of locations.
- Clean up pylint suppressions and rules using latest rules set by edx-lint.

9.460 [3.27.22]

- Prevent failures on integrated channels delete requests when courses are not found.

9.461 [3.27.21]

- Encode invalid course keys for CSOD customers

9.462 [3.27.20]

- Handle `content_last_modified` not provided by enterprise catalog

9.463 [3.27.19]

- Localize timezones on catalog modified min (not found) values

9.464 [3.27.18]

- Integrated channels util functions needed to base64 urlsafe encode/decode course keys for use with some LMS systems like Cornerstone.

9.465 [3.27.17]

- Integrated channels now checks and uses catalog modified times to determine if an update is needed before retrieving content metadata.

9.466 [3.27.16]

- Making bulk catalog query ID update params optional

9.467 [3.27.15]

- Added title field in `AdminNotification` table.

9.468 [3.27.14]

- Adding the ability to specify parameters in the bulk catalog query ID updated management command.

9.469 [3.27.13]

- Revert 'Start my course' links in bulk enrollment emails to courseware based links instead of learner portal.

9.470 [3.27.12]

- Prevent django admin deletions of catalog queries. Added management command to bulk update catalogs of their query IDs

9.471 [3.27.11]

- Avoid failure when an email send in the learners loop fails, for notify_enrolled_learners

9.472 [3.27.10]

- Use celery tasks for emails sent using EnterpriseCustomer's notify_enrolled_learners method

9.473 [3.27.9]

- Fix SAP Course Completion payload format again.

9.474 [3.27.8]

- Fix SAP Course Completion payload format.

9.475 [3.27.7]

- Replace EnrollmentApiClient calls from Bulk enrollment with a newly minted python api call (non-REST) from edx-platform

9.476 [3.27.6]

- Filter available IDPs for Enterprise Customers by new boolean flag on ProviderConfig model.

9.477 [3.27.5]

- Removing CSOD Integrated Channel from the list of supported channels for the content metadata transmission task.

9.478 [3.27.4]

- Add pagination handling to integrated channels Blackboard client

9.479 [3.27.3]

- Adds flag to SAP Success Factors customer configuration to switch SAP endpoints for learner completion calls.

9.480 [3.27.2]

- Ensure deletion and unlinking of a `EnterpriseCustomerUser` record only deletes the `enterprise_learner` system-wide role for that particular `EnterpriseCustomerUser`, as opposed to all `enterprise_learner` roles associated with the user.

9.481 [3.27.1]

- Updates bulk enrollment email template.

9.482 [3.27.0]

- Added enterprise uuid support in course enrollment. ERTE-5

9.483 [3.26.23]

- Fix the way that `page_size` is passed as a param to the `get_content_metadata` endpoint. Add a unit test for the `EnterpriseCatalogApiClient.get_content_metadata()` method, which was previously untested.

9.484 [3.26.22]

- Set the `EnterpriseCatalogApiClient.get_content_metadata` request `page_size` parameter to 50; the enterprise-catalog service has a default `page_size` of 10. This change means that we'll make a smaller overall number of SELECTs against the enterprise-catalog database.

9.485 [3.26.21]

- Adds error handling and logging to the assignment deduplication management command.

9.486 [3.26.20]

- Updates requirements and style changes to match the latest Pylint.

9.487 [3.26.19]

- Updates to integrated channels catalogs to transmit help text.

9.488 [3.26.18]

- Overriding default chunk size for SAP and Canvas integrations.

9.489 [3.26.17]

- Adds Segment tracking for bulk enrollment method.

9.490 [3.26.16]

- Added history tables for EnterpriseCustomerUser and SystemWideEnterpriseUserRoleAssignment.

9.491 [3.26.15]

- Added management command to clean up duplicate transmitted assignments for the integrated channels.

9.492 [3.26.14]

- Fixed issue with API version in Tableau client.

9.493 [3.26.13]

- Fixed issue with CourseEnrollment receiver when learner has multiple enterprises.

9.494 [3.26.12]

- Canvas integrated channel now supports create_or_update pattern for courses. Detects/logs deleted courses.

9.495 [3.26.11]

- Removed ENABLE_MULTIPLE_USER_ENTERPRISES_FEATURE waffle switch

9.496 [3.26.10]

- Fix forward for parameter rename changing the signals API in 3.26.7

9.497 [3.26.9]

- Added support to use default idp in Enterprise slug login if there are multiple.

9.498 [3.26.8]

- added support for redirecting user to default IDP, in case multiple IDPs's attached

9.499 [3.26.7]

- developer-only facing updates to standardize LMS Integrated Channels logging.

9.500 [3.26.6]

- added an update api call to assign tableau user roles

9.501 [3.26.5]

- fix: Bypass slumber's getattr definition when requesting enrollments for usernames starting with '_' (because slumber will raise an AttributeError from getattr when requesting a resource that starts with '_').

9.502 [3.26.4]

- removed unnecessary call to ecom in bulk enrollment (process of assigning a license already accounts for this)

9.503 [3.26.3]

- added --skip-unlink param in unlink_enterprise_customer_learners command to just remove DSC records.

9.504 [3.26.2]

- Added logs for enterprise users created in tableau.

9.505 [3.26.1]

- Added check to configure reports only for Catalog over SFTP.

9.506 [3.26.0]

- Added support for admin scheduled banners that run from date x to date y.

9.507 [3.25.2]

- Log exception stack trace during DSC licensed-enrollment flow, so that we can look at log messages and understand what exactly is failing.

9.508 [3.25.1]

- bug fix, properly handle API response pagination from Canvas.

9.509 [3.25.0]

- added management command to unlink learners from their enterprise and deleting DSC and EnterpriseCourseEnrollment records.

9.510 [3.24.0]

- added `enable_compression` flag in `EnterpriseCustomerReportConfiguration` table.

9.511 [3.23.12]

- Database based template system for enrollment emails, including support for Admin and Self enroll modes. Admin mode for Bulk enrollment, existing enrollment emails still use the current template.

9.512 [3.23.11]

- Log more specific information about HTTP client errors that are caught when using the LMS enrollment API. Also send an exception event to the monitoring service when this happens, even though we handle the exception “gracefully”.

9.513 [3.23.10]

- Send long dsc url in missing DSC email as individual params.

9.514 [3.23.9]

- Reduced the DSC url size to account for character limit in Segment event properties.

9.515 [3.23.8]

- Remove hardcoded admin permission constraints for `ContentMetadataItemTransmission` integrated channel model.

9.516 [3.23.7]

- Canvas integrated channel now ‘concludes’ course when sending deletion event, instead of ‘delete’.

9.517 [3.23.6]

- Optimized handling of conditions defining the absence of a DSC.

9.518 [3.23.5]

- Added exception handling in consent missing email.

9.519 [3.23.4]

- Added a check for enterprise DSC configuration in missing DSC drip.

9.520 [3.23.3]

- Added a check for course access before sending Segment event for missing DSC.

9.521 [3.23.2]

- Added new field `reply_to` in enterprise customer where learner's reply to enterprise emails will be delivered.
- Removed migrations that have been merged into squashed migrations.

9.522 [3.23.1]

- Fix: filter out `EnterpriseCourseEnrollments` without corresponding `CourseEnrollment` records in learner portal view.

9.523 [3.23.0]

- Added support for `--enrollment-before` and `--no-commit` params in `email_drip_for_missing_dsc_records` command.

9.524 [3.22.16]

- Fixed Segment json string issue for DSC email drip

9.525 [3.22.15]

- Added additional Segment event properties for missing DSC drip email

9.526 [3.22.14]

- Fixed timezone issue in comparison of course start datetime

9.527 [3.22.13]

- Make enterprise customer uuid mandatory for *TableauAuthView*

9.528 [3.22.12]

- Change the verbose name and help text for the `enable_integrated_customer_learner_portal_search` field on the `EnterpriseCustomer` model.

9.529 [3.22.11]

- No longer call into the removed `email_marketing` platform djangoapp

9.530 [3.22.10]

- Use Braze for sending data sharing consent drop emails, add the DSC link inside the drip email.

9.531 [3.22.9]

- Expose enterprise catalog uuids associated with an Enterprise Customer in the `enterprise-customer` API endpoint.

9.532 [3.22.8]

- Add dashboard admin rbac role permission on tableau auth view so that only enterprise dashboard admins can access this view.
- Add support to generate tableau auth token based on incoming enterprise customer's uuid

9.533 [3.22.7]

- chore: upgrade edx-enterprise requirements

9.534 [3.22.6]

- Improves performance of enterprise role assignment admin page
- Deletes custom `get_search_results()` method, since `enterprise_customer__name` is now a viable search field
- Improves pagination by asking for an estimated row count from Mysql `INFORMATION_SCHEMA.TABLES`
- Turns 1 + N query into 1 query via proper use of `list_select_related`

9.535 [3.22.5]

- Fix: no longer stringifying *None* values passed to enterprise catalog creations calls

9.536 [3.22.4]

- Fix: learner_data exporter bug fix and refactor for cleaner enrollment filtering

9.537 [3.22.3]

- Feature: including EnterpriseCatalogQuery UUID field in request payload to enterprise-catalog on Enterprise-Catalog updates

9.538 [3.22.2]

- Feature: new UUID field on EnterpriseCatalogQuery model (and update to all existing query objects)

9.539 [3.22.1]

- Refactor: integrated channels learner exporter replace course api client

9.540 [3.22.0]

- Added a management command to send emails to learners with missing DSC

9.541 [3.21.4]

- allow searching of enterprise customer records with hyphenated uuid
- add typeahead search dropdown to improve enterprise customer search on enterprise reporting configuration

9.542 [3.21.3]

- When a learner is linked from manage learners page, in-activate learner's other enterprises

9.543 [3.21.2]

- Added support of multiple identity_providers in enterprise.models.get_remote_id.

9.544 [3.21.1]

- Added multiple identity_providers in EnterpriseCustomerApi

9.545 [3.21.0]

- Added the ability to link/unlink enterprise customer catalogs with enterprise reporting configuration via its API endpoint.

9.546 [3.20.5]

- Integrated channels learner_data module refactored to avoid making some LMS REST API calls

9.547 [3.20.4]

- Refactored code in *proxied_get()* to clean up duplicate logic.

9.548 [3.20.3]

- Removing unused and out of date endpoints for Bulk Enrollment

9.549 [3.20.2]

- Allow licensed audit enrollment to have a path to upgrade into verified

9.550 [3.20.1]

- update edx-rbac to 1.4.2, plus a bunch of other version bumps.

9.551 [3.20.0]

- feat: add support for enterprise admins to create pending enterprise users

9.552 [3.19.0]

- feat: add support for creating multiple pending enterprise users

9.553 [3.18.7]

- Refactored bulk enrollment serializer and bug fixes to the bulk enrollment endpoint.

9.554 [3.18.6]

- fix: The `update_role_assignments_with_customers` command no longer updates records. It only creates new records, which helps de-risk the operation.

9.555 [3.18.5]

- fix: do not include unpublished courses when enrollment link resolves `course_runs`

9.556 [3.18.4]

- fix: The `update_role_assignments_with_customers` command no longer deletes open assignments. Allowing it to do so left us prone to error when an explicit `enterprise_customer_uuid` arg is provided. We should modify this command in the future to perform deletions of open assignments as its only action, and it should only be invoked this way after we have verified that all backfilled `enterprise_customer` fields on the assignments have been set correctly.

9.557 [3.18.3]

- Adds the catalog admin role to `roles_api.roles_by_name()`.

9.558 [3.18.2]

- Removes course mode as a required parameter to the bulk subscription enrollment endpoint.

9.559 [3.18.1]

- Adds bulk enterprise learner in bulk courses enrollment endpoint with pending user support.

9.560 [3.18.0]

- Adds a management command to update all `SystemWideEnterpriseUserRoleAssignment` records in a way that makes them more explicitly defined.

9.561 [3.17.47]

- Bug fix to remove a deprecated parameter that was causing bulk enrollments to fail.

9.562 [3.17.46]

- Made help text of `sender_alias` more generic.

9.563 [3.17.45]

- Fix bulk enrollment endpoint to process email_csv and email as well

9.564 [3.17.44]

- Replaced an LMS Enrollment API call with direct call the DB to avoid LMS rate limiting during integrated channels bulk jobs.

9.565 [3.17.43]

- Updated the default IDP priority of enterprises for social auth.

9.566 [3.17.42]

- Change canvas_course_id to BigInteger: Integrated Channels

9.567 [3.17.41]

- Upgrade django-ipware to version 3.0.2

9.568 [3.17.40]

- Read CSV files using *utf-8-sig* encoding to handle Byte Order Mark

9.569 [3.17.39]

- Rename *Owners* field to *Partners* for Cornerstone Integration

[3.17.38]

- Omitting assessment level reporting from integrated Canvas learners final grade to not have redundant reported points between final grades and subsection grades.

9.570 [3.17.37]

- Refactor to only create an `EnterpriseCourseEnrollment` if we successfully create/update a `CourseEnrollment` record

9.571 [3.17.36]

- Properly filtering integrated channels that support assessment level reporting.

9.572 [3.17.35]

- Map “estimated_hours” to “credit_hours” in addition to “total_hours” in SAP.

9.573 [3.17.34]

- Removing temporary logs from integrated channels.

9.574 [3.17.33]

- Enable manually adding learners to multiple enterprises

9.575 [3.17.32]

- Adding the logic to select default provider in case an enterprise has multiple identity providers attached.

9.576 [3.17.31]

- Change moodle course title in exporter, to include edX text.

9.577 [3.17.30]

- Investigatory logging to track down Integrated Channels transmission issues.

9.578 [3.17.29]

- Prevent NoneType string concatenation when handling multiple enterprises logistration without redirects.

9.579 [3.17.28]

- Adds default field in enterprise customer identity provider table to select default IDP if there are more than one IDPs attached with enterprise.

9.580 [3.17.27]

- Adding Logging to single learner assessment level reporting task.

9.581 [3.17.26]

- Updating docs to reflect method behaviors.

9.582 [3.17.25]

- Making failed SAP user remote ID retrievals log relevant context data.

9.583 [3.17.24]

- Making sure Canvas Integrated Channel properly url encodes user identifier fields.

9.584 [3.17.23]

- Fixing assessment level reporting audit retrieval.

9.585 [3.17.22]

- Adds content metadata item transmission table to Django Admin.

9.586 [3.17.21]

- Introduce and use a `roles_api` module and use the roles API in signal receivers that need to create or delete role assignments.
- For created or updated learner and admin enterprise users, associate their user-role with the `enterprise_customer` to which that user is linked.
- Install `django-cache-memoize`.

9.587 [3.17.20]

- Adds better exception handling to the SAP integrated channels.
- Adds better logging to the base transmission process in the integrated channels.

9.588 [3.17.19]

- Removes the `sync_enterprise_catalog_query` boolean field from the `EnterpriseCustomerCatalog` model.
- Adds migration to remove the `sync_enterprise_catalog_query` boolean field.

9.589 [3.17.18]

- Removes all references to the `sync_enterprise_catalog_query` boolean field from the `EnterpriseCustomerCatalog` model.
- Updates all conditional use of the `sync_enterprise_catalog_query` field to be `True`.
- A second PR will follow to remove the model field and perform the db migration (blue/green deployment safe).

9.590 [3.17.17]

- Added a catch all exception block to ensure login flow is not interrupted by analytics user sync.

9.591 [3.17.16]

- Include `course mode` for the user's `student.CourseEnrollment` in the `EnterpriseCourseEnrollmentSerializer`.

9.592 [3.17.15]

- In `SystemWideEnterpriseUserRoleAssignment`, Use either `applies_to_all_contexts` or `enterprise_customer` if they are `True` or non-null, respectively, in determining the result of `get_context()`, but continue to return list of all linked enterprise customer UUIDs if not, (which is the current behavior). This is a small step on our journey to explicitly defining user-role assignments.

9.593 [3.17.14]

- On the `SystemWideEnterpriseUserRoleAssignment` model, adds an `enterprise_customer` FK (nullable) and an `applies_to_all` boolean field (defaults to `False`) that indicates if the user has wildcard permissions.
- Updates the admin to show the “effective” customer in the detail view, and the explicit value in the list view. The effective value is the deprecated way we currently determine role assignment - by implicitly assigning the role on every customer to which the user is linked.
- In the detail view/form, the “Enterprise customer” dropdown contains only customers to which the user is currently linked.

9.594 [3.17.13]

- added check to make sure enterprise user can only use linked IdP with their enterprise customer.

9.595 [3.17.12]

- Conditionally allows the deletion of individual `EnterpriseCourseEnrollment` and related `LicensedEnterpriseCourseEnrollment` records via the Django Admin site, so that site admins can manually delete enterprise enrollments that were created in error. This is only allowed if a Django settings feature flag is set to `True`.

9.596 [3.17.11]

- Apply `edx-rbac` migration to add `applies_to_all_contexts` field to `SystemWideEnterpriseUserRoleAssignment`.
- Added endpoints for Cornerstone integrated channel.

9.597 [3.17.10]

- added home page logo for `EnterpriseSelectionView` and `EnterpriseLoginView`

9.598 [3.17.9]

- Fix deprecation warning: `third_party_auth` should be imported as `common.djangoapps.third_party_auth`.

9.599 [3.17.8]

- Added new API endpoints for Degreed integrated channel.

9.600 [3.17.7]

- Added new field `sender_alias` in enterprise customer which will be used in emails except of default alias.

9.601 [3.17.6]

- Non-effectual code cleanup / refactor to remove some final pieces of duplication (canvas, blackboard).

9.602 [3.17.5]

- Ensure enterprise course enrollments return valid course run statuses such that when a learner earns a passing certificate, the `enterprise_course_enrollments` API endpoint deems the course is complete even though the course itself may not have ended yet per the configured dates.

9.603 [3.17.4]

- Add some info to the `EnterpriseCourseEnrollment` docstring, add `is_active` property to same.

9.604 [3.17.3]

- Fixed unnecessary integrated channel signal transmission on course completion to inactive customers by adding guard condition.

9.605 [3.17.2]

- Stop listening for `student.CourseEnrollment` unenrollment signal, as introduced in 3.17.0

9.606 [3.17.1]

- Add management command to process expired subscriptions and field on subscriptions to persist that the subscription expiration has been processed

9.607 [3.17.0]

- Listen for `student.CourseEnrollment` `unenrollment` signal and delete associated `EnterpriseCourseEnrollment` record if one exists (we will have a historical record of the deletion).

9.608 [3.16.11]

- Retrieve `EnterpriseCustomerUser` by both `user_id` and `enterprise_customer` to handle users who are pending for more than 1 enterprise.

9.609 [3.16.10]

- Forcing embedded enrollment links within integrated Blackboard courses to open new windows to avoid security alert prompt.

9.610 [3.16.9]

- Upgrade celery to 5.0.4

9.611 [3.16.8]

- Added `ClientError` exception handling for `SAPSuccessFactorsAPIClient`.

9.612 [3.16.7]

- Modify the learner portal `enterprise_course_enrollments` endpoint to include an `is_enrollment_active` key that indicates the status of the enterprise enrollment's related `student.CourseEnrollment`. Allow the endpoint to optionally accept an `is_active` query param, so that clients may request only active enrollments from it.

9.613 [3.16.6]

- Improved error handling for SAP Success Factors OAuth2 response.

9.614 [3.16.5]

- Refactoring title content metadata in integrated course creation within the Blackboard integrated channel.

9.615 [3.16.4]

- Add SuccessFactors Customer Configuration API endpoint.

9.616 [3.16.3]

- Update unique constraints for pending Enterprise learners/admins to support users who may be pending for more than 1 Enterprise.
- Fix `handle_user_post_save` to account for the potential of being a pending learner/admin for more than 1 Enterprise.

9.617 [3.16.2]

- Refactor `handle_user_post_save` to be responsible for linking `PendingEnterpriseCustomerUser` records and granting admin permissions.

9.618 [3.16.1]

- Adding backend support for admin portal Blackboard configuration.

9.619 [3.16.0]

- Added the ability to enable multiple Identity Providers for a single enterprise customer.

9.620 [3.15.0]

- Converted relation between enterprise customer and identity provider to a one-to-many.

9.621 [3.14.1]

- Adds new API for Canvas LMS configurations.

9.622 [3.14.0]

- Rebranding update: Change fonts and colors, change mobile layout

9.623 [3.13.12]

- Adding decorators to missed integrated channel tasks.

9.624 [3.13.11]

- Add new API for external LMS configurations.

9.625 [3.13.10]

- Use logo from `get_platform_logo_url` in the legacy Django templates

9.626 [3.13.9]

- Adding Blackboard support for assessment level reporting in the integrated channels.

9.627 [3.13.8]

- Bug fix with course key lookup in the Canvas assessment level grade reporting flow.

9.628 [3.13.7]

- Rebranding update: move to more robust `get_platform_logo_url` and update default branding colors.

9.629 [3.13.6]

- Add log for enterprise enrollment page.

9.630 [3.13.5]

- Fixed deprecation warnings related with drf methods (detail_route, list_route).

9.631 [3.13.4]

- Empty sequence bugfix in catalog api.

9.632 [3.13.3]

- Course end date bugfix.

9.633 [3.13.2]

- Add course end date to course level metadata.

9.634 [3.13.1]

- Base implementation of assessment level reporting for Integrated Channels.

9.635 [3.13.0]

- Use full paths for edx-platform/common/djangoapps imports, as described in [edx-platform ADR #7](#).

9.636 [3.12.4]

- Fix silent exception in catalog api call.

9.637 [3.12.3]

- Add code_owner custom attribute for celery tasks.

9.638 [3.12.2]

- Refresh catalog metadata on create and update

9.639 [3.12.1]

- added support for grade, completion and course_structure type reports in enterprise report configurations. Added validation to allow these reports for Pearson enterprises only.

9.640 [3.12.0]

- Support uploading a course_id column in the “Manage Learners” CSV bulk upload to allow manual enrollments in multiple courses at once.

9.641 [3.11.1]

- Fixes the issue where user preference value can not be null.

9.642 [3.11.0]

- Added spanish translations for data sharing consent page.

9.643 [3.10.5]

- Update Moodle integration to single transmission to handle responses properly.

9.644 [3.10.4]

- Remove hyphens from enterprise_customer_uuid for admin user creation and tableau authentication.

9.645 [3.10.3]

- Fix timeout on update.

9.646 [3.10.2]

- Updated the logic to clear enterprise learner language in a way that db lock does not happen.

9.647 [3.10.1]

- change username with enterprise_customer_uuid for tableau trusted authentication and tableau user creation.

9.648 [3.10.0]

- Tests only: upgrade to pytest 6+ and factoryboy 3+ to bring up to date with edx-platform.

9.649 [3.9.13]

- Adding Blackboard customization to integrated channel content metadata creation.

9.650 [3.9.12]

- change username with user_id for tableau trusted authentication and tableau user creation.

9.651 [3.9.11]

- add logs to know if data sharing consent is failing because catalog does not contain the course

9.652 [3.9.10]

- added POST enterprise-customer/<uuid>/enterprise_learner endpoint to mimic Manage Learners admin form functionality

9.653 [3.9.9]

- upgrade version to create new release on pypi.

9.654 [3.9.8]

- added error_codes in the logging/error messages for the CourseEnrollmentView for better debugging capability.

9.655 [3.9.7]

- Unset learners language so that default_language from enterprise customer may take effect.

9.656 [3.9.6]

- Fix DSC tests to verify enrolling a learner with a license_uuid

9.657 [3.9.5]

- ENT-2450: Add action to kick off jobs to refresh enterprise catalogs so changes will be immediately visible

9.658 [3.9.4]

- Style/UX changes for Moodle integration.

9.659 [3.9.3]

- Adding integrated course customization for Blackboard courses.

9.660 [3.9.2]

- Re-add check for license uuid when enrolling learners into a course

9.661 [3.9.1]

- Added the EnterpriseAnalyticsUser model and tableau integration functions.

9.662 [3.9.0]

- Enable enterprise to have a default language configuration for its learners.

9.663 [3.8.43]

- ENT-3557: Improve blackboard view logging to better report root cause of auth failure.

9.664 [3.8.42]

- ENT-3460: Adding properties to safely use branding config.

9.665 [3.8.41]

- Embedded enterprise in the username was removed for tableau trusted authentication.

9.666 [3.8.40]

- Bug fix: SAML stripping for unlinking was not properly removing saml prefix.

9.667 [3.8.39]

- Blackboard client update/delete and unit tests.

9.668 [3.8.38]

- Reverting changes to EnterpriseCustomerBrandingConfig.

9.669 [3.8.37]

- Using python properties for EnterpriseCustomerBrandingConfiguration colors.

9.670 [3.8.36]

- Authenticate user with Tableau.

9.671 [3.8.35]

- Add default branding config object to the Customer record if null.

9.672 [3.8.34]

- Implementing Blackboard completion data transmission.

9.673 [3.8.33]

- During license revocation, if no audit track exists for the course, attempt to unenroll the learner from it.

9.674 [3.8.32]

- Catches/Handles error occurring with Moodle integrated channel.

9.675 [3.8.31]

- Refactors the revoke endpoint into smaller parts, so that implementing new logic is easier to manage.

9.676 [3.8.30]

- Moodle client bug fix

9.677 [3.8.29]

- Make email field optional for sftp delivery for enterprise reporting config

9.678 [3.8.28]

- Blackboard exporter

9.679 [3.8.27]

- Update `get_service_usernames()` to read from a list variable (that may not exist).

9.680 [3.8.26]

- Moodle completion data implementation

9.681 [3.8.25]

- Blackboard client OAuth2 implementation

9.682 [3.8.24] 2020-10-02

- Allow learners to enroll with their license in courses when DSC is disabled.

9.683 [3.8.23] 2020-10-01

- Added Audit grade for Audit mode enrollments in integrated channels.

9.684 [3.8.22]

- Updated `seed_enterprise_devstack_data` to enable the test customer's subscription management screen

9.685 [3.8.21] 2020-09-28

- Add functionality to save logo file at only one location when saving EnterpriseCustomerBrandingConfiguration instance

9.686 [3.8.20] 2020-09-24

- Better exception handling for integrated channels.

9.687 [3.8.19] 2020-09-24

- Copy test from edx-platform over to enterprise to test migrations early.

9.688 [3.8.18] 2020-09-23

- Initial setup for Blackboard Integrated Channel.

9.689 [3.8.17] 2020-09-23

- Update logo name and path after the instance is saved to replace None with instance id.

9.690 [3.8.16] 2020-09-22

- Token expiration handling in canvas client.

9.691 [3.8.15] 2020-09-22

- Update Data Sharing Consent language.

9.692 [3.8.14] 2020-09-21

- Add Moodle integration to integrated_channels.

9.693 [3.8.13] 2020-09-20

- Fix issue with canvas channel not finding a course, by using search endpoint

9.694 [3.8.12] 2020-09-21

- Fix column width issue for DSC and other pages

9.695 [3.8.11] 2020-09-18

- Upgrading celery version to 4.4.7 for python 3.8 support

9.696 [3.8.10] 2020-09-17

- Reverting PR #952.

9.697 [3.8.9] 2020-09-16

- Standardizing log format in integrated channels learner data export.

9.698 [3.8.8] 2020-09-15

- Fixing the construction of the next param in the proxy login view for SSO.

9.699 [3.8.7] 2020-09-15

- Adding more informative logs to the integrated channels.

9.700 [3.8.6] 2020-09-15

- Using viewname in reverse as part of args to prevent IndexError exception

9.701 [3.8.5] 2020-09-14

- Add a field to EnterpriseCustomer to disable main menu navigation for integrated channel customer users.

9.702 [3.8.4] 2020-09-14

- Add a field for enabling analytics screen in the admin portal for an EnterpriseCustomer.

9.703 [3.8.3] 2020-09-14

- Add management command to create DSC records.

9.704 [3.8.2] 2020-09-11

- Course and Course Run enrollment_url now points to learner portal course page if LP enabled.

9.705 [3.8.1] 2020-09-10

- Canvas channel discovery improvements assorted changes.

9.706 [3.8.0] 2020-09-09

- Assign “enterprise_admin” system-wide role to pending admin users when registering their user account.

9.707 [3.7.8] 2020-09-09

- Fixes migration mismatch for Canvas models.

9.708 [3.7.7] 2020-09-04

- The seed_enterprise_devstack_data management command now accepts an enterprise name when creating an enterprise, and the learner portal is activated by default.

9.709 [3.7.6] 2020-09-09

- Adds the learner data exporter and transmitter to the Canvas integrated channel.

9.710 [3.7.5] 2020-09-08

- Celery version is now upgraded to latest one

9.711 [3.7.4] 2020-09-04

- Adds support to capture contract discounts from the Enrollment API by adding `default_contract_discount` to the `EnterpriseCustomer` model and passing it to ecommerce when creating orders

9.712 [3.7.3] 2020-09-01

- Override the `EnterpriseContentCatalog.save()` method to sync the `content_filter` from an associated `EnterpriseCatalogQuery`, if appropriate.
- Add some logging to the `update_enterprise_catalog_query` signal.

9.713 [3.7.2] 2020-09-01

- The `seed_enterprise_devstack_data` management command is now idempotent when creating an enterprise, and creates users and operator roles for the license-manager and enterprise-catalog workers.

9.714 [3.7.1] 2020-08-28

- Also send course `image_url` to Canvas when creating course.

9.715 [3.7.0] 2020-08-27

- Fixed Duplicate Calls to OCN API.

9.716 [3.6.9] 2020-08-26

- Return requested user's linked enterprises only. For staff user return all enterprises.

9.717 [3.6.8] 2020-08-26

- Added course update and deletion capabilities to the canvas integrated channel.

9.718 [3.6.7] 2020-08-26

- Changed strings in Manage Learners DSC view.

9.719 [3.6.6] 2020-08-24

- Added a fix for “Manual Order Not Fulfilled” bug.

9.720 [3.6.5] 2020-08-24

- Added course mode in ecommerce manual enrollment API.

9.721 [3.6.4] 2020-08-18

- Canvas transmitter implementation for course creation

9.722 [3.6.3] 2020-08-19

- Adding Django admin forms for Canvas integration config and cleanup on models.

9.723 [3.6.2] 2020-08-17

- Adding Canvas integrated channels API endpoint for the oauth process completion

9.724 [3.6.1] 2020-08-17

- Added logging in enrollment endpoint for test purposes.

9.725 [3.6.0] 2020-08-12

- ENT-2939: removing waffle flag and utility function used in enterprise-catalog transition

9.726 [3.5.4] 2020-08-12

- Fixed date format in Cornerstone catalog sync call

9.727 [3.5.3] 2020-08-11

- Fix permissions issue with license_revoke endpoint in LicensedEnterpriseCourseEnrollmentViewSet.

9.728 [3.5.2] 2020-08-11

- Add Content Metadata Exporter for Canvas Integration.

9.729 [3.5.1] 2020-08-11

- Add client instantiation and oauth validation for Canvas integration.

9.730 [3.5.0] 2020-08-10

- Add *update_course_enrollment_mode_for_user* method to the EnrollmentApiClient.
- Create new API endpoint to update the mode for a user's licensed enterprise course enrollments when their enterprise license is revoked.
- Introduce new course run status for *saved_for_later*.
- On revocation of an enterprise license, mark the user's licensed course enrollments as *saved_for_later* and *is_revoked*.

9.731 [3.4.40] 2020-08-05

- Create fresh migrations from scratch for Canvas since this app is yet to run migrations in platform.

9.732 [3.4.39] 2020-08-04

- Remove field 'key' from a canvas integrated_channel model (but not migration yet), step 2/3

9.733 [3.4.38] 2020-08-04

- Migration to remove banner_border_color and banner_background_color branding config fields.

9.734 [3.4.37] 2020-08-04

- Add new field client_id to canvas model for removing older key field (step 1/3)

9.735 [3.4.36] 2020-08-04

- Remove references to deprecated banner_border_color and banner_background_color branding config fields.

9.736 [3.4.35] 2020-08-04

- Add postman collection for Canvas integrated channel

9.737 [3.4.34] 2020-08-03

- Migration to copy old color field values to new field.

9.738 [3.4.33] 2020-08-03

- Add BrandingConfiguration primary/secondary/tertiary color fields.

9.739 [3.4.32] 2020-07-31

- Add Canvas integrated_channel first cut.

9.740 [3.4.31] 2020-07-30

- The PendingEnterpriseCustomerUser create action will create an EnterpriseCustomerUser if an auth.User record with the given user_email already exists.

9.741 [3.4.30] 2020-07-29

- Add flag to sync updates in an EnterpriseCatalogQuery with its associated EnterpriseCustomerCatalogs.
- Create a post_save signal to overwrite the content_filter with the update.
- Changes should also be sent to the Enterprise Catalog service.

9.742 [3.4.29] 2020-07-29

- Added new view for requesting the DSC for learners for specific course.

9.743 [3.4.28] 2020-07-24

- Add query params to proxy login redirect for new welcome template to be rendered.
- Fixing proxy_login SSO redirect, adding default next param from proxy_login.

9.744 [3.4.27] 2020-07-23

- Adds hide_course_original_price field to the serializer for the EnterpriseCustomer endpoint.

9.745 [3.4.26] 2020-07-20

- Adds proxy login view to allow unauthenticated enterprise learners to login via existing flow from the learner portal.

9.746 [3.3.26] 2020-07-17

- Uses correct course mode slugs during enrollment from GrantDataSharingPermissions.

9.747 [3.3.25] 2020-07-16

- Use the GrantDataSharingPermissions view to enroll licensed users in courses

9.748 [3.3.24] 2020-07-15

- Remove get_due_dates and always return an empty list for due_dates

9.749 [3.3.23] 2020-07-13

- Remove unnecessary data migration

9.750 [3.3.22] 2020-07-13

- Final removal of marked_done field

9.751 [3.3.21] - 2020-07-10

- Gracefully handle when list of subjects for content metadata contains either a list of strings and list of dictionaries

9.752 [3.3.20] - 2020-07-09

- Added new SAML Config option to EnterpriseCustomer in Django admin.

9.753 [3.3.19] - 2020-07-08

- Remove database references to marked_done.

9.754 [3.3.18] - 2020-07-07

- Admin dashboard rules predicates now pass an object into the edx-rbac utility functions.

9.755 [3.3.17] - 2020-07-07

- Created `LicensedEnterpriseCourseEnrollment`.

9.756 [3.3.16] - 2020-07-02

- Change `marked_done` on `EnterpriseCourseEnrollment` mode nullable.

9.757 [3.3.15] - 2020-06-30

- Added health checks for enterprise service.

9.758 [3.3.14] - 2020-06-30

- Added `saved_for_later` field to the `EnterpriseCourseEnrollment` model. This will eventually replace the `marked_done` field.

9.759 [3.3.13] - 2020-06-29

- Changed `GrantDataSharingPermission` to redirect to the intended course instead of dashboard, if consent is not required

9.760 [3.3.12] - 2020-06-27

- Repair invalid key references in Discovery API Client method.

9.761 [3.3.11] - 2020-06-25

- Restore `EnterpriseCatalogQuery` functionality to previous state.

9.762 [3.3.10] - 2020-06-24

- xAPI: Include course UUID in activity extensions collection

9.763 [3.3.9] - 2020-06-24

- Remove verbose names from EnterpriseCourseEnrollment model Meta class

9.764 [3.3.8] - 2020-06-23

- Add support to override enrollment attributes for learners

9.765 [3.3.7] - 2020-06-19

- Bug fix: Added missing migration for content_filter validation changes.

9.766 [3.3.6] - 2020-06-17

- Add validation for content_filter subfields in EnterpriseCatalogQuery and EnterpriseCustomerCatalog

9.767 [3.3.5] - 2020-06-17

- Update processing of marked_done field slightly for cleaner boolean usage in client

9.768 [3.3.4] - 2020-06-15

- Update GrantDataSharingPermissionView to accept both; course_run_id as well as course_key

9.769 [3.3.3] - 2020-06-12

- Exclude unpublished course runs when determining available/enrollable status

9.770 [3.3.2] - 2020-06-10

- Added status key to default content filter for EnterpriseCustomerCatalog.

9.771 [3.3.1] - 2020-06-10

- Added marked_done field in /enterprise_course_enrollments/ response

9.772 [3.3.0] - 2020-06-09

- xAPI Integrated Reporting Channel, Version 2

9.773 [3.2.22] - 2020-06-09

- Added rollback for EnterpriseCourseEnrollment enroll

9.774 [3.2.21] - 2020-06-03

- Downgrade an error log to a warning to reduce alert noise

9.775 [3.2.20] - 2020-06-01

- Suppress the 404 exception in get_enterprise_catalog when we expect it
- Add enterprise_customer_uuid to an error message to be more informative
- Delete “enterprise_learner” role assignment when an EnterpriseCustomerUser record is soft deleted (i.e., *linked* attribute is False)
- Update seed_enterprise_devstack_data command to include name on user profiles when creating enterprise users

9.776 [3.2.19] - 2020-06-01

- Updating the catalog preview URL to use the Catalog Service

9.777 [3.2.18] - 2020-05-28

- Added the enterprise slug login functionality.

9.778 [3.2.17] - 2020-05-27

- Improve xAPI enrollment/completion event filtering, transmitting, and recording

9.779 [3.2.16] - 2020-05-27

- Removing caniusepython3 as it is no longer needed since python3 upgrade.

9.780 [3.2.15] - 2020-05-26

- Improve EnterpriseRoleAssignment exception messaging

9.781 [3.2.14] - 2020-05-19

- Converting UUID fields to string for use in can_use_enterprise_catalog

9.782 [3.2.13] - 2020-05-15

- Added can_use_enterprise_catalog utility function to exclude enterprises from the transition to enterprise-catalog

9.783 [3.2.12] - 2020-05-13

- Created migration to *update_or_create* a system-wide enterprise role named *enterprise_catalog_admin*

9.784 [3.2.11] - 2020-05-12

- Moving the post model save logic for Enterprise Catalog to signals.py.

9.785 [3.2.10] - 2020-05-08

- Updated EnterpriseCustomerCatalogAdmin save hook to check if a corresponding catalog exists in the enterprise-catalog service. If it does, the save hook will update the existing catalog; otherwise, a new catalog will be created.
- Added extra logging when syncing Enterprise Catalog data to the Enterprise Catalog Service.

9.786 [3.2.9] - 2020-05-08

- Added a flag to enable the slug login for an enterprise customer.

9.787 [3.2.8] - 2020-05-07

- Makes the data sharing consent template guard against empty/null branding configuration logo values.

9.788 [3.2.7] - 2020-05-07

- Added extra logging in 'create_enterprise_course_enrollments' management command.

9.789 [3.2.6] - 2020-05-06

- Added use of `traverse_pagination` for `get_content_metadata` in the `enterprise_catalog` api client.

9.790 [3.2.5] - 2020-05-06

- Pass enterprise customer's name to enterprise-catalog service during create/update of enterprise catalogs
- Refactor `migrate_enterprise_catalogs` management command to check if a catalog already exists in the enterprise-catalog service. If a catalog already exists, it will be updated with a PUT request; otherwise, a new catalog will be created with a POST request.

9.791 [3.2.4] - 2020-05-06

- Specified python3.5 version for PyPI release

9.792 [3.2.3] - 2020-05-06

- Removed support for Django<2.2 & Python3.6
- Added support for python3.8.
- Changes to use catalog query content filter if defined instead of catalog content filter.

9.793 [3.2.2] - 2020-05-05

- Made enrollment reason optional when linking learners without enrollment.

9.794 [3.2.1] - 2020-05-04

- Added extra logging in 'create_enterprise_course_enrollments' management command.

9.795 [3.2.0] - 2020-04-23

- Squashed the sap_success_factors and integrated_channel app migrations.

9.796 [3.1.3] - 2020-04-23

- Revised “end date” window for determining course active/inactive status in catalog API responses.

9.797 [3.1.2] - 2020-04-21

- Added extra exception handling in *create_enterprise_course_enrollments* management command.

9.798 [3.1.1] - 2020-04-20

- removed get_cache_key and using it from edx-django-utils.

9.799 [3.1.0] - 2020-04-14

- Squashed the enterprise app migrations.

9.800 [3.0.15] - 2020-04-14

- Fixed HTML tags bug from short course description in enterprise course enrollment page

9.801 [3.0.14] - 2020-04-10

- Fixing the traversal of results in `get_content_metadata` for the enterprise-catalog API client

9.802 [3.0.13] - 2020-04-10

- Switch `catalog_contains_course` method to use enterprise catalog service behind waffle sample

9.803 [3.0.12] - 2020-04-10

- Add `USE_ENTERPRISE_CATALOG` waffle sample, and remove `USE_ENTERPRISE_CATALOG` waffle flag
- Switch the use of `waffle.flag_is_active` to `waffle.sample_is_active`
- Updates the `EnterpriseCatalogApiClient` to make the user argument optional. If the user argument is not provided, it will use the “enterprise_worker” user instead
- No longer passes user to the `EnterpriseCatalogApiClient` during initialization in places where a request and/or user object doesn’t already exist

9.804 [3.0.11] - 2020-04-10

- Fix issue with matching urls for redirect to enterprise selection page

9.805 [3.0.10] - 2020-04-08

- Use the `USE_ENTERPRISE_CATALOG` waffle flag for transitioning integrated channels to using the enterprise-catalog service

9.806 [3.0.9] - 2020-04-08

- Add `USE_ENTERPRISE_CATALOG` waffle flag
- Switch `get_course`, `get_course_run`, `get_program`, and `get_course_and_course_run` methods to use enterprise catalog service behind waffle flag

9.807 [3.0.8] - 2020-04-08

- Converted the `EnrollmentApiClient` to JWT client.

9.808 [3.0.7] - 2020-04-07

- Additional xAPI transmission workflow logging

9.809 [3.0.6] - 2020-04-06

- Added support for bypassing enterprise selection page for enrollment url triggered login

9.810 [3.0.5] - 2020-03-31

- Added “active” key in enterprise_catalog API for “course” content_type if the “course” has “course_run” available for enrollment.

9.811 [3.0.4] - 2020-03-31

- Removed the ‘EDX_API_KEY’ from CourseApiClient.

9.812 [3.0.3] - 2020-03-27

- Updated enterprise-catalog endpoint urls to match rename

9.813 [3.0.2] - 2020-03-26

- Improved xApi logging to include statement and LRS endpoint’

9.814 [3.0.1] - 2020-03-18

- Updated xApi integrated channel to use the updated CourseOverview method ‘get_from_ids()’

9.815 [3.0.0] - 2020-03-16

- Removed use of Bearer Authentication

9.816 [2.5.5] - 2020-03-13

- Add field for enabling subscription management screen in the admin portal to EnterpriseCustomer.

9.817 [2.5.4] - 2020-03-12

- Reset authentication cookies on enterprise selection to update JWT cookie with user's enterprise

9.818 [2.5.3] - 2020-03-11

- Added the salesforce opportunity_id in manage learner django admin.

9.819 [2.5.2] - 2020-03-10

- Fixed formatting on JSON fields in django admin forms

9.820 [2.5.1] - 2020-03-05

- Added new data type for enterprise report configurations

9.821 [2.5.0] - 2020-03-03

- Removing enterprise_learner_portal_hostname from ent cust model (including api)

9.822 [2.4.2] - 2020-02-27

- Removed the code for enrolling the program from manage learner django admin panel.

9.823 [2.4.1] - 2020-02-26

- Update log level from INFO to DEBUG for transmit_content_metadata management command

9.824 [2.4.0] - 2020-02-25

- Restricted PendingEnterpriseCustomerUser to be linked with only one EnterpriseCustomer at a time

9.825 [2.3.9] - 2020-02-17

- Added discount percentage support in pending enrollment use case.

9.826 [2.3.8] - 2020-02-10

- Added totalHours field for successfactors completion event

9.827 [2.3.7] - 2020-02-07

- Learner attached to multiple enterprises, logging in via SSO should be taken to Enterprise selection page

9.828 [2.3.6] - 2020-02-06

- Fixed learner data transmission command when grades API return *user_not_enrolled* error

9.829 [2.3.4] - 2020-02-04

- Remove totalHours field from content metadata export

9.830 [2.3.3] - 2020-02-03

- Added exception handling for enrollment api calls during manual enrollment

9.831 [2.3.2] - 2020-01-31

- Adding contact_email to enterprisecustomer admin form

9.832 [2.3.1] - 2020-01-29

- Updated calls to *manual enrollments api* to include enterprise customer info

9.833 [2.3.0] - 2020-01-29

- Add soft deletion support for EnterpriseCustomerUser model

9.834 [2.2.0] - 2020-01-28

- Adding new fields to EnterpriseCustomer and EnterpriseCustomerBrandingConfiguration models

9.835 [2.1.7] - 2020-01-28

- Revert Edx-API-Key-replacement-changes

9.836 [2.1.6] - 2020-01-27

- Updating enterprise catalog migration management command

9.837 [2.1.5] - 2020-01-27

- Added totalHours field for successfactors export

9.838 [2.1.4] - 2020-01-24

- add boolean field to track linked/unlinked EnterpriseCustomerUser records

9.839 [2.1.03] - 2020-01-24

- Code refactor and ability to send learner completion if grade is changed

9.840 [2.1.01] - 2020-01-21

- Initialized EnrollmentApiClient with enterprise service worker user

9.841 [2.1.0] - 2020-01-16

- Added hooks to sync EnterpriseCustomerCatalog creation, deletion, and model updates in Django Admin to the new enterprise-catalog service

9.842 [2.0.50] - 2020-01-16

- Replaced EnrollmentApiClientJwt name back to original client's name.

9.843 [2.0.49] - 2020-01-15

- Added management command to reset SAPSF completion data.

9.844 [2.0.48] - 2020-01-14

- Updated enterprise catalog client json formatting.

9.845 [2.0.47] - 2020-01-13

- Replaced Edx-API-Key in the remaining endpoints of EnrollmentApiClient

9.846 [2.0.46] - 2020-01-10

- Introduced management command to migrate enterprise catalog data to new service.

9.847 [2.0.45] - 2020-01-09

- ENT-2489 | Extracting JSON from discovery service response to calculate size

9.848 [2.0.43] - 2020-01-08

- Replaced Edx-API-Key in the ThirdPartyAuthApiClient
- Changed the client in one endpoint of ThirdPartyAuthApiClient
- Endpoint name: model-EnterpriseCustomerUser

9.849 [2.0.42] - 2020-01-07

- Updated context for user with multiple linked enterprises

9.850 [2.0.41] - 2020-01-06

- Added enterprise discount percentage in a manual enrollment

9.851 [2.0.40] - 2020-01-06

- Replaced Edx-API-Key in the EnrollmentApiClient
- Changed the client in one endpoint of EnrollmentApiClient
- Endpoint name: admin-views-EnterpriseCustomerManageLearnersView

9.852 [2.0.39] - 2020-01-06

- Replaced Edx-API-Key in the CourseApiClient
- Changed the client in one endpoint of CourseApiClient
- Endpoint name: exporters-learnerdata

9.853 [2.0.38] - 2020-01-02

- Changed logging of response size from 2.0.37 (ENT-2489) to use size of response in bytes

9.854 [2.0.37] - 2020-01-02

- Added logging of response size when requests are made to discovery service for data not in cache

9.855 [2.0.36] - 2019-12-30

- Use *edx-tincan-py35* PYPI package instead of downloading via git

9.856 [2.0.35] - 2019-12-30

- Version upgrade for edx-rbac

9.857 [2.0.34] - 2019-12-24

- Disabled the manual enrollment orders for audit mode enterprise learners.

9.858 [2.0.33] - 2019-12-23

- Added ability to include or exclude date from the report configuration file name.

9.859 [2.0.32] - 2019-12-17

- Aligned xAPI statement formats with TinCan/Rustici standards
- While uploading bulk users in ‘manager learners’ from django admin, better handling if invalid encoding found.

9.860 [2.0.31] - 2019-12-11

- Added ADR for Multiple User Enterprises.

9.861 [2.0.30] - 2019-12-04

- Get the enterprise_customer linked with SAML and mark it active.

9.862 [2.0.29] - 2019-12-04

- Update the enterprise customer in the session in case of customer with multiple linked enterprises

9.863 [2.0.28] - 2019-12-3

- Added logic to set the EnterpriseCourseEnrollmentSource for the Enterprise Enrollments through offers and management task.

9.864 [2.0.27] - 2019-11-26

- Make the SAML enterprise active at login and de-activate other enterprises learner is linked to.

9.865 [2.0.26] - 2019-11-26

- Updated xapi exports with an active enterprise setting for users with multiple linked enterprises.

9.866 [2.0.25] - 2019-11-22

- Added logic to set the EnterpriseCourseEnrollmentSource for the Enterprise Enrollments background task.

9.867 [2.0.24] - 2019-11-21

- Added logic to set the EnterpriseCourseEnrollmentSource for Enterprise Enrollments by URL.

9.868 [2.0.23] - 2019-11-20

- Display enterprise course enrollments separate from non-enterprise course enrollments in the “Enterprise Customer Learner” Django admin form

9.869 [2.0.22] - 2019-11-18

- Custom get function in EnterpriseCustomerUserManager to enable multiple user enterprises.

9.870 [2.0.21] - 2019-11-14

- Remove success url validation for select enterprise page.

9.871 [2.0.20] - 2019-11-13

- Added Source to Enterprise API Enrollments.

9.872 [2.0.19] - 2019-13-08

- Add manual enrollment audit creation for enrollments created in Manage Learners form.

9.873 [2.0.19] - 2019-11-13

- Sorted results of enterprise-learner API by active flag in descending order so active enterprises are on the top
- [2.0.18] - 2019-11-13
-

- Better handling when Integrated Channels return unexpected results

9.874 [2.0.17] - 2019-11-08

- Added in models to track enterprise enrollment source and updated the Enterprise Course Enrollments and PendingEnrollments to track that source.

9.875 [2.0.16] - 2019-11-07

- Address defect ENT-2463. Add protection within EnterpriseCustomerUser model in enroll method during course enrollments.

9.876 [2.0.15] - 2019-11-07

- Added missing migration for EnterpriseCustomerUser

9.877 [2.0.14] - 2019-11-07

- Add Enterprise selection page to allow a learner to select one of linked enterprises

9.878 [2.0.13] - 2019-11-07

- Add manual order creation to enterprise manual enrollment admin form

9.879 [2.0.12] - 2019-11-06

- Update 'EnterpriseCustomerUser' model. Add 'create_order_for_enrollment'. Called during 'enroll'. Will create an ecommerce order for pending course enrollments.

9.880 [2.0.11] - 2019-11-06

- Add management command to populate sample enterprise data in the LMS within devstack

9.881 [2.0.10] - 2019-10-29

- Add method to Ecommerce API client to call the manual enrollment order API

9.882 [2.0.9] - 2019-10-28

- Updated image url field in content metadata export for cornerstone and degreed

9.883 [2.0.8] - 2019-10-22

- Adding logging to search/all/ endpoint in discovery api client

9.884 [2.0.7] - 2019-10-21

- Added certificate and grades api calls for transmitting learner export to integrated channels

9.885 [2.0.6] - 2019-10-18

- Add query_param to remove expired course runs from /enterprise/api/v1/enterprise_catalogs/UUID/ endpoint

9.886 [2.0.5] - 2019-10-15

- Adding migration file to remove EnterpriseCustomerEntitlement from table schema

9.887 [2.0.4] - 2019-10-10

- Added preview button for EnterpriseCustomerCatalogs in EnterpriseCustomer admin page

9.888 [2.0.3] - 2019-10-09

- Add message box to code management page and admin portal

9.889 [2.0.2] - 2019-10-07

- Updating create_enterprise_course_enrollment task to accept object ids instead of python objects to play nicely with async.
- Also converts course_id to str before handing it to task to play nicely with async.

9.890 [2.0.1] - 2019-10-07

- Commenting out code while troubleshooting signal issue in the LMS

9.891 [2.0.0] - 2019-10-02

- Removing EnterpriseCustomerEntitlement code

9.892 [1.11.0] - 2019-10-02

- Adding post-save receiver to spin off EnterpriseCourseEnrollment creation tasks on CourseEnrollment creation signals

9.893 [1.10.8] - 2019-10-01

- Resolved issue with content_metadata image_url.

9.894 [1.10.7] - 2019-09-25

- Added support to transmit single learner data.

9.895 [1.10.6] - 2019-09-25

- Added ability set supported languages in Cornerstone Global Config.

9.896 [1.10.5] - 2019-09-23

- Updating enterprise_learner_portal LMS API calls to refer to new function locations in the LMS.

9.897 [1.10.4] - 2019-09-05

- Added new endpoint basic_list to EnterpriseEnrollment.

9.898 [1.10.3] - 2019-09-19

- Add enable_portal_reporting_config_screen field to EnterpriseCustomer model.
- Add enable_portal_reporting_config_screen to EnterpriseCustomerSerializer.

9.899 [1.10.2] - 2019-09-18

- Added ability to set password on reporting configuration.

9.900 [1.10.1] - 2019-09-16

- Upgrading requirements.

9.901 [1.10.0] - 2019-09-16

- Add learner portal configuration fields to EnterpriseCustomer model.

9.902 [1.9.12] - 2019-09-06

- Implement “move to completed” functionality for Enterprise Enrollments.

9.903 [1.9.11] - 2019-09-05

- Add new field ‘marked_done’ to EnterpriseCourseEnrollment.

9.904 [1.9.10] - 2019-09-04

- Improved enterprise enrollment workflow logging.

9.905 [1.9.9] - 2019-08-29

- Updated learner portal enrollments endpoint to require an enterprise id.

9.906 [1.9.8] - 2019-08-29

- Corrected missing db migration data for the EnterpriseCustomerReportingConfigurations model

9.907 [1.9.7] - 2019-08-28

- Added API endpoints for EnterpriseCustomerReportingConfigurations and updated permissions to use Feature role based auth.

9.908 [1.9.6] - 2019-08-23

- Added XAPILearnerDataTransmissionAudit model for xapi integrated channel.

9.909 [1.9.5] - 2019-08-21

- Preventing another error in enterprise_learner_portal serializer when certificate info is None.

9.910 [1.9.4] - 2019-08-20

- Adding type check to enterprise_learner_portal serializer.
- Adding enterprise_learner_portal to quality check commands.

9.911 [1.9.3] - 2019-08-20

- Fix for include course run dates and pacing type in the course description sent to SAP. Prior release (1.9.2) did not include bumping the version in __init__.py.

9.912 [1.9.2] - 2019-08-20

- Include course run dates and pacing type in the course description sent to SAP.

9.913 [1.9.1] - 2019-08-19

- Added enterprise_learner_portal to MANIFEST.in file to recursively grab files app on build
- Minor fixes to typos and an image link

9.914 [1.9.0] - 2019-08-12

- Adding enterprise_learner_portal app to support data needs of frontend enterprise learner portal app

9.915 [1.8.9] - 2019-08-15

- Remove tincan from src directory

9.916 [1.8.8] - 2019-08-01

- For CornerstoneCourseListAPI handled corner cases for default values.

9.917 [1.8.7] - 2019-07-31

- Added history models for PendingEnrollment and PendingEnterpriseCustomerUser.
- Sending default values for required fields in Cornerstone Course List API

9.918 [1.8.6] - 2019-07-25

- Add/Update logs for GrantDataSharingPermissions and DataSharingConsentView views to improve monitoring.

9.919 [1.8.5] - 2019-07-25

- Change coupon code request email from address.

9.920 [1.8.4] - 2019-07-24

- Introduce enterprise catalog queries.

9.921 [1.8.3] - 2019-07-24

- Upgrade python requirements.

9.922 [1.8.2] - 2019-07-23

- Log success of coupon code request email send.

9.923 [1.8.1] - 2019-07-22

- Show linked enterprise customer on *Enterprise Customer Learners* and *System wide Enterprise User Role Assignments* admin screen

9.924 [1.8.0] - 2019-07-22

- Replace edx-rbac jwt utils with edx-drf-extensions jwt utils

9.925 [1.7.3] - 2019-07-19

- Change the way we declare dependencies so we can avoid breaking make upgrade in edx-platform.

9.926 [1.7.2] - 2019-07-18

- Added ability to send user's progress to cornerstone

9.927 [1.7.1] - 2019-07-15

- Reverted page size of SAPSF inactive user results from 1000 to 500

9.928 [1.7.0] - 2019-07-15

- Pin certain constraints from edx-platform so that edx-enterprise will install properly there.

9.929 [1.6.23] - 2019-07-15

- Upgrade python requirements

9.930 [1.6.22] - 2019-07-11

- Revert changes made in 1.6.20

9.931 [1.6.21] - 2019-07-11

- Added additional logging for enterprise api

9.932 [1.6.20] - 2019-07-10

- Updated catalog preview URL on enterprise customer catalog admin list display

9.933 [1.6.19] - 2019-07-09

- Added ability to skip keys if their value is None for content exporter

9.934 [1.6.18] - 2019-06-24

- Changed page size of SAPSF inactive user results from 500 to 1000

9.935 [1.6.17] - 2019-06-20

- Fixed Server Error on enterprise course enroll url caused by week_to_complete None value

9.936 [1.6.16] - 2019-06-20

- Capture user attributes sent by cornerstone

9.937 [1.6.15] - 2019-06-18

- Fix error where the search/all/ endpoint in discovery is called with course_key=None

9.938 [1.6.14] - 2019-06-18

- Pass language code instead of language name in languages field of course-list API for cornerstone

9.939 [1.6.13] - 2019-06-17

- Improved logging of *unlink_inactive_sap_learners* command and matching social auth user by *uid* field

9.940 [1.6.12] - 2019-06-14

- Updated discovery clients to always call the enterprise customer site if available

9.941 [1.6.11] - 2019-06-14

- Update the format of course_duration in xAPI payload data.

9.942 [1.6.10] - 2019-06-13

- Remove old catalog model field.

9.943 [1.6.9] - 2019-06-12

- Install django-filter so this app is compatible with newer DRF packages.

9.944 [1.6.8] - 2019-06-11

- Fix error in enrollment flow caused by the way course keys were parsed.

9.945 [1.6.7] - 2019-06-11

- added enable_audit_data_reporting in EnterpriseCustomerSerializer

9.946 [1.6.6] - 2019-06-10

- Use OAuth2AuthenticationAllowInactiveUser as oauth2 authentication instead of BearerAuthentication for course-list API.

9.947 [1.6.5] - 2019-06-06

- Use edx-rbac functions and pin edx-rbac so that we can continue to release edx-enterprise.

9.948 [1.6.4] - 2019-06-05

- Upgrade packages to get latest edx-drf-extensions version.

9.949 [1.6.3] - 2019-06-04

- Remove RBAC waffle switch

9.950 [1.6.2] - 2019-05-31

- Remove old style catalogs

9.951 [1.6.1] - 2019-05-30

- Fallback to request.auth if JWT cookies are not found.

9.952 [1.6.0] - 2019-05-29

- Added new integrated channel *cornerstone* with course-list API.

9.953 [1.5.9] - 2019-05-27

- Reverting changes from 1.5.6.

9.954 [1.5.8] - 2019-05-24

- Bumping version to 1.5.8. 1.5.7 was tagged and released without actually bumping the version

9.955 [1.5.7] - 2019-05-24

- Updating get_paginated_content ent catalog method to use count value given from discovery service

9.956 [1.5.6] - 2019-05-24

- Fix the way a course identifier is found for a given course run.

9.957 [1.5.5] - 2019-05-21

- Clean up rbac authorization related waffle switches and logic

9.958 [1.5.4] - 2019-05-20

- Updating test packages to be inline with edx-platform. Specifically Bleach >2.1.3

9.959 [1.5.3] - 2019-05-16

- Add total number of weeks to view from data consent screen

9.960 [1.5.2] - 2019-05-15

- Remove usages of `get_decoded_jwt_from_request` from `rbac` in favor of `get_decoded_jwt` from `edx-drf-extensions`

9.961 [1.5.1] - 2019-05-09

- Updating consent granted view to redirect to dashboard if consent is not required

9.962 [1.5.0] - 2019-05-08

- Add `sync_learner_profile_data` flag to data returned by `enterprise-learner` endpoint

9.963 [1.4.10] - 2019-05-08

- Add enterprise customer column in the `list_display` admin interface for *SystemWideEnterpriseUserRoleAssignment*
- Update *SystemWideEnterpriseUserRoleAssignment* admin interface search to support search by enterprise customer

9.964 [1.4.9] - 2019-05-02

- Upgrade `edx-rbac` version

9.965 [1.4.8] - 2019-04-26

- Reduce course mode match exception log level

9.966 [1.4.7] - 2019-04-17

- Fix invalid object attribute references in exception message

9.967 [1.4.6] - 2019-04-17

- Stop masking discovery call failures from the client for enterprise catalog endpoint calls.

9.968 [1.4.5] - 2019-04-12

- Revise course mode match exception message in CourseEnrollmentView.

9.969 [1.4.4] - 2019-04-11

- Revise course load exception message in CourseEnrollmentView.

9.970 [1.4.3] - 2019-04-11

- Added *availability* key to default content filter for ECC.

9.971 [1.4.2] - 2019-04-11

- Update *assign_enterprise_user_roles* management command to also assign catalog and enrollment api admin roles.

9.972 [1.4.1] - 2019-04-10

- Update *RouterView* if user is already enrolled in course run of a course then user will land on that *course_run*.

9.973 [1.4.0] - 2019-04-08

- Add new rbac permission checks to enterprise api endpoints.

9.974 [1.3.11] - 2019-04-07

- Update context for *enterprise-openedx-operator* role.

9.975 [1.3.10] - 2019-04-03

- Provide ability to add ECE even if course is closed from manage learners admin interface.

9.976 [1.3.9] - 2019-03-29

- Update role metadata for *edx-openedx-operator* role.
-

9.977 [1.3.8] - 2019-03-29

- Update *assign_enterprise_user_roles* management command to also assign enterprise operator role.

9.978 [1.3.7] - 2019-03-28

- Add data migration for adding edx enterprise operator role.

9.979 [1.3.6] - 2019-03-27

- Introduce rbac models for feature specific roles within edx-enterprise.

9.980 [1.3.5] - 2019-03-22

- Assign an enterprise learner role to new EnterpriseCustomerUser.

9.981 [1.3.4] - 2019-03-21

- Management command to assign enterprise roles to users.

9.982 [1.3.3] - 2019-03-21

- Fixed error in enrollment flow when audit track is selected and no DSC required.

9.983 [1.3.2] - 2019-03-18

- Adding django admin for SystemWideEnterpriseUserRoleAssignments.

9.984 [1.3.1] - 2019-03-13

- Optimizations around unlinking of SAP Success factor inactive users

9.985 [1.3.0] - 2019-03-07

- Introducing Enterprise System Wide Roles and edx-rbac.

9.986 [1.2.12] - 2019-02-15

- Updating enterprise views with new logging
- Updating enterprise views to render new error page in a number of circumstances

9.987 [1.2.11] - 2019-02-07

- Allow admins with enterprise permissions to edit Data Sharing Consent Records

9.988 [1.2.10] - 2019-01-30

- Include Enterprise Catalog UUID in Enterprise Customer django admin inline.

9.989 [1.2.9] - 2019-01-23

- Upgrade requirements, and add code-annotations.
- Add PII annotations to all apps in this repo.
- Enable PII checking during CI.

9.990 [1.2.8] - 2019-01-22

- Revert 1.2.4 to restore DSC functionality.

9.991 [1.2.7] - 2019-01-18

- Replace error level log with info level log when enterprise user is not enrolled in course yet and the *transmit_learner_data* command is run

9.992 [1.2.5] - 2019-01-16

- Updating launch_points data in SapSuccessFactorsContentMetadataExporter so SuccessFactors can be mobile ready

9.993 [1.2.4] - 2019-01-16

- Remove HandleConsentEnrollment view and replaced with a function inside GrantDataSharingPermissions view. Removed GET side effect

9.994 [1.2.3] - 2019-01-10

- Add management command “unlink_inactive_sap_learners” to unlink inactive SAP learners from the related enterprises

9.995 [1.2.2] - 2019-01-09

- Update styling for future courses start date visibility

9.996 [1.2.1] - 2018-12-21

- Handle /search/all/ endpoint large catalog queries to discovery through HTTP POST

9.997 [1.2.0] - 2018-12-19

- Updating the course grade api url called in lms api

9.998 [1.1.4] - 2018-12-19

- Upgrade django-simple-history required version

9.999 [1.1.3] - 2018-12-18

- Add option on EnterpriseCustomer for displaying code management in portal

9.1000 [1.1.2] - 2018-12-12

- Update EnterpriseCustomer model to introduce customer type field

9.1001 [1.1.1] - 2018-12-11

- Use LMS-defined segment track() method

9.1002 [1.1.0] - 2018-12-06

- Updating EnterpriseCustomerReportingConfiguration model. ManyToMany relationship with EnterpriseCustomerCatalog
- Updating EnterpriseCustomerReportingConfigurationAdminForm validation
- Updating EnterpriseCustomerReportingConfigurationSerializer

9.1003 [1.0.6] - 2018-11-28

- Added username and user email in EnterpriseCustomerUserAdmin list display.
- Added search by username and user email in EnterpriseCustomerUserAdmin.

9.1004 [1.0.5] - 2018-11-14

- Added enterprise api for requesting additional coupon codes.

9.1005 [1.0.4] - 2018-11-07

- Make HTTP POST request to get catalog results from discovery.

9.1006 [1.0.3] - 2018-11-02

- Fix translations for enterprise pages.

9.1007 [1.0.2] - 2018-10-25

- Updated EnterpriseCustomerReportingConfiguration model with PGP key

9.1008 [1.0.1] - 2018-10-24

- Made autocohorting API availability based on a configuration option.

9.1009 [1.0.0] - 2018-10-16

- Upgrade edx-drf-extensions with refactored imports.
- Remove Hawthorn testing for upcoming backward incompatible change.

9.1010 [0.73.6] - 2018-10-04

- SuccessFactors: Submit batch/chunk of OCN items to tenants until error status

9.1011 [0.73.5] - 2018-09-21

- Added ability to query enterprises by slug on the with_access_to endpoint

9.1012 [0.73.4] - 2018-09-17

- Added ability to assign cohort upon enrollment.
- Added ability to unenroll in enrollment API.

9.1013 [0.73.3] - 2018-09-14

- Added Country field to the EnterpriseCustomer model.

9.1014 [0.73.2] - 2018-09-11

- Fixed 500 error on enterprise customer admin screen.

9.1015 [0.73.1] - 2018-08-30

- Remove the SailThru flags for enterprise learner when un-linking it from enterprise.

9.1016 [0.73.0] - 2018-08-21

- Changed permission logic and added filtering options for the enterprise with_access_to endpoint.

9.1017 [0.72.7] - 2018-08-20

- Added preview field that takes user to Discovery with elastic search results for the catalog

9.1018 [0.72.6] - 2018-08-17

- Added management command to send course enrollment and course completion info for enterprise customers.

9.1019 [0.72.5] - 2018-08-09

- Revise management command query to include all potentially-applicable enrollment records

9.1020 [0.72.4] - 2018-08-08

- Move some fields from Global Degreed Configuration to Enterprise Degreed Configuration.

9.1021 [0.72.3] - 2018-08-08

- Added LearnerInfoSerializer and CourseInfoSerializer for serializing xAPI payload data.

9.1022 [0.72.2] - 2018-07-27

- Added endpoint to check a user's authorization to Enterprises based on membership in a given django group.

9.1023 [0.72.1] - 2018-07-26

- Added missing migrations for xAPI LRS Configuration model

9.1024 [0.72.0] - 2018-07-24

- Implemented reporting channel of course completion via X-API

9.1025 [0.71.2] - 2018-07-23

- Add thumbnail images in exported metadata content by content type.

9.1026 [0.71.1] - 2018-07-23

- Updated message for invalid Enterprise Customer Catalog references in B2B enrollment workflow.

9.1027 [0.71.0] - 2018-07-20

- Updated TinCanPython package to support python 3
- Updated UUID field to nowrap in admin interface of enterprise customer catalog model.

9.1028 [0.70.8] - 2018-07-13

- Display customer catalog content filter's default value on enterprise customer admin.

9.1029 [0.70.7] - 2018-07-12

- Make customer catalog content filter's default value configurable.

9.1030 [0.70.6] - 2018-07-09

- Pass catalog value only when provided on enterprise course enrollment page.

9.1031 [0.70.5] - 2018-07-06

- Send learner data transmissions to integrated channels by course key and course run id.

9.1032 [0.70.4] - 2018-07-03

- Use query param “catalog” instead of “enterprise_customer_catalog_uuid” for catalog based enterprise discounts.

9.1033 [0.70.3] - 2018-06-29

- Apply enterprise catalog conditional offer by the provided enterprise catalog UUID.

9.1034 [0.70.2] - 2018-06-28

- Modify enterprise branding config API to use enterprise slug as the lookup_field.

9.1035 [0.70.1] - 2018-06-27

- Paginate linked learners list on manage learners Django admin view.

9.1036 [0.70.0] - 2018-06-26

- Add unique slug field to EnterpriseCustomer.

9.1037 [0.69.6] - 2018-06-25

- Update requirements to fix pip install issues and to keep in line with edx-platform.

9.1038 [0.69.5] - 2018-06-25

- Fix the Direct-to-Audit enrollment issue in case of course instead of course run.

9.1039 [0.69.4] - 2018-06-20

- Strip locale values.

9.1040 [0.69.3] - 2018-06-20

- Add and transmit customer specific locales so that SuccessFactors show course title and description.

9.1041 [0.69.2] - 2018-06-18

- Fix the Direct-to-Audit enrollment issue in case of course.

9.1042 [0.69.1] - 2018-06-07

- 500 error when attempting to enroll using course-level URL.

9.1043 [0.69.0] - 2018-05-31

- Add a *progress_v2* option in the reporting config to be used for data API fetching.

9.1044 [0.68.9] - 2018-05-31

- Increased character limit from 20 to 255 for field title in EnterpriseCustomerCatalog model
- Reorder list display for EnterpriseCustomerCatalogAdmin
- Add sorting order for EnterpriseCustomerCatalogAdmin

9.1045 [0.68.8] - 2018-05-30

- Mark ECU as inactive internally if SAPSF says the ECU is inactive on their side.

9.1046 [0.68.7] - 2018-05-24

- Admin tooling enterprise customer reporting configuration enhancement - Order by Enterprise Customer Name.

9.1047 [0.68.6] - 2018-05-22

- Update DSC to show notification interstitial communicating to enterprise learner they are leaving company's site.

9.1048 [0.68.5] - 2018-05-17

- Configuration to show/hide original price on enterprise course landing page.

9.1049 [0.68.4] - 2018-05-16

- Remove constraints on the reporting config.

9.1050 [0.68.3] - 2018-05-11

- Update enrollment api authorization to check group permissions.

9.1051 [0.68.2] - 2018-05-10

- Dropped sap_success_factors_historicalsapsuccessfactorsenterprise80ad table.

9.1052 [0.68.1] - 2018-05-09

- Add *json* report type.

9.1053 [0.68.0] - 2018-05-09

- Allow reporting configs to work for arbitrary data and report types.

9.1054 [0.67.8] - 2018-05-04

- Added ordering to resolve warnings of probable invalid pagination data.

9.1055 [0.67.7] - 2018-04-23

- Update the messages when an enterprise learner leave an organization.

9.1056 [0.67.6] - 2018-04-20

- Update user session when they become an Enterprise learner.

9.1057 [0.67.5] - 2018-04-18

- Added ability to specify data sharing consent wording on a per enterprise basis.

9.1058 [0.67.4] - 2018-04-12

- Add configuration to allow replacing potentially sensitive SSO usernames.

9.1059 [0.67.3] - 2018-04-05

- Improved integrated channel logging.

9.1060 [0.67.2] - 2018-04-05

- Fix the enterprise manage learner django admin tool is loading correctly for chrome users.

9.1061 [0.67.1] - 2018-04-04

- Integrated channel refactoring cleanup.

9.1062 [0.67.0] - 2018-03-26

- Refactored integrated channel code to allow for greater flexibility when transmitting content metadata.

9.1063 [0.66.2] - 2018-03-26

- Update isort version and sort imports after making consent and integrated_channels first party apps.

9.1064 [0.66.1] - 2018-03-23

- Temporarily disable linked learners list on manage learners Django admin view until paging can be added.

9.1065 [0.66.0] - 2018-03-05

- Add EnterpriseCustomerCatalog course detail endpoint.

9.1066 [0.65.8] - 2018-02-23

- Add “Enrollment Closed” in course title if the course is no longer open for enrollment.

9.1067 [0.65.7] - 2018-02-14

- Support multiple emails in EnterpriseCustomerReportingConfiguration.
- Only require email(s) in EnterpriseCustomerReportingConfiguration if the selected delivery method is email.

9.1068 [0.65.6] - 2018-02-13

- Remove the renderer.py file.

9.1069 [0.65.5] - 2018-02-13

- Add functionality in enterprise django admin for transmitting courses metadata related to a specific enterprise.

9.1070 [0.65.4] - 2018-02-09

- Indicate when a course is no longer open for enrollment by updating course title for transmit courses metadata.

9.1071 [0.65.3] - 2018-02-06

- Decreased SuccessFactors course metadata chunk size from 1000 to 500, per SAP's recommendation.

9.1072 [0.65.2] - 2018-02-05

- Updated the “Data Sharing Policy” language.

9.1073 [0.65.1] - 2018-02-02

- Provide an option for enterprise to pull enterprise catalog API in XML format not just JSON.

9.1074 [0.65.0] - 2018-01-30

- Add migration for removing old password fields from the database.

9.1075 [0.64.0] - 2018-01-29

- Removed code references to old password fields.

9.1076 [0.63.0] - 2018-01-25

- Improved handling of password fields on database models.

9.1077 [0.62.0] - 2018-01-18

- Exclude credit course mode option from course enrollment page.

9.1078 [0.61.6] - 2018-01-18

- Group Name, Active, Site, and Logo together.
- Rename “Provider id” form label to “Identity Provider”
- Rename “Entitlement id” form label to “Seat Entitlement”
- Rename “Coupon URL” form label to “Seat Entitlement URL”
- Add a “View details” hyperlink next to identity provider drop-down.
- Add a “Create a new catalog” link under the Catalog drop-down.
- Add a “View details” hyperlink next to catalog field, if catalog is selected.
- Add a “Create a new identity provider” link under the Identity Provider drop-down.

9.1079 [0.61.5] - 2018-01-18

- Include start date in all course runs title when pushing to Integrated Channels.

9.1080 [0.61.4] - 2018-01-12

- Add localized currency to enterprise landing page.

9.1081 [0.61.3] - 2018-01-11

- Fix enterprise logo stretching issue in enterprise sidebar on course/program enrollment pages.

9.1082 [0.61.2] - 2018-01-09

- Add missing migrations for sap_success_factors and degreed.

9.1083 [0.61.1] - 2018-01-09

- Update django admin list view for enterprise customer model.

9.1084 [0.61.0] - 2018-01-09

- SuccessFactors Admin Update: Enterprise Customer Configuration.

9.1085 [0.60.0] - 2018-01-03

- Add sftp configuration options for EnterpriseCustomerReportingConfiguration.

9.1086 [0.59.0] - 2017-12-28

- Add check for active companies when getting list of channels

9.1087 [0.58.0] - 2017-12-22

- Add save_enterprise_customer_users command.

9.1088 [0.57.0] - 2017-12-21

- Remove references to SSO IdP config drop_existing_session flag.

9.1089 [0.56.5] - 2017-12-20

- Fix templates to use new bootstrap bundle library.

9.1090 [0.56.4] - 2017-12-19

- Fix syntax error in template-embedded Javascript.

9.1091 [0.56.3] - 2017-12-14

- Make sure root url has a fallback for proxy enrollment email links.

9.1092 [0.56.2] - 2017-12-13

- Add course_enrollments API endpoint to swagger specification.

9.1093 [0.56.1] - 2017-12-13

- Add publish_audit_enrollment_url flag to EnterpriseCustomerCatalog.

9.1094 [0.56.0] - 2017-12-13

- Update create_enterprise_course_enrollment command.

9.1095 [0.55.7] - 2017-12-13

- Ensure that proxy enrollment email links trigger SSO.

9.1096 [0.55.6] - 2017-12-12

- Check site configuration for from email address first

9.1097 [0.55.5] - 2017-12-11

- Added course start date to title string for instructor-led courses

9.1098 [0.55.4] - 2017-12-06

- Redirect to embargo restriction message page if user is blocked from accessing course.

9.1099 [0.55.3] - 2017-12-05

- Add integrated channel configuration info to course metadata push task logging.

9.1100 [0.55.2] - 2017-12-04

- Include additional context for learner data transmission job exceptions.

9.1101 [0.55.1] - 2017-11-30

- Track enterprise course enrollment events.

9.1102 [0.55.0] - 2017-11-29

- Add Degreed as new integrated channel.

9.1103 [0.54.1] - 2017-11-29

- Increase font size on data sharing consent page.

9.1104 [0.54.0] - 2017-11-28

- Introduce the bulk enrollment/upgrade api endpoint for Enterprise Customers.

9.1105 [0.53.19] - 2017-11-28

- Do not change EnterpriseCustomerReportingConfiguration.password on update.

9.1106 [0.53.18] - 2017-11-28

- Add Identity Provider's ID to enterprise customer API response.

9.1107 [0.53.17] - 2017-11-27

- Remove inaccurate landing page audit track language.

9.1108 [0.53.16] - 2017-11-22

- Use `LMS_INTERNAL_ROOT_URL` instead of `LMS_ROOT_URL` for API base.

9.1109 [0.53.15] - 2017-11-16

- Use the cryptography package instead of the unmaintained pycrypto.

9.1110 [0.53.14] - 2017-11-14

- Link learner to enterprise customer directly using “tpa_hint” URL parameter.

9.1111 [0.53.13] - 2017-11-14

- Update DSC policy to match legal requirements.

9.1112 [0.53.12] - 2017-11-09

- Remove “Discount provided by...” text on the program landing page.

9.1113 [0.53.11] - 2017-11-06

- Removing `SAP_USE_ENTERPRISE_ENROLLMENT_PAGE` switch via django waffle and use landing page URL instead of track selection page.

9.1114 [0.53.10] - 2017-11-02

- Move data sharing policy to its own partial to improve theming of the data sharing consent page

9.1115 [0.53.9] - 2017-11-02

- Apply appropriate content filtering to the EnterpriseCustomerCatalog detail endpoints.

9.1116 [0.53.8] - 2017-11-02

- Show generic info message on enterprise course enrollment page.

9.1117 [0.53.7] - 2017-10-30

- Added inline admin form to EnterpriseCustomer admin for EnterpriseCustomerCatalog.

9.1118 [0.53.6] - 2017-10-30

- Fix error for empty course start date on DSC page.

9.1119 [0.53.5] - 2017-10-26

- Fetch catalog courses in large chunks to avoid API limit.

9.1120 [0.53.4] - 2017-10-26

- Preserve catalog querystring on declining DSC.

9.1121 [0.53.3] - 2017-10-26

- Fixing logo size on themed enterprise pages

9.1122 [0.53.2] - 2017-10-24

- Remove unused dependency on django-extensions

9.1123 [0.53.1] - 2017-10-24

- Fix alteration in querystring parameters for decorator “enterprise_login_required”.

9.1124 [0.53.0] - 2017-10-24

- Get rid of the *EnterpriseIntegratedChannel* model and any other related but unused code.

9.1125 [0.52.10] - 2017-10-23

- Fix migration issue for *enabled-course-modes* field of EnterpriseCustomerCatalog

9.1126 [0.52.9] - 2017-10-20

- Update the call level to enrollment uls from EnterpriseCustomer to EnterpriseCustomerCatalog.

9.1127 [0.52.8] - 2017-10-20

- Update EnterpriseApiClient.get_enterprise_courses to account for EnterpriseCustomerCatalogs.

9.1128 [0.52.7] - 2017-10-20

- Update course enrollment view for enterprise enabled course modes.

9.1129 [0.52.6] - 2017-10-19

- Update the EnterpriseCustomerCatalog migration.

9.1130 [0.52.5] - 2017-10-19

- Add EnterpriseCustomerCatalog UUID as query parameter “catalog” in enterprise course and program enrollment URL’s.

9.1131 [0.52.4] - 2017-10-18

- Upgrade django-simple-history to 1.9.0. Add needed migrations.

9.1132 [0.52.3] - 2017-10-18

- Introducing EnterpriseCustomerReportingConfig model for enterprise_reporting.

9.1133 [0.52.2] - 2017-10-18

- If a course is unenrollable, the program and course enrollment landing pages will display only a subset of information.

9.1134 [0.52.1] - 2017-10-15

- Change a log level from *error* to *info* in our LMS API Client, as it wasn't really an error.

9.1135 [0.52.0] - 2017-10-14

- Implement a direct-audit-enrollment pathway for course enrollment.
- Implement a RouterView that the enrollment URLs have to go through before redirection to a downstream view.

9.1136 [0.51.5] - 2017-10-11

- Added enabled_course_modes JSONField to EnterpriseCustomerCatalog model

9.1137 [0.51.4] - 2017-10-11

- Added UTM parameters to marketing, track selection, and course/program enrollment URLs returned by Enterprise API.

9.1138 [0.51.3] - 2017-10-10

- Fix bug related to EnterpriseCustomer creation form introduced with 0.51.0.

9.1139 [0.51.2] - 2017-10-10

- Modify EnterpriseCustomer.catalog_contains_course to check EnterpriseCustomerCatalogs.

9.1140 [0.51.1] - 2017-10-06

- Refactor user-facing DSC view's logic.

9.1141 [0.51.0] - 2017-10-05

- Make discovery-service lookups site-aware

9.1142 [0.50.1] - 2017-10-03

- Improved robustness for *force_fresh_session* decorator in conjunction with *enterprise_login_required*
- Consciously avoid attempting to sync back details for SAPSF users who aren't linked via SSO

9.1143 [0.50.0] - 2017-10-03

- Add contains_content_items endpoint to EnterpriseCustomerViewSet and EnterpriseCustomerCatalogViewSet.

9.1144 [0.49.0] - 2017-10-02

- Rewrite all of our CSS in SASS/SCSS.
- Use Bootstrap for our modals.
- Fix existing course modal UI issues using Bootstrap & SASS/SCSS.

9.1145 [0.48.2] - 2017-09-29

- Step 2 in making enrollment email template linked to enterprise. Remove site from model. No migration.

9.1146 [0.48.1] - 2017-09-25

- Step 1 in making enrollment email template linked to enterprise. Make 'site' nullable, add 'enterprise_customer'.

9.1147 [0.48.0] - 2017-09-25

- Add extra details to the program enrollment landing page.

9.1148 [0.47.1] - 2017-09-25

- Add proper permissions/filtering schemes for all of our endpoints.

9.1149 [0.47.0] - 2017-09-21

- Step 3 in safe deployment of removing old consent models: make migrations to delete the outstanding fields/models.

9.1150 [0.46.8] - 2017-09-21

- Step 2 in safe deployment of removing old consent models: remove *require_account_level_consent*, but no migration.

9.1151 [0.46.7] - 2017-09-21

- Step 1 in safe deployment of removing old consent models: make *require_account_level_consent* nullable.

9.1152 [0.46.6] - 2017-09-21

- Added some log messages to trace possible 404 issue.

9.1153 [0.46.5] - 2017-09-21

- Remove old account-level consent features as well as consent from `EnterpriseCourseEnrollment`.

9.1154 [0.46.4] - 2017-09-20

- Abstract away usage of *configuration_helpers*.

9.1155 [0.46.3] - 2017-09-19

- Make bulk enrollment emails more intelligent

9.1156 [0.46.2] - 2017-09-19

- Add exception handling for transmit course metadata task.

9.1157 [0.46.1] - 2017-09-18

- Remove the *auth-user* endpoint completely.

9.1158 [0.46.0] - 2017-09-15

- Allow multi-course enrollment for enterprise users in admin.

9.1159 [0.45.0] - 2017-09-14

- Modified enterprise-learner API endpoint to include the new DataSharingConsent model data.

9.1160 [0.44.0] - 2017-09-08

- Added MVP version of the Programs Enrollment Landing Page.

9.1161 [0.43.5] - 2017-09-08

- Wrapped API error handling into the clients themselves.

9.1162 [0.43.4] - 2017-09-07

- Removed the text if there is no discount on the course enrollment landing page.

9.1163 [0.43.3] - 2017-09-06

- Ensure that segment is loaded and firing page events for all user facing enterprise views.

9.1164 [0.43.2] - 2017-09-06

- Display the enterprise discounted text on the course enrollment landing page.

9.1165 [0.43.1] - 2017-09-05

- Remove support for writing `consent_granted` in `enterprise-course-enrollment` api.

9.1166 [0.43.0] - 2017-08-31

- Add architecture for program-scoped data sharing consent.

9.1167 [0.42.0] - 2017-08-24

- Do not create baskets and orders for audit enrollments.

9.1168 [0.41.0] - 2017-08-24

- Migrate the codebase to the new `consent.models.DataSharingConsent` model for when dealing with consent.

9.1169 [0.40.7] - 2017-08-23

- Fix bug causing 500 error on course enrollment page when the course does not have a course image configured.

9.1170 [0.40.6] - 2017-08-23

- Update Consent API to use Discovery worker user for auth, rather than request user.

9.1171 [0.40.5] - 2017-08-23

- Update SAP course export to use enterprise courses API.

9.1172 [0.40.4] - 2017-08-23

- Fix 500 server error on enterprise course enrollment page.

9.1173 [0.40.3] - 2017-08-21

- Change landing page course modal to use discovery api for populating course details.

9.1174 [0.40.2] - 2017-08-16

- Increase capability and compatibility of Consent API.

9.1175 [0.40.1] - 2017-08-11

- Add new unified DataSharingConsent model to the *consent* app.

9.1176 [0.40.0] - 2017-08-08

- Add Enterprise API Gateway for new Enterprise Catalogs and Programs endpoints.
- Add `/enterprise/api/v1/enterprise-catalogs/` endpoint.
- Add `/enterprise/api/v1/enterprise-catalogs/{uuid}/` endpoint.
- Add `/enterprise/api/v1/programs/{uuid}/` endpoint.

9.1177 [0.39.9] - 2017-08-08

- Added management command “`create_enterprise_course_enrollments`” for missing enterprise course enrollments.

9.1178 [0.39.8] - 2017-08-04

- Fixed session reset decorator bug.

9.1179 [0.39.7] - 2017-08-04

- Make whether Enterprise Customers get data for audit track enrollments configurable.

9.1180 [0.39.6] - 2017-08-02

- Fixed the text cutoff in the bottom of the course info overlay.

9.1181 [0.39.5] - 2017-08-02

- Only send one completion status per enrollment for SAP SuccessFactors.

9.1182 [0.39.4] - 2017-08-01

- Create Audit enrollment in E-Commerce system when user enrolls in the audit mode in enterprise landing page.

9.1183 [0.39.3] - 2017-07-28

- Remove Macro use from swagger api config as it is not supported by AWS.

9.1184 [0.39.2] - 2017-07-27

- Introduce new endpoint to the Enterprise API to query for courses by enterprise id.

9.1185 [0.39.1] - 2017-07-27

- Ensure catalog courses API endpoint users are associated with an EnterpriseCustomer.

9.1186 [0.39.0] - 2017-07-24

- Officially include Consent application by ensuring it is installable.

9.1187 [0.38.7] - 2017-07-22

- Add a new Consent application.
- Add initial implementation of a generic Consent API.

9.1188 [0.38.6] - 2017-07-21

- Remove SSO-related consent capabilities

9.1189 [0.38.5] - 2017-07-19

- Add page_size in querystring and data mapping template to fix “next” and “previous” urls in API response.

9.1190 [0.38.4] - 2017-07-18

- Fix DSC Policy Language Needs

9.1191 [0.38.3] - 2017-07-14

- Fix dependency installation process in setup.py.

9.1192 [0.38.2] - 2017-07-14

- Add consent declined message to course enrollment landing page.

9.1193 [0.38.1] - 2017-07-13

- Remove requirement on too-new django-simple-history version
- Require slightly older django-config-models version

9.1194 [0.38.0] - 2017-07-11

- Move to edx-platform release-focused testing
- Add Django 1.11 support in Hawthorn testing branch

9.1195 [0.37.1] - 2017-07-11

- Update Enterprise landing page styling/language

9.1196 [0.37.0] - 2017-07-06

- Update enterprise catalog api endpoint so that api returns paginated catalogs.

9.1197 [0.36.11] - 2017-06-29

- Update DSC page language.

9.1198 [0.36.10] - 2017-06-29

- Introducing SAP_USE_ENTERPRISE_ENROLLMENT_PAGE switch via django waffle.

9.1199 [0.36.9] - 2017-06-28

- Refactor of automatic session termination logic.

9.1200 [0.36.8] - 2017-06-28

- Enforce data sharing consent at login for SSO users only if data sharing consent is requested at login.

9.1201 [0.36.7] - 2017-06-25

- UI tweaks to the enterprise landing page and course overview modal.

9.1202 [0.36.6] - 2017-06-25

- Disable atomic transactions for CourseEnrollmentView to ensure that new EnterpriseCustomerUser records are saved to the database in time for ecommerce API calls.

9.1203 [0.36.5] - 2017-06-23

- Apply automatic session termination logic to enterprise landing page based on enterprise customer configuration.

9.1204 [0.36.4] - 2017-06-21

- Sort course modes in landing page.

9.1205 [0.36.3] - 2017-06-21

- Fix for being unable to create course catalog clients due to upstream removal of the library.

9.1206 [0.36.2] - 2017-06-21

- Add the ability to pass limit, offset and page_size parameters to enterprise catalog courses.

9.1207 [0.36.1] - 2017-06-20

- Properly bump PyPI to latest changes from v0.36.0.

9.1208 [0.36.0] - 2017-06-20

- Migrate from old, monolithic python-social-auth to latest, split version.
- Rework the NotConnectedToOpenEdX exception to be just one, and to say which method/dependency is missing.

9.1209 [0.35.2] - 2017-06-20

- Fix Next and Previous page urls for enterprise catalog courses.

9.1210 [0.35.1] - 2017-06-15

- Displayed course run price with entitlement on landing page and course information overlay

9.1211 [0.35.0] - 2017-06-15

- Allow account-level data sharing consent in a course-specific context

9.1212 [0.34.7] - 2017-06-14

- Enable “Continue” button flows on enterprise landing page

9.1213 [0.34.6] - 2017-06-14

- Fixed layout of data sharing consent decline modal on mobile view

9.1214 [0.34.5] - 2017-06-09

- Add Django 1.10 support back

9.1215 [0.34.4] - 2017-06-09

- Added course information overlay

9.1216 [0.34.3] - 2017-06-07

- Make enterprise landing page url available in the enterprise api and SAP course export.

9.1217 [0.34.2] - 2017-06-06

- Fix UI issues (unexpected html escape) on enterprise landing page.

9.1218 [0.34.1] - 2017-06-06

- Bug fix for Data sharing consent pop up page.

9.1219 [0.34.0] - 2017-06-05

- Update data backing and behavior of enterprise landing page
- Fix template prioritization bug
- Fix URL rendering in enterprise login decorator

9.1220 [0.33.24] - 2017-06-02

- UI updates for data sharing consent page.

9.1221 [0.33.23] - 2017-06-02

- Fix a bug with unexpected image data in SAP course export job.

9.1222 [0.33.22] - 2017-06-02

- Add an *EnterpriseApiClient* method for getting enrollment data about a single user+course pair
- Add logic to enterprise landing page that redirects users to the course when already registered

9.1223 [0.33.21] - 2017-06-01

- UI updates for course mode selection in enterprise landing page.

9.1224 [0.33.20] - 2017-05-23

- Migrate from mako templates to django templates

9.1225 [0.33.19] - 2017-05-18

- Display account created/linked messages on enterprise landing page

9.1226 [0.33.18] - 2017-05-17

- Add Enable audit enrollment flag

9.1227 [0.33.17] - 2017-05-16

- Add django admin for enterprise course enrollment models

9.1228 [0.33.16] - 2017-05-15

- Bug fixes for SAP learner completion data passback.

9.1229 [0.33.15] - 2017-05-10

- Additional minor UI updates for enterprise landing page.

9.1230 [0.33.14] - 2017-05-10

- Add new externally managed consent option for enterprise customers.

9.1231 [0.33.13] - 2017-05-09

- Fix invalid API Gateway URIs

9.1232 [0.33.12] - 2017-05-03

- Add enterprise landing page

9.1233 [0.33.11] - 2017-05-02

- Add tpa hint if available for launchURLs for SAP Course metadata push.

9.1234 [0.33.10] - 2017-05-02

- Fix bug with inactivating SAP courses that are no longer in the catalog.

9.1235 [0.33.9] - 2017-04-26

- Fix enterprise logo validation message for max image size limit

9.1236 [0.33.8] - 2017-04-26

- Updated calls to `get_edx_api_data` as its signature has changed in `openedx`.

9.1237 [0.33.7] - 2017-04-24

- Redirect to login instead of raising `Http404` if `EnterpriseCustomer` missing.
- Add `confirmation_alert_prompt_warning` to context of account-level consent view.

9.1238 [0.33.6] - 2017-04-21

- Increase max size limit for enterprise logo

9.1239 [0.33.5] - 2017-04-20

- Added vertical hanging indent mode to isort settings and adjusted current imports

9.1240 [0.33.4] - 2017-04-18

- Enforce login for course-specific data sharing consent views.

9.1241 [0.33.3] - 2017-04-18

- Fixed the CSS for the expand arrow in the data sharing consent page.

9.1242 [0.33.2] - 2017-04-17

- Update Data Sharing Consent message.

9.1243 [0.33.1] - 2017-04-17

- Order enterprise customers by name on enterprise customer django admin

9.1244 [0.33.0] - 2017-04-11

- Improve accounting for inactive courses for SAP course export.

9.1245 [0.32.1] - 2017-04-06

- Bug Fix: Added Handling for user enrollment to courses that do not have a start date.

9.1246 [0.32.0] - 2017-04-06

- Refine SAP course export parameters

9.1247 [0.31.4] - 2017-04-05

- Added missing migration file for recent string updates

9.1248 [0.31.3] - 2017-04-04

- Modified SAP completion status data to correctly indicate a failing grade to SAP systems.

9.1249 [0.31.2] - 2017-04-03

- Bugfix: Resolve IntegrityError getting raised while linking existing enterprise users when data sharing consent is disabled for the related enterprise.

9.1250 [0.31.1] - 2017-03-31

- Bugfix: Allow unlinking of enterprise learners with plus sign or certain other characters in email address.

9.1251 [0.31.0] - 2017-03-30

- Edited UI and error strings.

9.1252 [0.30.0] - 2017-03-27

- Fully implements `sap_success_factors` transmitters and client to communicate with the SAP SuccessFactors API, and to handle auditing and other business logic for both catalog and learner data calls.

9.1253 [0.29.1] - 2017-03-27

- Support for `segment.io` events on data sharing consent flow

9.1254 [0.29.0] - 2017-03-23

- Updates `integrated_channels` management command *transmit_learner_data* to support sending completion data for self-paced courses, and to use the Certificates API for instructor-paced courses.

9.1255 [0.28.0] - 2017-03-23

- New data sharing consent view supporting `failure_url` parameter

9.1256 [0.27.6] - 2017-03-21

- Removed `OAuth2Authentication` class from API viewset definitions

9.1257 [0.27.5] - 2017-03-17

- Updated `api.yaml` to resolve swagger configuration issues.

9.1258 [0.27.4] - 2017-03-17

- Allows enterprise enrollments to be made on servers that sit behind a load balancer.

9.1259 [0.27.3] - 2017-03-16

- Added `integrated_channels` management command to transmit courseware metadata to SAP SuccessFactors.

9.1260 [0.27.2] - 2017-03-10

- Added `integrated_channels` management command to transmit learner completion data to SAP SuccessFactors.

9.1261 [0.27.1] - 2017-03-13

- Added `api.yaml` and `api-compact.yaml` files to introduce api endpoints for catalog api-manager.

9.1262 [0.27.0] - 2017-03-02

- Added API endpoint for fetching catalogs and catalog courses.

9.1263 [0.26.3] - 2017-03-02

- Added `integrated_channels` to `MANIFEST.in` to properly include migrations for the new packages.

9.1264 [0.26.2] - 2017-03-02

- Fixed package listing in `setup.py` to avoid import errors when using as a library

9.1265 [0.26.1] - 2017-02-28

- Added support for retrieving access token from SAP SuccessFactors
- Added indicator in Sap SuccessFactors admin tool for checking the configuration's access to SuccessFactors.

9.1266 [0.26.0] - 2017-02-28

- Formally introducing new integrated_channels apps
- Adding new models and admin interfaces for integrated_channel and sap_success_factors

9.1267 [0.25.0] - 2017-02-28

- Refactor _enroll_users() method to pay down technical debt
- Improve admin messaging around enrollment actions

9.1268 [0.24.0] - 2017-02-27

- API for SSO pipeline is simplified to a single element.
- SSO users are linked to relevant Enterprise Customer when data sharing consent is disabled.

9.1269 [0.23.2] - 2017-02-22

- SSO users are not created as EnterpriseCustomerUsers until all consent requirements have been fulfilled.

9.1270 [0.22.1] - 2017-02-20

- Course Catalog API degrades gracefully in absence of Course Catalog service.

9.1271 [0.22.0] - 2017-02-14

- Added API endpoint for fetching entitlements available to an enterprise learner

9.1272 [0.21.2] - 2017-02-07

- Add id in EnterpriseCustomerUserSerializer fields

9.1273 [0.21.0] - 2017-01-30

- Add UI handling for course-specific data sharing consent

9.1274 [0.20.0] - 2017-01-30

- Add ability to select existing learners to be enrolled in courses from admin

9.1275 [0.19.1] - 2017-01-30

- Resolved conflicting urls for User API endpoint.

9.1276 [0.19.0] - 2017-01-30

- Added read-only enterprise API endpoint for IDAs.
- Moved utility functions from api.py to utils.py

9.1277 [0.18.0] - 2017-01-27

- Add the ability to notify manually-enrolled learners via email.

9.1278 [0.17.0] - 2017-01-25

- Add the EnterpriseCourseEnrollment model and related methods

9.1279 [0.16.0] - 2017-01-25

- Fix a bug preventing a course catalog from being unlinked from an EnterpriseCustomer

9.1280 [0.15.0] - 2017-01-25

- Enroll users in a program.

9.1281 [0.14.0] - 2017-01-20

- Added view of seat entitlements on enterprise admin screen

9.1282 [0.13.0] - 2017-01-06

- Dynamically fetch available course modes in the Manage learners admin

9.1283 [0.12.0] - 2017-01-05

- Create pending enrollment for users who don't yet have an account.

9.1284 [0.11.0] - 2017-01-05

- Added links from the Manage Learners admin panel to individual learners.

9.1285 [0.10.0] - 2017-01-04

- Added the ability to search the Manage Learners admin panel by username and email address.

9.1286 [0.9.0] - 2016-12-29

- In django admin page for enterprise customer added alphabetical ordering for catalog drop down and displayed catalog details link next to selected catalog.

9.1287 [0.8.0] - 2016-12-08

- added the branding information api methods to return the enterprise customer logo on the basis of provider_id or uuid.
- Updated the logo image validator to take an image of size maximum of 4kb.

9.1288 [0.7.0] - 2016-12-07

- Added a feature to enroll users in a course while linking them to an enterprise customer.

9.1289 [0.6.0] - 2016-12-04

- Fixed EnterpriseCustomer form to make Catalog field optional
- Added user bulk linking option
- Added Data Sharing Consent feature

9.1290 [0.5.0] - 2016-11-28

- Added checks to make sure enterprise customer and identity provider has one-to-one relation.
- Added a helper method to retrieve enterprise customer branding information

9.1291 [0.4.1] - 2016-11-24

- Fixed User.post_save handler causing initial migrations to fail

9.1292 [0.4.0] - 2016-11-21

- Set up logic to call course catalog API to retrieve catalog listing to attach to EnterpriseCustomer.

9.1293 [0.3.1] - 2016-11-21

- Fixed missing migration.

9.1294 [0.3.0] - 2016-11-16

9.1294.1 Added

- Added Pending Enterprise Customer User model - keeps track of user email linked to Enterprise Customer, but not yet used by any user.
- Added custom “Manage Learners” admin view.

9.1294.2 Technical features

- Added sphinx-napoleon plugin to support rendering Google Style docstrings into documentation properly (i.e. make it recognize function arguments, returns etc.)
- Added translation files

9.1295 [0.2.0] - 2016-11-15

- Linked EnterpriseCustomer model to Identity Provider model

9.1296 [0.1.2] - 2016-11-04

- Linked EnterpriseCustomer model to django Site model

9.1297 [0.1.1] - 2016-11-03

- Enterprise Customer Branding Model and Django admin integration

9.1298 [0.1.0] - 2016-10-13

- First release on PyPI.
- Models and Django admin integration

MULTIPLE ENTERPRISE SUPPORT

10.1 Status

Accepted

10.2 Context

There is a business need to be able to support a learner being linked to multiple enterprises. Today, edX workflows only support a 1-to-1 linkage and our system does not prevent linking multiple Enterprises to a single learner. Hence not all workflows work as expected.

Some examples of these errant workflows are:

1. When a learner is linked with an enterprise and tries to link with another enterprise(through SSO), the learner is redirected to the login page instead of the registration page for the second enterprise. This is because the system detects that these credentials are already in the system.
2. Enrollment, Progression, Completion, Consent records become ambiguous if a learner completes a course from enterprise A and then attempts the same course through enterprise B.
3. Incorrect discounts are applied when an employee linked with multiple enterprises redeems the benefit applicable only to one of the enterprises.

10.3 Decisions

Short term updates:

As part of the login flow the user would select one of the linked enterprises. This selection would only happen once, after user authentication in the login flow.

Long term updates:

An account/profile switcher like we have in New Relic, Google, etc. with a flow similar to the following could be Implemented.

1. User flow will allow the learner to login without any enterprise related barriers.
2. After login, user is reminded that they are linked to multiple Enterprises and that they need to select the “profile” they intend to use during this logged-in session before executing any enrollment or other transaction in the system.
3. The enterprise “profile” would be specified using a “switch enterprise” page.

4. If a learner logs in via SSO with account-A, but intends to proceed using account-B, they would be logged out so that they can then re-login via account-B.
5. The text on the new “switch enterprise” page makes the above user-flow and selection as clear as possible for the learner.

10.4 Consequences

1. Discounts applied correctly for users connected to multiple enterprises.
2. Unambiguous transactional data history for enterprise users.

NEW ENTERPRISE CATALOG IDA

11.1 Status

Draft

11.2 Context

We are interested in improving the reliability and performance of our system at large, and we have identified that the enterprise catalog api and related functionality are a central dependency that often causes issues, such as users experiencing random failures during enrollment or other workflows. Almost every feature that our users interact with depends on enterprise catalogs: Enrollment, Code Redemption, Data Sharing Consent, Data transmission to external LMS systems, and API access given to select customers.

Our main goals are as follows:

1. Get the response time of the enterprise customer catalog endpoints to be below 1 second for 99% of calls.
2. Since many features do not hit the enterprise catalog api endpoints but perform the same calculations directly in code, we would also like to improve the performance of those code paths and drastically reduce the calls to the discovery service. Examples include the data sharing consent workflow, the enterprise enrollment api, and the task that “catches” enterprise enrollments from the B2C site.

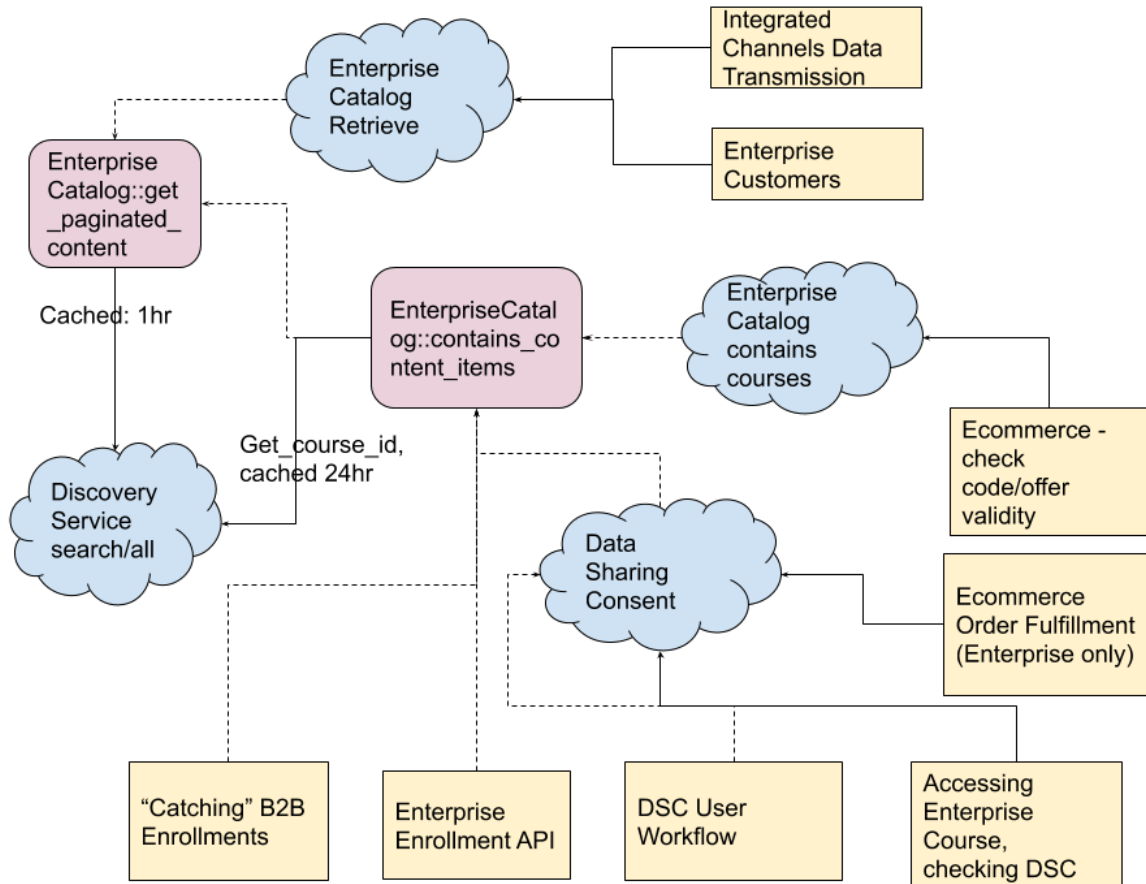
11.2.1 What is Enterprise Catalog?

Enterprise catalogs are essentially saved elasticsearch queries attached to an Enterprise Customer. There can be many catalogs configured for a given customer, which enables different sets of learners from the same Enterprise to be given access to different sets of courses, and allows different discounts to be configured for different sets of courses.

We expose two apis that are frequently used by various enterprise features: one to retrieve all of the content metadata that would be returned by the configured catalog query, and one to determine if a given course id or program id is part of the set of content returned by the catalog query. These apis exist both as python apis and REST apis that can be hit from within our edX ecosystem and externally from customers. They are wrappers around the search/all endpoint in the discovery service. Although there is caching in place, the calls to discovery are implemented synchronously, and we’ve found that half the time requests do not hit the cache, and thus the most of the time is spent waiting on the discovery service to return results.

11.2.2 Who's Calling What now?

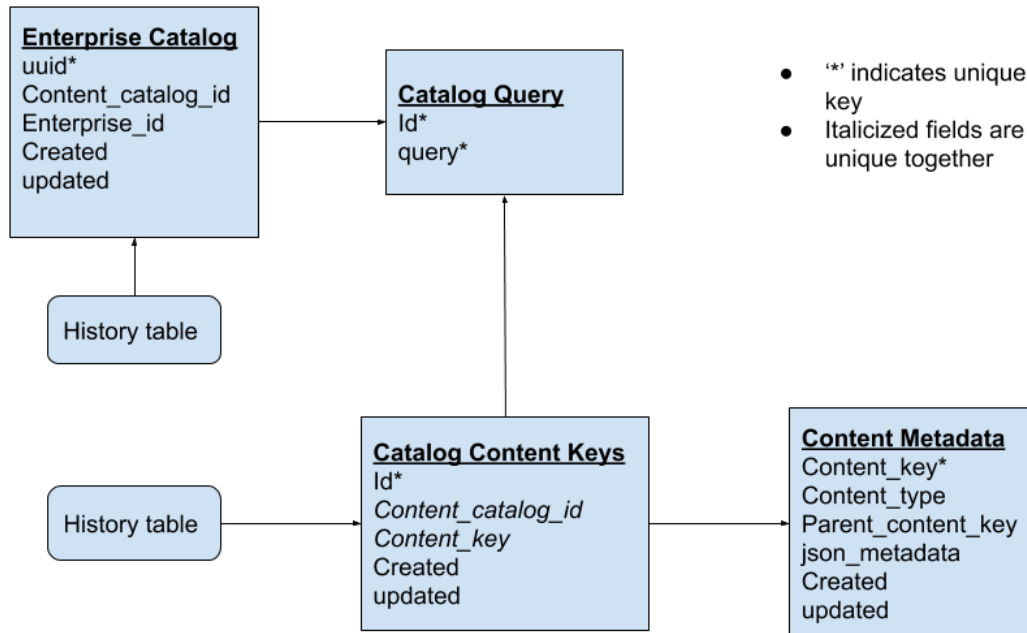
Here is a diagram showing a simplified view of the calls to these apis. The yellow boxes represent the features or user workflows that trigger calls to the various apis. The blue clouds represent REST apis, and the purple rounded boxes represent python code apis. Solid lines are synchronous api calls and dotted lines are code apis (not hitting the network).



11.3 Decisions

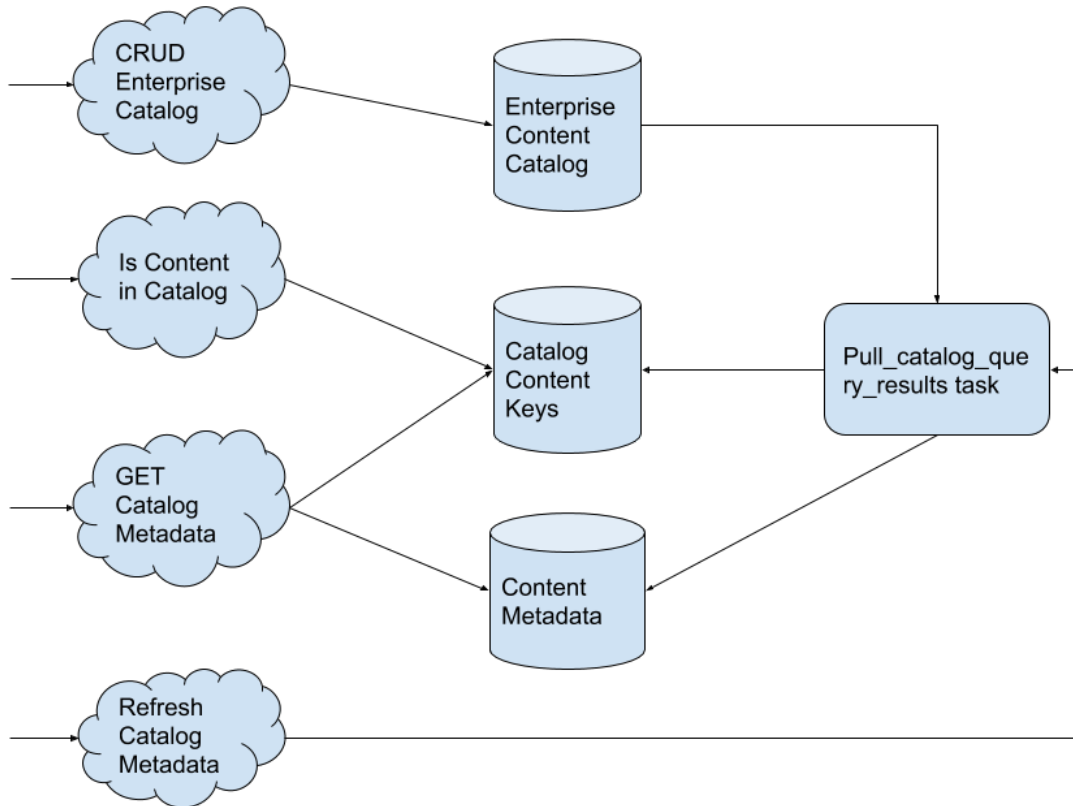
In order to address these performance issues and to more effectively scale and build new features related to enterprise catalogs in the future, we will create a new IDA (Independently Deployable Application) to replace the existing enterprise catalog functionality. This IDA will support the existing API contracts established in the edx-enterprise API, but will be a new implementation separate from what we do today.

The microservice will be supported by the following relational data model:



The microservice will expose the following APIs for use:

- CRUD Operations for Enterprise Catalog
- is_content_id_in_catalog: Given a course, course run, program, or other content id, and an enterprise catalog, returns true if the content id is contained within the catalog's current results.
- get_catalog_metadata: Given an enterprise catalog, return the full content metadata results for content currently in the catalog.
- pull_catalog_query_results: Given an enterprise catalog, kick off a task to get the freshest content metadata for that catalog.



All metadata will be fetched from the discovery service through asynchronous tasks that are entirely decoupled from when a call is made to get metadata for a particular catalog. We will guarantee that catalog data is fresh within the past 24 hours. Catalog refreshes will be performed through asynchronous tasks whenever a catalog is created or updated, when it is manually refreshed, or on a daily cadence. We will also gain efficiencies by deduping elasticsearch queries across different enterprise catalogs so that catalogs with the same query only fetch data from discovery once.

We will maintain the existing api signature so that the only update current api clients will have to make is to the url; request and response data will maintain the same format.

We will transition all internal api calls (both code and network calls) to use this new service. Once this transition is complete, we will deprecate and remove the original functionality contained in this repo.

11.4 Consequences

Transitioning to this new service will require careful consideration, since many internal systems as well as external clients rely on this API. Network calls should be simple enough to swap out, but within edx-enterprise there are many code api calls to enterprise catalogs, and these will be tougher to transition. We will need to be mindful of how we are caching data from this new service throughout our system. Hopefully, it can be kept to a minimum because this api should be much more performant.

This will also be the first enterprise specific IDA, which will require us to gain an understanding and ownership of a service from end to end. This should include monitoring and alerting SLAs that we expect the service to maintain. Hopefully this will allow us to move towards extracting more functionality in this repo out to microservices.

CANVAS INTEGRATION VIA CLASSIC INTEGRATION MODEL VS LTI

12.1 Status

Accepted

12.2 Context

We would like to capture points about the decision to choose the Class integration model over LTI model, in order to make edX courses available in Canvas LMS for partner institutions.

12.3 Decision

For Canvas integration, these following use cases are prioritized:

- Students can easily discover courses within their institution's online learning experience
- Faculty can easily view student performance data within their institution's online learning experience

There is another use case that is deemed not applicable w.r.t Canvas integration:

- Students can consume course content from within their institution's online learning experience

The classic integration model nicely supports the former use cases and allows students to take courses on the edX platform once discovered in Canvas.

Furthermore, LTI does not support course discovery, which is essential to independent and facilitated learning use-cases. hence we made a decision to choose the classic integration model.

12.4 Consequences

Choosing classic over LTI does mean that the "Digital Textbook" use case will be harder/clunkier to implement. LTI does very well as this. This factor may be re-considered in the future.

ENTERPRISE ROLE ASSIGNMENTS WILL RECORD A CUSTOMER UUID

13.1 Status

Accepted - January 2021

13.2 Context

The `SystemWideEnterpriseUserRoleAssignment` model is used to grant role-based permissions to enterprise users. These permissions generally control the users' access to data and ability to perform certain functions within an enterprise customer. This assignment model does *not* currently refer to any specific customer - the "context" to which a role is applied is inferred from a user (an `auth.User` record) having been "linked" to a customer via the `EnterpriseCustomerUser` model.

However, a single `auth.User` may be linked to multiple `EnterpriseCustomers`, and if that user is assigned a role (such as `enterprise_admin`), it is inferred that the role assignment applies to **every** customer to which the user is linked. This is undesirable; we generally do not want an admin of *Customer A* to also have admin permissions for *Customer B* (to which the user may be linked as a mere learner). Instead, the `SystemWideEnterpriseUserRoleAssignment` model should explicitly record *which* customer the assigned user has the given role in.

13.3 Decisions (interspersed with Consequences)

13.3.1 We'll add an `enterprise_customer` field to the assignment model

- It will be an optional field. It should be a foreign key on `EnterpriseCustomer.uuid`.
- Via an `edx-rbac` change, we will also add a boolean `applies_to_all_contexts` field, which is a wildcard that, if true, indicates the user has the role assigned for every customer. It should default to false.
- The semantics of `applies_to_all_contexts` imply that `enterprise_customer` will sometimes be null. For example, a user assigned the `enterprise_openedx_operator` role will have `applies_to_all_contexts` set to true and an `enterprise_customer` as null.
- We will add a customer validator to enforce that the `enterprise_customer` field can be null only when `applies_to_all_contexts` is true.

13.3.2 We'll backfill the values of this field

- We'll make the current, implicit assignments (that is, if a role is assigned to a user, that role applies to the user in every customer to which the user is linked) explicit by filling in the UUID value of the linked enterprise customer.
- For `enterprise_openedx_operator` roles, the `applies_to_all_contexts` field will be marked true.

13.3.3 We'll do a hand-audit of `enterprise_admin` role assignments

- For every user who is effectively an admin of two different enterprise customers, we'll have to manually determine (with help from our support team) which of the two customers they are actually an admin of, and which they are simply a learner within (unless they're truly an admin of both).

13.3.4 The context of a system-wide enterprise assignment is determined from `enterprise_customer`

- This is true for every role.
- If `applies_to_all` is true, `get_context()` should return the wildcard context token `"*"`.
- The code that populates the user JWTs with (role name, customer UUID) pairs shouldn't have to change - that code should only be looking at the `get_context()` method of the role assignment class.

13.3.5 Admin forms must record a value for `enterprise_customer`

- It would be nice to "auto-populate" it somehow if the user is linked to only one enterprise customer.

13.3.6 This field will be populated by signal handlers related to enterprise user CRUD operations

- Every user will get an `enterprise_learner` role assignment for the given enterprise.
- Pending admin "conversion" will result in an `enterprise_admin` role assignment for the given enterprise.
- We'll delete the role assignment when an `EnterpriseCustomerUser` is deactivated.

BULK ENROLLMENT ENDPOINT FOR SUBSCRIPTION LEARNERS

14.1 Status

Accepted

14.2 Context

Business need: The admin portal feature to enroll multiple learners in multiple subscription-courses needs a suitable backend.

Various enrollment apis exist, but they do not cater to or optimize for multiple enrollments.

There is an *EnrollmentApiClient* which has an *enroll_user_in_course()* method used by:

- `models::EnterpriseCustomerUser::enroll()`
- `utils::enroll_user()` (handles multiple courses)
- `views::CourseEnrollmentView::post()`
- `views::HandleConsentEnrollment::get()`
- `views::GrantDataSharingPermission::_enroll_learner_in_course()`

There is also a method to create pending user enrollments for non-edX users at:

`models::EnterpriseCustomer::enroll_user_pending_registration()`

This method also handles multiple courses.

Due to the scale and volume considerations, an api endpoint is needed that caters to the bulk use case without hitting rate limiting issues or server overload.

14.3 Decisions

- We will add a new endpoint to edx-enterprise to handle bulk enrollment in a single request.
- For existing edX users, we will reuse the `utils::enroll_user()` method. This method handles multiple courses already, but still calls the *EnrollmentApiClient* for now. There is a cleanup effort anticipated to replace the HttpClient calls embedded in the `utils::enroll_user()` method with direct database call utilities in the edx-platform Django application. Such a cleanup is not in scope for this feature work.
- For emails not matching existing users in edX LMS Django application, we will use the `models::EnterpriseCustomer::enroll_user_pending_registration()` method

- We will create an ecommerce order for each successful enrollment of an existing edX user. For this we will use the *EcommerceApiClient::create_manual_enrollment_orders()* method which creates an order for each enrollment passed.
- Any failures in individual enrollments will not cause failures of the entire batch. In other words, the endpoint will be non transactional. This avoid wasteful and confusing workflow for the client of the api, by performing a 'best effort' enrollment and reporting the failures.
- We will not have MFEs call this endpoint directly. Rather they will call a license-manager endpoint which will call this endpoint. There are several reasons for this: #. Only licensed learners should be enrolled in the subscription's courses. This design will guard against any frontend requests containing non-desired learners. #. Calling license-manager from edx-enterprise requires edx-enterprise to know about license-manager which is counter to the architectural design for edx-enterprise #. Finally, semantically it makes sense to request a license management service to authorize license based enrollments.
- The edx-enterprise endpoint will return a response with these pieces of information: *{ 'successes': [], 'pending': [], 'failures': [] }*. Note, pending enrollments are supported. This will allow the license-manager to inform the user if any enrollments failed, while successfully (meaning: success or pending) enrolling the rest, in a single request cycle.
- The endpoint will be safe to invoke with pre-existing enrollment pairs (meaning learner email + course_id). These enrollments will also be added to 'successes' in the response.
- The endpoint will also support a boolean input to notify learners of enrollments (or not) by email

14.4 Consequences

1. MFEs can make a single reliable request for bulk enrollment
2. Spurious enrollments of non-eligible learners is avoided
3. Partially enrolling 'what it can' and recording the rest in a response, allows for a better user experience for admins and the use of a single coherent endpoint.

ADDING ENTERPRISE_UUID IN COURSE ENROLLMENT EVENT

15.1 Status

Accepted

15.2 Context

`event_routing_backends` application needs `enterprise_uuid` in order to send transformed enrollment events to the relevant enterprise customers. At present, `enterprise_uuid` is not included in the course enrollment events generated by edx enterprise.

15.3 Decision

`enterprise_uuid` will be added in the course enrollment event generated by edx enterprise and will be added to the context of `EVENT_NAME_ENROLLMENT_ACTIVATED` emitted by edx platform.

15.4 Consequences

`enterprise_uuid` will be present in the context of `EVENT_NAME_ENROLLMENT_ACTIVATED` only if the user has provided an enterprise url during enrollment.

TRANSMITTING ONLY UPDATES IDA

16.1 Status

Draft

16.2 Context

The `transmit_content_metadata` jenkins job (along with the underlying `integrated_channels.transmit_content_metadata()` task and `export()` method) makes an uncomfortably large number of calls to the enterprise-catalog service's `/api/v1/enterprise-catalog/<uuid>/get_content_metadata` endpoint with a relatively low number of customers which results in an unacceptably high database load.

At the same time the data being requested by this frequently-running job is often unchanged between runs - content metadata tends to change in enterprise-catalog roughly once/day.

Our main goals are as follows:

1. Reduce the overall number of calls required by the Integrated Channels in order to retrieve the content metadata for any particular transmission.
2. Prevent the Integrated Channels from making an abundance of requests to the enterprise catalog if there is no update needed (ie if nothing in the catalog has changed)

16.2.1 What is the `transmit_content_metadata` job and underlying export method?

`transmit_content_metadata` is one of the three core tasks of the Integrated Channels, the other two being `transmit_learner_data` and `transmit_subsection_learner_data`. Specifically `transmit_content_metadata`'s job is to gather course metadata contained within the customer's enterprise catalog and transmit create, update and deletes of course entities, through customer configurations, to the external LMS'.

The `export()` method is the main method responsible for fetching said content metadata. It will take all listed *catalogs to transmit* under the customer's configurations (otherwise will default to all the customer's catalogs if *catalogs to transmit* is not provided) and request content metadata from the enterprise catalog. These catalogs can range from "all possible content" to a specific, single course.

16.3 Decisions

In order to address these performance issues and to more effectively handle onboarding more customers to the Integrated Channels, we have decided to address requests to the enterprise catalog on two fronts:

1. Increase the page size of requests going to `/api/v1/enterprise-catalog/<uuid>/get_content_metadata` from a base of 10 to 100. This will reduce the number of calls necessary to retrieve the entire catalog from 150~ to about 15 (as of 07/2021).
2. Add an additional request/check to the Integrated Channel's exporter that will retrieve the last time either a catalog or the associated content metadata of a catalog has been changed (whichever is most recent). This timestamp will be compared to the `content_last_changed` field of the `ContentMetadataItemTransmission` object, which is the *catalog last modified* value retrieved from the customer's most recent successful transmission. If there is no update needed, then the exporter will refrain from querying enterprise catalog for the catalog's metadata.

16.4 Exceptions

1. The Cornerstone (CSOD) LMS Integration will not use the 'transmit only updates' rule."
2. CSOD related transmissions will record `ContentMetadataItemTransmission` objects, same as all other channels.

16.5 Consequences

The `transmit-content-metadata` jenkins job can continue to run on a frequent schedule without having to worry about unacceptably large database loads, as it will record when updates are needed and will only request data when necessary. If any metadata associated with a catalog has changed between job run N and N+1, then we'll transmit the full catalog to the integrated channel during run N+1. It should be noted that there could be potential further improvements to the entire export process, as currently we are requesting the entire catalog and not only updated content. Reducing requests to only the updated metadata could further reduce database loads.

DETERMINING COURSE COMPLETION FOR VARIOUS COURSE MODES

17.1 Status

Draft

17.2 Context

LMS integrations have a learner_data module that reports out (to external LMS systems), a field (boolean) called *course_completed*, along with any available grading data.

For Non Audit-mode enrollments: the value *completed_date* is used as an indication of course completion. This is deterministic for courses that issue certificates or grading info

However, for Audit mode enrollments, there is a case where the learner has already finished all non-gated content. But in this state, *course_completed* will never be true for these learners who never upgrade to verified, for example. Also there is no more content learner can do anything with. So in this sense, they are done with the course.

Also, these learners don't always have access to all the content. So we cannot rely on certificate or grade information to detect if they are 'done' with a course.

In fact there is no one deterministic definition of *course_completed* for audit track.

Therefore, for LMS integration customers to get a sense of how audit learners are doing, with completion, we need to offer a best approximation sense of *course_completed*.

17.3 Decisions

Integrated channels will determine the *course_completed* for each enrollment with the following logic:

```
if audit_enrollment:
    Transmit course as complete for this learner, if no non-gated content is remaining.
    ↳ to be finished
else:
    Transmit course as complete if certificate or grading data indicates completion_date.
    ↳ is available.
```

17.4 Consequences

We will have more accurate reporting of completion of course (in some sense) for audit track. There is no precise way to define that a learner is done with a course unless we are issuing a certificate. Therefore this best approximation is being used for LMS integrations.

ADDING COURSE KEY - UUID MAPPING FOR CORNERSTONE

18.1 Status

Draft

18.2 Context

At one point we were having transmitted courses to CSOD tenants fail with invalid chars contained within course metadata. To combat this, we decided to encode the course keys. This has resulted in some of the courses failing validation with a “course key too long” failures. There’s a 50 char limit to course keys on the CSOD lms, so we need to limit the size of our encoded course keys within the content metadata.

Simply truncating it would not allow for backwards compatibility or ensuring uniqueness. We also considered encoding and then compress, but while this would most likely work for every course key (the largest length found in prod was 54 chars), it can’t properly guarantee the key to be under 50 chars, and we would have to worry with hashing the fact that it is one way.

18.3 Decisions

Using uuids as the course key as they are guaranteed to be 36 characters and don’t have to be url encoded. We will be creating a dictionary-like table to map the edx course key to a key that will be used as the course key in Cornerstone.

Since we already have many courses in Cornerstone that the keys are under 50 characters, we will not be creating new uuids in the dictionary, because that could lead to a course being duplicated on their system. The logic that is currently implemented (using url encoded course key) will remain, unless the encoding is over 50 characters. The uuid will be generated when we are transmitting course content metadata, and also when they report learner data back to us.

We also started an exploration of whether there are existing uuids we can use to map to course keys. We found that there are uuids in Discovery for courses: https://github.com/edx/course-discovery/blob/master/course_discovery/apps/course_metadata/models.py#L768 and courseruns: https://github.com/edx/course-discovery/blob/master/course_discovery/apps/course_metadata/models.py#L1277 but we are still in the stages of finding out whether they are stable throughout their lifespan to be used. Turns out only courses with ‘published’ state in discovery have stable uuids. Because there are some unknowns there, this fix will allow us a more immediate remediation.

18.4 Consequences

As always, adding an additional table leads to more memory storage, and have a customized approach to Cornerstone might lead to some confusion down the line. Code will be properly documented and lead back to this ADR to alleviate some of this.

ZERO STATE BROWSING WITH UNIVERSAL LINK

19.1 Status

DRAFT

19.2 Context

Today, the only way to access our learner portal is by explicit, named invitation from an administrator sent from the admin portal. The invitation can only be sent while assigning a subsidy (code or license). Our cost-conscious admins need a way to publicize their catalog and gauge interest before assigning a subsidy (spend).

We want to allow admins to generate a universal link that can be shared with their enterprise. Users visiting the universal link would be able to browse the enterprise catalog and request course access.

We will take a two-phase approach for the implementation of universal links.

19.2.1 Phase 1

Users visiting the link would be able to logistrate and be automatically associated with the enterprise. Users linked to an enterprise through a universal link will have no subsidies assigned and be in the “zero state”. Users in the “zero state” are shown an aggregate of all the catalogs for an enterprise on the search page and have access to all the tools that do not require a subsidy. They would then be able to browse the offerings and request course access.

19.2.2 Phase 2

We want to make enterprise catalogs publicly accessible through a universal link so that users who don't want to logistrate can still view a catalog. Users visiting this link would be able to browse the course offerings and request access. Upon course request, users will logistrate and be redirected to the learner portal course access request flow. Note that a user must be authenticated and associated with an enterprise to request course access.

19.3 Decision

19.4 Phase 1 - Browsing via the learner portal search page

Currently a `PendingEnterpriseCustomerUser` is created whenever a subsidy is assigned that associates a user to an enterprise through email. Upon logistration, we check for any `PendingEnterpriseCustomerUser` and links a user by replacing `PendingEnterpriseCustomerUser` with an `EnterpriseCustomerUser` and assigning the `enterprise_learner` role.

A user has to be associated with an enterprise to have the correct permissions when accessing our APIs. We have to link a user to an enterprise in order to support browsing without a subsidy. This can be accomplished by exposing a secured endpoint that links a user to an enterprise which gets called by the learner portal. A user linked this way will be in the “zero state” by default and be able to browse the learner portal freely.

A new endpoint `POST /link-enterprise/` that links a user and an enterprise will be created. The payload will contain `enterprise_customer_uuid`, `user_id`, and `enterprise_customer_key` - a UUID generated for the enterprise. A new model `EnterpriseCustomerInviteKey` will be created to store these keys.

`EnterpriseCustomerInviteKey` contains the following fields: `* uuid` - key used to generate the universal link/proof that a user had the universal link; `* enterprise_customer_uuid` - uuid of the enterprise; `* usage_limit` - the number of times a key can be used to link users; `* expiry_date` - defaults to the end date of the enterprise’s subscription plan; `* timestamps`.

CRUD endpoints for creating/deleting an `EnterpriseCustomerInviteKey` model will also be created.

A universal link can be generated by adding the key as a query param to the url that points to the learner portal for an enterprise, i.e. `/?enterprise_customer_key=key`. An enterprise admin will be able to generate/view links through the admin portal by calling `POST /EnterpriseCustomerInviteKey/` to generate a new key. Note that only enterprise admins should be able to generate a key, and only for enterprises they are an admin of. The generated link could also include the enterprise slug, but relying on the `enterprise_customer_key` instead to direct the user to the correct enterprise means that the link would still work if an enterprise changes its slug.

A user will be prompted to logistrate when visiting the link if not already logged in and be redirected to the learner portal afterwards. Currently, a not found page is shown on the learner portal if the user is not linked. If the user is not linked and an `enterprise_customer_key` is found in the query params, the portal will call the new endpoint to link the user. If the call succeeds, the portal will reload and the user will be able to browse as usual. If the call fails, appropriate messaging regarding the failure will be shown to the user. This will also link users that were already logged in before visiting the link.

19.4.1 Limiting logistration through universal link

Any authenticated user with a universal link that contains a valid key is able to link themselves with an enterprise. We want to take necessary precautions to prevent abuse of this feature. `EnterpriseCustomerInviteKey` will be expirable and have a limit on the number of times it can be used. We will validate the key (not expired and usage limit has not been reached) when linking a user to an enterprise. If the key is invalid, the user will not be linked and be informed that the link is invalid. We will also validate that the `enterprise_customer_uuid` in the request payload matches with the `enterprise_customer_uuid` associated with the key.

19.4.2 Limiting number of course requests to prevent email spamming

In the future we want to limit the number of course requests to prevent admins from being spammed with request notifications. This should be factored in when implementing course request and will not be addressed here.

19.4.3 Expiring a universal link

A universal link can have a lifetime dictated by the expiry date set on the `EnterpriseCustomerInviteKey` used in the link. A key will not be valid if expired. An enterprise admin should be able to manually expire a key to revoke a link.

19.4.4 Analytics

Segment events will be fired when * an `enterprise_customer_key` is created - i.e. `'edx.bi.user.enterprise.key.created'` * a user is linked to an enterprise using the universal link, include key used in the event - `'edx.bi.user.enterprise.key.used'` * a key expires or reaches its usage cap - `'edx.bi.user.enterprise.key.invalidated'` * a user attempts to use a bad key - `'edx.bi.user.enterprise.key.attempted'`

We will also keep track of which user were linked using a universal link in the model (possibly a new field on `EnterpriseCustomerUser`) and include that information in future tracking events, i.e. search events in the learner portal.

We also want to track how many users land on the logistration page but abandons the page. This could possibly be done through google analytics.

19.4.5 Consequences

- After a user requests course access and is granted a subsidy(ies), the learner portal will only display offerings associated with the subsidy(ies).
- The JWT token will need to be refreshed to include the `enterprise_learner` role after a user has been linked. The learner portal has logic to refresh the token if there are no roles included.
- The generated link would point to the learner portal search by default but having the `enterprise_customer_key` should allow a user to be linked from any page. We might want to redirect a user back to the search afterwards and strip the key from the url.
- Although key usage can be limited, users might still get unintended access to an `enterprise_customer_key` and link themselves to an enterprise. The admin portal will add a feature to list all of the learners associated with an enterprise and allow admins to manually remove them.

19.5 Phase 2 - Browsing anonymously

To support anonymous browsing, we will create a public page on the learner portal. Anyone with a universal link will be able to browse an enterprise catalog and request course access on this page. The learner portal will make calls to new/modified endpoints to fetch data and display the catalog. The phase 1 implementation enables users to be associated with an enterprise which is required for requesting course access. Note that the anonymous browsing component could be built before the phase 1 implementation if it doesn't include course access request.

The process of generating the universal link will remain largely the same as in phase 1. The link would point to this new page rather than the search page, ie. `/catalog?enterprise_customer_key=key`. We will also validate the `enterprise_customer_key` in the query params before rendering the page. Revoking a link will also be the same process as in phase 1.

Since the learner portal makes calls to protected endpoints, we have to add/modify them to support the new public catalog page. The following are the APIs that the learner portal interacts with to display the enterprise catalog/course information:

- edx-enterprise GET `/enterprise-customer/`
 - Fetches enterprise customer data such as uuid, `enterprise_customer_catalogs`, etc. by the enterprise slug.
 - This is a protected endpoint.
 - There is a lite version GET `/enterprise-customer/basic_list` that currently returns only id and name of an enterprise. We will modify this to also include the `enterprise_customer_catalogs` and that will be the minimal information we need to render the page.
- Algolia search API
 - Queries Algolia for catalog data.
 - This only requires an api key which the learner portal already has access to.
- course-discovery GET `/courses/{key}/`
 - Queries course-discovery to get course information once a user clicks on a course.
 - This is a protected endpoint.
 - Enterprise catalog also hosts course data. We could expose `enterprise-catalogs/get_content_metadata/{key}` as a public endpoint for the learner portal to query course data without going through course-discovery. The enterprise-catalog is synced daily with course-discovery.
- enterprise-catalog GET `/contains_content_items/`
 - Checks whether or not the specified content is available to the `EnterpriseCustomer`.
 - This is a protected endpoint.
 - We will make this endpoint public.

Authenticated users would not be able to view this page and instead be redirected to the normal search page since it's outside of the normal learner portal flow.

19.5.1 Consequences

- We have to expose data through public endpoints which leads to concerns. For any endpoints we add/modify, we have to keep the amount of information returned to the minimum.
- An `enterprise_customer_key` remains valid until it's expired or the usage limit has been reached. However the usage limit only refers to the number of users that can be linked using the key. If we want to limit the number of times the key can be used to view the public catalog, we can keep track the number of unique visits and add a constraint.
- Having a public catalog could potentially increase the load on our system. This is not a huge concern and we will monitor the number of calls made.

MOVING THE RESPONSIBILITY OF CATALOG DIFFING TO THE ENTERPRISE-CATALOG SERVICE

20.1 Status

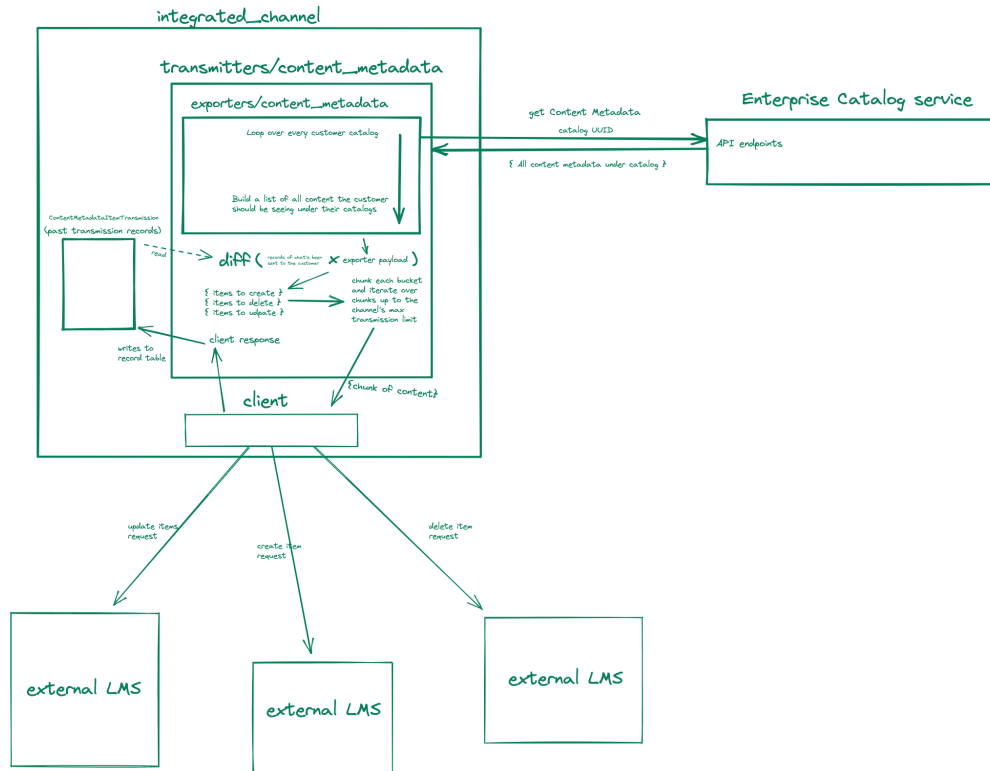
Accepted

20.2 Context

Recent, now reverted, attempts to make efficiency improvements to the integrated channels requesting data from the enterprise catalogs have highlighted short comings of the current architecture in terms of the capabilities of the Integrated Channels to accurately report the last time a customer's full catalog has been successfully updated. Because the Integrated Channels are responsible for chunking transmissions (only sending as much as the individual channel's API allows at a given time) such that not all content under a customer's catalog is guaranteed to be transmitted in a single scheduled run, we cannot accurately determine if all content under a customer's catalog has been updated since the catalog has last updated without first using the enterprise catalog's currently available API *get_content_metadata* endpoint.

- **How do the Integrated Channels currently handle customer catalog data?**

Existing flow chart:



Currently on a scheduled job, the Integrated Channels' content metadata exporter iterates over each customer's catalog and builds up a dict of content representing the current state of the customer's catalog content. Take away: as it stands now, the Integrated Channels assumes that the content metadata exporter will report the entirety of the customer's purchased content, then pass that payload to the content metadata transmitter which compares the payload with all saved *ContentMetadataItemTransmission* items under the customer to determine which items it (the transmitter) needs to create, update, and delete. It then chunks up each of the buckets, and sends those chunks (up to a limited number of times) to the client to send to the customer's LMS and saves the updated *ContentMetadataItemTransmission* entry.

- **Why does this matter?**

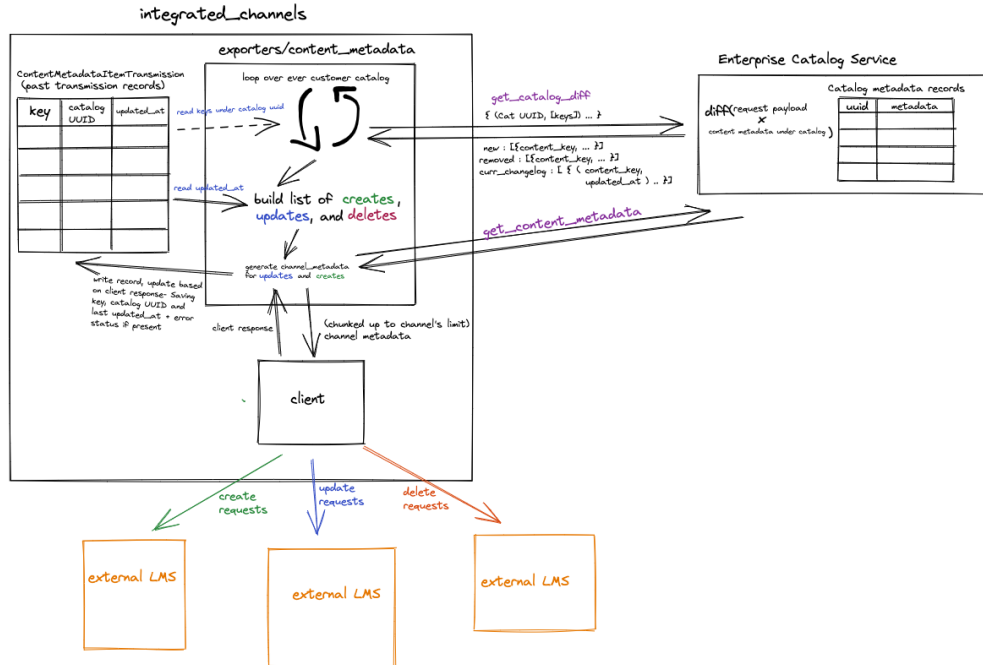
querying all content from the catalog service is expensive- Asking for all content under the catalog, for every customer, for every catalog, on every transmission is not only an undesirably large request to be sending, but also consumes much of the enterprise catalog's DB load. We need a smarter solution to how the Integrated Channels requests content from the catalog service.

putting the responsibility of determining differences between what's already been sent to the customers and what's currently in customer's catalogs adds much complexity to the integrated channels which can make it unapproachable from an onboarding perspective but also from a debugging point of view. Log noise and number of potential error sources means tracking down issues within the Integrated Channels are time consuming and thus, costly.

20.3 Decisions

- How the Integrated Channels could better determine if customer metadata needs updating

Proposed flow chart:



When we build an endpoint that takes a catalog UUID and a set of content keys linked to the uuid and returns three buckets of data:

- 1) What the last `updated_at` times of the content keys are for the content keys provided that are under the specified catalog
- 2) What content keys exist under the specified catalog that were not provided in the list of content keys
- 3) What content keys don't exist under the specified catalog that were provided in the list of content keys

We would be able to move the responsibility of diffing what we believe the customer has with what the customer should have, to the enterprise-catalog service via a new API endpoint. This would mean that the only job of the Integrated Channels would be to compare the last updated times of the content that already exist to get which content need updates, creates and deletes would already determined by the first endpoint.

20.4 Consequences

Adding the new diffing enterprise-catalog service API endpoint would allow us to accurately determine if catalog updates are needed for any particular iteration of the content metadata transmissions. We would be able to limit the number of calls needed to the enterprise-catalog service, saving time and DB load since the `content_metadata` endpoint is a hefty one. It would also allow us to easily implement a filtered content retrieval endpoint in the future such that we won't have to fetch any extraneous content whatsoever. Additionally these changes cut back on the complexity of the integrated channels, making it easier to approach and accurately identify future issues. Lastly, it moves the source of truth of catalog content information to the enterprise-catalog service and out of the Integrated Channels which should not be handling that responsibility in the first place.

20.5 Further Improvements

There are two additional improvements to explore with the Integrated Channels' metadata transmission flow. Firstly, we remove records from our transmission audit table when a delete request is issued. This can result in *lost* nodes of content (as we've actively seen from customers) where content can be assumed to be deleted on our end but have it exist still on the customer's external LMS. The fix here is pretty easy- if we choose to not delete records, but rather create a new, nullable *deleted_at* or equivalent field in the transmission record table. Then we would be able to mark and exclude any deleted records from appropriate look ups, but still have something to help us identify past courses that were sent to customers.

As stated earlier, currently the only method to retrieve content under a customer's catalog is the singular, bulk *get_content_metadata* endpoint. This returns all content metadata belonging to an enterprise catalog. If we were to build out the existing endpoint where individual content key's worth of metadata could be specified in the request body, then we would be able to further reduce the amount the Integrated Channels request unused data.

ENTERPRISE SUBSIDY ENROLLMENTS AND ENTITLEMENTS

21.1 Status

Accepted

21.2 Context

Enterprise course enrollment and course entitlement records can come into existence from different types of subsidies:

- Subscription licenses - we'd want to track the license uuid at time of enrollment creation.
- Learner credit subsidy transactions - we'd want to track the transaction uuid at time of enrollment or entitlement creation.
- Coupon codes - not yet tracked in the context of this ADR.

On top of architecting the tables to support subsidy licenses and credit transactions we want to build a system that can support other, not yet known types of subsidies as there is the chance that new ones will be added in the future.

We will create a new `EnterpriseCourseEntitlement` model, which will mirror the purpose of the `EnterpriseCourseEnrollment` model but instead track a subsidy that is not yet connected to a course enrollment and will support converting to said enrollment.

21.3 Decisions

Why have `EnterpriseCourseEntitlement`s`?

There are situations where enterprise admins give a learner a subsidy for a specific course, but that course does not yet have a valid run for the learner to enroll in. Entitlements allow us to track the subsidy for the specific user before the enrollment can be created so that when the time comes for the learner to start/create the enrollment, the entitlement can be easily converted and used. These entitlements can also be gathered, tracked and provided as a metric for the admins of any enterprise customer.

It is also important that these models link to and mirror the state of the B2C enrollment and entitlement models (both `CourseEntitlement` and `CourseEnrollment`), such that there is a 1:1 relationship between B2C and B2B rows for enterprise related enrollments and entitlements. Meaning that when we write an enterprise enrollment or entitlement, we should also create the B2C counterpart for that record.

Why implement abstract table inheritance and what is `EnterpriseFulfillmentSource``?

- We can more easily track the entitlement to enrollment lifecycle.
- The null/not-null data integrity constraints are easy to grok and helps keep our data valid.

- Inheritance means that things are easily extendable. As we onboard new types of fulfillments, we can simply add new child models.

Benefits of this rework

- It helps standardize subsidy consumption for multiple content types and providers, given that the content is represented in the edX catalog service.
- We can do this in a de-coupled way with Event-bus and/or polling.
- We can easily support new subsidy types

21.4 A rework of the enterprise subsidy enrollment models and creation of enterprise entitlements

The new enterprise entitlement table:

EnterpriseCourseEntitlement

- uuid, created, modified, history (boilerplate)
- enterprise_customer_user_id (NOT NULL, FK to EnterpriseCustomerUser)
- enterprise_course_enrollment_id (NULL, FK to EnterpriseCourseEnrollment)
- converted_at (NULL DateTime).
- (cached property) course_entitlement_id (query look up of related CourseEntitlement)

– TBD: A built in method of entitlement conversion to enrollment

Reworked and added table inheritance to all subsidy based enrollment tables. As such all subsidy based fulfillment records will have access to the following fields:

EnterpriseFulfillmentSource

- uuid, created, modified, history (boilerplate)
- fulfillment_type (NOT NULL, char selection: (*license, learner_credit, coupon_code*))
- enterprise_course_entitlement (NOT NULL, FK to EnterpriseCourseEntitlement)
- enterprise_course_enrollment (NOT NULL, FK to EnterpriseCourseEnrollment)
- is_revoked (Default False, Bool)

Models inheriting *EnterpriseFulfillmentSource*:

LicensedEnterpriseCourseEnrollment (inherited from EnterpriseFulfillmentSource)

- license_uuid (NOT NULL, UUID field)

LearnerCreditEnterpriseCourseEnrollment (inherited from EnterpriseFulfillmentSource)

- transaction_id (NOT NULL, UUID field)

[NOTE] Even though these models are labeled as ...*Enrollment*'s, they can reference entitlements as well as enrollments. In fact, despite both `enterprise_course_entitlement` and `enterprise_course_enrollment` both being nullable, there is validation on the `EnterpriseFulfillmentSource` which will guarantee one of these values must exist.

To support interactions with these reworked and new models, we've buffed out the bulk enrollment (`enroll_learners_in_courses`) `EnterpriseCustomerViewSet` view to support subsidy enrollments. `enrollment_info` parameters supplied to the endpoint can now include transaction ID's that will be detected and realized into a `LearnerCreditEnterpriseCourseEnrollment` record.

How we'd use this in code

```
# In the parent class...
@classmethod
def get_fulfillment_source(cls, enrollment_id, entitlement_id=None):
    return cls.objects.select_related(
        # all child tables joined here
    ).filter(
        cls.enterprise_course_enrollment=enrollment_id
    )
    # do kwargs stuff here to optionally pass in a non-null
    # entitlement id to filter by...

@property
def fulfillment_status(self):
    if not self.enterprise_course_enrollment:
        return 'entitled'
    return 'enrolled'
```

21.5 Consequences

- Table inheritance means that we'll most likely have to do JOINS in our code and in our analytics/reporting.
- There exists a subsidy based enrollment table already (`LicensedEnterpriseCourseEnrollment`), any form of concrete inheritance added to this existing table would result in needing a data migration. Using abstract model inheritance avoids this complexity.

21.6 Further Improvements

- Verify transaction ID's are real on creation through the bulk enrollment endpoint. Note that this is not wholly necessary as we expect requests to come from an authenticated `__system__` user.
- Add a programatic way to turn entitlements into enrollments
- Continue extending the `enroll_learners_in_courses` endpoint to support bulk entitlement creation of entitlements. (suggestion here is that if course run keys are supplied for enrollments, if course uuid's are supplied then we generate entitlements instead)

21.7 Alternatives Considered

- *Concrete model inheritance*: Concretely inherit the children of the subsidy fulfillment table meaning that shared fields would be contained in the parent table. However, due to complexity required to do data and schema migrations was too great to justify the costs and risks. Abstract inheritance still gives us the benefits of:

1. flexibility to have subsidy-specific methods and fields on the child models.
2. common behavior of methods via the base class.
3. common and unique uuid field for easy reference across systems (e.g. the uuid is what's stored on ledger transactions to commit them).

and as for reporting and analytics, we can easily write a view to aggregate these two tables into a single relation especially with a fulfillment type column.

- *One big table*: Jam everything into one big table; almost every field is optional - might do code-level validation in the model's save() method to ensure the presence of non-null fields depending on type of fulfillment.
- *Table-hierarchy based on FK relationships*: Instead of strict inheritance, we could implement subsidy based tables that rely on foreign keys instead. This option was dropped as it required all the same work as concrete table inheritance with a subset of the benefits.

WAFFLE-BASED FEATURE FLAGS FOR ENTERPRISE

22.1 Status

Accepted (September 2023)

22.2 Context

Enterprise typically uses environment configuration to control feature flags for soft/dark releases. For micro-frontends, this control involves environment variables that are set at build time and used to compile JavaScript source or configuration settings derived from the runtime MFE configuration API in edx-platform. For backend APIs, this control typically either involves configuration settings or, on occasion, a Waffle flag.

By solely relying on environment configuration, we are unable to dynamically control feature flags in production based on the user's context.

For example, we may want to enable a feature for all staff users but keep it disabled for customers/users while it's in development. Similarly, we may want to enable a feature for a subset of specific users (e.g., members of a specific engineering squad) in production to QA before enabling it for all users.

However, neither of these are really possible with environment configuration.

22.3 Decisions

We will adopt the Waffle-based approach to feature flags for Enterprise micro-frontends in favor of environment variables or the MFE runtime configuration API. This approach will allow us to have more dynamic and granular control over feature flags in production based on the user's context (e.g., all staff users have a feature enabled, a subset of users have a feature enabled, etc.).

22.4 Consequences

- We are introducing a third mechanism by which we control feature flags in the Enterprise micro-frontends. We may want to consider migrating other feature flags to this Waffle-based approach in the future. Similarly, such an exercise may be a good opportunity to revisit what feature flags exist today and what can be removed now that the associated features are stable in production.
- The majority of the feature flag setup for Enterprise systems lives in configuration settings. By moving more towards Waffle, the feature flag setup will live in databases and be modified via Django Admin instead of via code changes.

22.5 Further Improvements

- To further expand on the capabilities of Waffle-based feature flags, we may want to invest in the ability to enable such feature flags at the **enterprise customer** layer, where we may be able to enable soft-launch features for all users linked to an enterprise customer without needing to introduce new boolean fields on the `EnterpriseCustomer` model.

22.6 Alternatives Considered

- Continue using the environment variable to enabling feature flags like Enterprise has been doing the past few years. In order to test a disabled feature in production, this approach requires developers to allow the environment variable to be intentionally overridden by a `?feature=` query parameter in the URL. Without the query parameter in the URL, there is no alternative ways to temporarily enable the feature without impacting actual users and customers. Waffle-based feature flags give much more flexibility to developers to test features in production without impacting actual users and customers.
- Exposes a net-new API endpoint specific to returning feature flags for Enterprise needs. This approach was not adopted as it would require new API integrations within both the enterprise administrator and learner portal micro-frontends, requiring users to wait for additional network requests to resolve. Instead, we are preferring to include Waffle-based feature flags on existing API endpoints made by both micro-frontends.

0013 CREATING CUSTOM GROUPS FOR ENTERPRISE CUSTOMERS

23.1 Status

Accepted

23.2 Terminology

Enterprise Group - Collection of enterprise customer users

Membership - Relationship between enterprise learner and enterprise group

23.3 Context

Groups is a new feature that will allow admins further customization and control of their learners. At a high level, a group is a subset of learners within an organization that can be tied to an enterprise customer's policy. A customer can have multiple policies and groups, and a learner can be in multiple groups. These changes will improve the user management and budgeting from the admin perspective, as well as adding personalization to the learner experience. In the past, we've used workarounds to sidestep this functionality, but with this new implementation, we will increase scalability and personalization.

In this MVP, groups will be associated with a learner credit budget, though they are something that exists independently. Groups will allow for analytics segmentation, bulk actions, access control, and more throughout the admin experience.

23.4 EnterpriseGroup

23.4.1 Model properties

- uuid, name, created, modified, history (boilerplate)
- enterprise_customer_uuid (NOT NULL, FK to EnterpriseCustomer, related_name="groups")

23.4.2 CRUD

`api/v1/enterprise-group/<group_uuid>`

Outputs

The root URL for getting the basic information about the group

```
{
  'group_uuid': 'group_uuid',
  'name': 'group_name',
  'enterprise_customer_uuid': 'enterprise_customer_uuid',
}
```

`api/v1/enterprise-group/?learner_uuid=[enterprise_customer_user_id]&enterprise_uuid=[enterprise_uuid]`

Accepted Query Params

- `learner_uuid` (optional): Get all the groups that the learner is associated with
- `enterprise_uuid` (optional): Get all the groups under the enterprise

Outputs

Returns a paginated list of groups filtered by the query params

```
{
  'count': 1,
  'next': null,
  'previous': null,
  'results': [
    {
      'group_uuid': 'group_uuid',
      'name': 'group_name',
      'enterprise_customer_uuid': 'enterprise_customer_uuid',
    }
  ]
}
```

23.4.3 GET (list) /learners

`api/v1/enterprise-group/<group_uuid>/learners/`

Outputs

Returns a paginated list of learners that are associated with the enterprise group uuid

```
{
  'count': 1,
  'next': null,
  'previous': null,
  'results': [
    {
      'learner_uuid': 'enterprise_customer_user_id',
      'enterprise_group_membership_uuid': 'enterprise_group_membership_uuid',
    }
  ]
}
```

23.4.4 POST /assign_learners

api/v1/enterprise-group/<group_uuid>/assign_learners

Inputs

- learner_uuids (POST data, required): A list of enterprise_customer_user_ids to assign to the group

Outputs

Returns a list of the EnterpriseGroupMembership objects that were created

```
{
  'count': 1,
  'next': null,
  'previous': null,
  'results': [
    {
      'learner_uuid': 'enterprise_customer_user_id',
      'enterprise_group_membership_uuid': 'enterprise_group_membership_uuid',
    }
  ]
}
```

23.4.5 POST /remove_learners

api/v1/enterprise-group/<group_uuid>/remove_learners

Inputs

- `learner_uuids` (POST data, required): A list of `enterprise_customer_user_ids` to assign to the group

23.5 EnterpriseGroupMembership

23.5.1 Model properties

- `uuid`, `created`, `modified`, `history` (boilerplate)
- `group` (NOT NULL, FK to `EnterpriseGroup` with related name `members`)
- `enterprise_customer_user_id` (FK to `EnterpriseCustomerUser` with related_name of `memberships`)
- `pending_enterprise_customer_user_id` (FK to `PendingEnterpriseCustomerUser` with related_name of `pending_memberships`)

23.6 Consequences

Now with the implementation of groups, this will be another facet that we will filter on. Now, not all learners under organizations necessarily have equal access to content. These subsets will provide a more personalized experience for the learner, and more control for the admin.

23.7 Further Improvements

Groups will have analytics, learning goals, and other customizations associated with them in the future

PYTHON MODULE INDEX

e

enterprise, 273
enterprise.admin, 42
enterprise.admin.actions, 25
enterprise.admin.forms, 25
enterprise.admin.paginator, 35
enterprise.admin.utils, 35
enterprise.admin.views, 37
enterprise.admin.widgets, 42
enterprise.api, 98
enterprise.api.filters, 94
enterprise.api.pagination, 95
enterprise.api.throttles, 96
enterprise.api.urls, 97
enterprise.api.utils, 97
enterprise.api.v1, 94
enterprise.api.v1.decorators, 79
enterprise.api.v1.fields, 80
enterprise.api.v1.permissions, 80
enterprise.api.v1.serializers, 80
enterprise.api.v1.urls, 94
enterprise.api.v1.views, 79
enterprise.api.v1.views.analytics_summary, 55
enterprise.api.v1.views.base_views, 55
enterprise.api.v1.views.coupon_codes, 56
enterprise.api.v1.views.enterprise_catalog_query, 57
enterprise.api.v1.views.enterprise_course_enrollment, 58
enterprise.api.v1.views.enterprise_customer, 60
enterprise.api.v1.views.enterprise_customer_api_credentials, 62
enterprise.api.v1.views.enterprise_customer_branding_configuration, 64
enterprise.api.v1.views.enterprise_customer_catalog, 65
enterprise.api.v1.views.enterprise_customer_invite_key, 68
enterprise.api.v1.views.enterprise_customer_reporting, 69
enterprise.api.v1.views.enterprise_customer_sso_configuration, 70
enterprise.api.v1.views.enterprise_customer_user, 71
enterprise.api.v1.views.enterprise_group, 72
enterprise.api.v1.views.enterprise_subsidy_fulfillment, 74
enterprise.api.v1.views.notifications, 77
enterprise.api.v1.views.pending_enterprise_customer_user, 78
enterprise.api.v1.views.plotly_auth, 79
enterprise.api_client, 113
enterprise.api_client.braze, 98
enterprise.api_client.client, 99
enterprise.api_client.discovery, 100
enterprise.api_client.ecommerce, 103
enterprise.api_client.enterprise, 103
enterprise.api_client.enterprise_catalog, 104
enterprise.api_client.lms, 106
enterprise.api_client.open_ai, 112
enterprise.api_client.sso_orchestrator, 112
enterprise.apps, 128
enterprise.config, 114
enterprise.config.models, 113
enterprise.constants, 129
enterprise.decorators, 130
enterprise.errors, 131
enterprise.forms, 132
enterprise.heartbeat, 117
enterprise.heartbeat.checks, 114
enterprise.heartbeat.exceptions, 115
enterprise.heartbeat.throttles, 116
enterprise.heartbeat.utils, 116
enterprise.heartbeat.views, 116
enterprise.logging, 132
enterprise.management, 127
enterprise.management.commands, 127
enterprise.management.commands.add_executed_exclusion_to_catalog, 117
enterprise.management.commands.backfill_learner_role_assignment, 117
enterprise.management.commands.bulk_update_catalog_query_info, 118

- [enterprise.management.commands.compare_discovery_and_enterprise_catalogs,](#)
[118](#)
- [enterprise.management.commands.create_enterprise_course_enrollments,](#)
[118](#)
- [enterprise.management.commands.create_missing_dsc_records,](#)
[119](#)
- [enterprise.management.commands.email_drip_for_missing_dsc_records,](#)
[119](#)
- [enterprise.management.commands.ensure_singular_active_enterprise_customer_user,](#)
[120](#)
- [enterprise.management.commands.fix_dsc_records,](#)
[120](#)
- [enterprise.management.commands.manufacture_data,](#)
[120](#)
- [enterprise.management.commands.migrate_enterprise_catalogs,](#)
[122](#)
- [enterprise.management.commands.monthly_impact_report,](#)
[122](#)
- [enterprise.management.commands.nudge_dormant_enrolled_enterprise_learners,](#)
[123](#)
- [enterprise.management.commands.reencrypt_enterprise_customer_reporting_config_passwords,](#)
[123](#)
- [enterprise.management.commands.remove_expired_pending_group_memberships,](#)
[123](#)
- [enterprise.management.commands.revert_enrollment_objects,](#)
[124](#)
- [enterprise.management.commands.save_enterprise_customer_users,](#)
[124](#)
- [enterprise.management.commands.seed_enterprise_devstack_data,](#)
[125](#)
- [enterprise.management.commands.unlink_enterprise_customer_learners,](#)
[125](#)
- [enterprise.management.commands.update_config_last_errored_at,](#)
[126](#)
- [enterprise.management.commands.update_role_assignments_with_customers,](#)
[126](#)
- [enterprise.messages,](#) [133](#)
- [enterprise.middleware,](#) [134](#)
- [enterprise.models,](#) [134](#)
- [enterprise.roles_api,](#) [244](#)
- [enterprise.rules,](#) [245](#)
- [enterprise.settings,](#) [127](#)
- [enterprise.settings.test,](#) [127](#)
- [enterprise.signals,](#) [245](#)
- [enterprise.tasks,](#) [246](#)
- [enterprise.templatetags,](#) [128](#)
- [enterprise.templatetags.enterprise,](#) [127](#)
- [enterprise.toggles,](#) [247](#)
- [enterprise.tpa_pipeline,](#) [247](#)
- [enterprise.urls,](#) [248](#)
- [enterprise.utils,](#) [248](#)
- [enterprise.validators,](#) [266](#)
- [enterprise.views,](#) [267](#)

INDEX

A

abstract (*enterprise.models.EnterpriseFulfillmentSource.Meta* attribute), 186
actions (*enterprise.admin.EnterpriseCustomerAdmin* attribute), 45
actions (*enterprise.admin.EnterpriseCustomerCatalogAdmin* attribute), 48
activate_admin_permissions() (in module *enterprise.api*), 98
activated_at (*enterprise.models.EnterpriseGroupMembership* attribute), 189
activated_at (*enterprise.models.HistoricalEnterpriseGroupMembership* attribute), 217
active (*enterprise.models.EnterpriseCustomer* attribute), 148
active (*enterprise.models.EnterpriseCustomerReportingConfiguration* attribute), 170
active (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 173
active (*enterprise.models.EnterpriseCustomerUser* attribute), 178
active (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 198
active (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 209
active (*enterprise.models.HistoricalEnterpriseCustomerUser* attribute), 212
active_customers (*enterprise.models.EnterpriseCustomer* attribute), 148
add_arguments() (*enterprise.management.commands.backfill_learner_role_assignments.Command* method), 117
add_arguments() (*enterprise.management.commands.bulk_update_catalog_query.Command* method), 118
add_arguments() (*enterprise.management.commands.create_enterprise_course_enrollments.Command* method), 118
add_arguments() (*enterprise.management.commands.create_missing_dsc_records.Command* method), 119
add_arguments() (*enterprise.management.commands.email_drip_for_missing_dsc_records.Command* method), 119
add_arguments() (*enterprise.management.commands.ensure_singular_active_enterprise.Command* method), 120
add_arguments() (*enterprise.management.commands.fix_dsc_records.Command* method), 120
add_arguments() (*enterprise.management.commands.manufacture_data.Command* method), 121
add_arguments() (*enterprise.management.commands.migrate_enterprise_catalogs.Command* method), 122
add_arguments() (*enterprise.management.commands.monthly_impact_report.Command* method), 122
add_arguments() (*enterprise.management.commands.nudge_dormant_enrolled_enterprises.Command* method), 123
add_arguments() (*enterprise.management.commands.remove_expired_pending_group_members.Command* method), 124
add_arguments() (*enterprise.management.commands.revert_enrollment_objects.Command* method), 124
add_arguments() (*enterprise.management.commands.save_enterprise_customer_users.Command* method), 125
add_arguments() (*enterprise.management.commands.seed_enterprise_devstack_data.Command* method), 125
add_arguments() (*enterprise.management.commands.unlink_enterprise_customer_learners.Command* method), 125
add_arguments() (*enterprise.management.commands.update_role_assignments_with_courses.Command* method), 126
add_arguments() (*enterprise.management.commands.manufacture_data.Node* method), 121
add_child() (*enterprise.management.commands.manufacture_data.Node* method), 121
add_consent_declined_message() (in module *enterprise*), 121

- `prise.messages`), 133
- `add_generic_error_message_with_code()` (in module `enterprise.messages`), 133
- `add_missing_price_information_message()` (in module `enterprise.messages`), 133
- `add_reason_to_failure_url()` (in module `enterprise.views`), 272
- `add_unenrollable_item_message()` (in module `enterprise.messages`), 133
- `additional_fields` (enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyRoleSerializer.Meta attribute), 87
- `admin_notification` (enterprise.models.AdminNotificationRead attribute), 137
- `admin_notification_filter` (enterprise.models.AdminNotification attribute), 134
- `admin_notification_id` (enterprise.models.AdminNotificationRead attribute), 137
- `admin_registration_url` (enterprise.models.PendingEnterpriseCustomerAdminUser attribute), 238
- `admin_role()` (in module `enterprise.roles_api`), 244
- `AdminNotification` (class in `enterprise.models`), 134
- `AdminNotification.DoesNotExist`, 134
- `AdminNotification.MultipleObjectsReturned`, 134
- `AdminNotificationAdmin` (class in `enterprise.admin`), 42
- `AdminNotificationAPIRequestError`, 131
- `AdminNotificationFilter` (class in `enterprise.models`), 136
- `AdminNotificationFilter.DoesNotExist`, 136
- `AdminNotificationFilter.MultipleObjectsReturned`, 136
- `AdminNotificationFilterAdmin` (class in `enterprise.admin`), 42
- `AdminNotificationForm` (class in `enterprise.admin.forms`), 25
- `AdminNotificationForm.Meta` (class in `enterprise.admin.forms`), 25
- `AdminNotificationRead` (class in `enterprise.models`), 137
- `AdminNotificationRead.DoesNotExist`, 137
- `AdminNotificationRead.MultipleObjectsReturned`, 137
- `adminnotificationread_set` (enterprise.models.AdminNotification attribute), 134
- `adminnotificationread_set` (enterprise.models.EnterpriseCustomerUser attribute), 178
- `AdminNotificationReadAdmin` (class in `enterprise.admin`), 42
- `AdminNotificationSerializer` (class in `enterprise.api.v1.serializers`), 80
- `AdminNotificationSerializer.Meta` (class in `enterprise.api.v1.serializers`), 80
- `alert_messages()` (in module `enterprise.template_tags.enterprise`), 127
- `all_objects` (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 173
- `all_objects` (enterprise.models.EnterpriseCustomerUser attribute), 178
- `all_subclasses()` (in module `enterprise.management.commands.manufacture_data`), 121
- `allow_request()` (enterprise.api.throttles.BaseThrottle method), 96
- `ALLOWED_API_GROUPS` (enterprise.api.v1.permissions.IsInEnterpriseGroup attribute), 80
- `ALLOWED_NON_COMPRESSION_DATA_TYPES` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
- `AnalyticsSummarySerializer` (class in `enterprise.api.v1.serializers`), 80
- `AnalyticsSummarySerializer.LearnerEngagementSerializer` (class in `enterprise.api.v1.serializers`), 80
- `AnalyticsSummarySerializer.LearnerProgressSerializer` (class in `enterprise.api.v1.serializers`), 81
- `AnalyticsSummaryView` (class in `enterprise.api.v1.views.analytics_summary`), 55
- `API` (enterprise.models.EnterpriseEnrollmentSource attribute), 182
- `API_BASE_URL` (enterprise.api_client.client.APIClientMixin attribute), 99
- `API_BASE_URL` (enterprise.api_client.discovery.NoAuthDiscoveryClient attribute), 102
- `API_BASE_URL` (enterprise.api_client.ecommerce.NoAuthEcommerceClient attribute), 103
- `API_BASE_URL` (enterprise.api_client.enterprise.EnterpriseApiClient attribute), 103
- `API_BASE_URL` (enterprise.api_client.enterprise_catalog.EnterpriseCatalog attribute), 104
- `API_BASE_URL` (enterprise.api_client.enterprise_catalog.NoAuthEnterpriseCatalog attribute), 106
- `API_BASE_URL` (enterprise.api_client.lms.CertificatesApiClient attribute), 106
- `API_BASE_URL` (enterprise.api_client.lms.CourseApiClient attribute), 107
- `API_BASE_URL` (enterprise.api_client.lms.EnrollmentApiClient attribute), 107
- `API_BASE_URL` (enterprise.api_client.lms.GradesApiClient attribute), 110
- `API_BASE_URL` (enterprise.api_client.lms.NoAuthLMSClient

attribute), 111

API_BASE_URL (enterprise.api_client.lms.ThirdPartyAuthApiClient attribute), 138

attribute), 111

APIClientMixin (class in enterprise.api_client.client), 99

APICredEnabledPermission (class in enterprise.api.v1.views.enterprise_customer_api_credentials), 62

APICredentialsRegenerateViewSet (class in enterprise.api.v1.views.enterprise_customer_api_credentials), 62

APICredentialsViewSet (class in enterprise.api.v1.views.enterprise_customer_api_credentials), 63

APPEND_SLASH (enterprise.api_client.client.APIClientMixin attribute), 99

APPEND_SLASH (enterprise.api_client.discovery.CourseCatalogApiClient attribute), 100

APPEND_SLASH (enterprise.api_client.discovery.NoAuthDiscoveryClient attribute), 102

APPEND_SLASH (enterprise.api_client.ecommerce.NoAuthEcommerceClient attribute), 103

APPEND_SLASH (enterprise.api_client.enterprise.EnterpriseApiClient attribute), 103

APPEND_SLASH (enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient attribute), 104

APPEND_SLASH (enterprise.api_client.enterprise_catalog.NoAuthEnterpriseCatalogClient attribute), 106

APPEND_SLASH (enterprise.api_client.lms.CertificatesApiClient attribute), 106

APPEND_SLASH (enterprise.api_client.lms.CourseApiClient attribute), 107

APPEND_SLASH (enterprise.api_client.lms.GradesApiClient attribute), 110

APPEND_SLASH (enterprise.api_client.lms.NoAuthLMSClient attribute), 111

applies_to_all_contexts (enterprise.models.EnterpriseFeatureUserRoleAssignment attribute), 184

applies_to_all_contexts (enterprise.models.EnterpriseGroup attribute), 187

applies_to_all_contexts (enterprise.models.HistoricalEnterpriseGroup attribute), 215

applies_to_all_contexts (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 231

applies_to_all_contexts (enterprise.models.SystemWideEnterpriseUserRoleAssignment attribute), 242

ARBITRARILY_LARGE_NUMBER (enterprise.admin.BigTableMysqlPaginator attribute), 43

arguments (enterprise.models.BulkCatalogQueryUpdateCommandConfig attribute), 138

assign_admin_role() (in module enterprise.roles_api), 244

assign_feature_roles() (in module enterprise.api.utils), 97

assign_learner_role() (in module enterprise.roles_api), 244

assign_learners() (enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet method), 72

assign_or_delete_enterprise_learner_role() (in module enterprise.signals), 245

AT_ENROLLMENT (enterprise.models.EnterpriseCustomer attribute), 148

AUDIT (enterprise.constants.CourseModes attribute), 129

audit_reporting_disabled (enterprise.models.EnterpriseCourseEnrollment property), 144

auth_org_id (enterprise.models.EnterpriseCustomer attribute), 148

auth_org_id (enterprise.models.HistoricalEnterpriseCustomer attribute), 198

auth_user_model (enterprise.apps.EnterpriseConfig attribute), 148

authentication_classes (enterprise.api.v1.views.base_views.EnterpriseViewSet attribute), 55

authentication_classes (enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryViewSet attribute), 57

authentication_classes (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet attribute), 68

authentication_classes (enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingViewSet attribute), 69

authentication_classes (enterprise.api.v1.views.notifications.NotificationReadView attribute), 77

auto_complete_fields (enterprise.admin.EnterpriseCustomerCatalogAdmin attribute), 48

auto_complete_fields (enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin attribute), 49

auto_complete_fields (enterprise.admin.EnterpriseGroupMembershipAdmin attribute), 49

- attribute), 52
- ## B
- backend_service_edx_oauth2_key (enterprise.apps.EnterpriseConfig attribute), 128
- backend_service_edx_oauth2_provider_url (enterprise.apps.EnterpriseConfig attribute), 128
- backend_service_edx_oauth2_secret (enterprise.apps.EnterpriseConfig attribute), 128
- BackendServiceAPIClient (class in enterprise.api_client.client), 99
- backfill_learner_role_assignments() (enterprise.management.commands.backfill_learner_role_assignments.Command method), 117
- Base64EmailCSVField (class in enterprise.api.v1.fields), 80
- base_fields(enterprise.admin.forms.AdminNotificationForm attribute), 26
- base_fields(enterprise.admin.forms.EnterpriseCustomerAdminForm attribute), 27
- base_fields(enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm attribute), 29
- base_fields(enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm attribute), 29
- base_fields(enterprise.admin.forms.EnterpriseCustomerReportingConfigAdminForm attribute), 30
- base_fields(enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentsForm attribute), 31
- base_fields(enterprise.admin.forms.ManageLearnersDataSharingConsentForm attribute), 32
- base_fields(enterprise.admin.forms.ManageLearnersForm attribute), 33
- base_fields(enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentsForm attribute), 34
- base_fields(enterprise.admin.forms.TransmitEnterpriseCourseForm attribute), 34
- base_fields(enterprise.forms.EnterpriseLoginForm attribute), 132
- base_fields(enterprise.forms.EnterpriseSelectionForm attribute), 132
- base_saml_config_fields (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 173
- BaseEnterpriseCustomerInviteKeySerializer (class in enterprise.api.v1.serializers), 81
- BaseEnterpriseCustomerInviteKeySerializer.Meta (class in enterprise.api.v1.serializers), 81
- BaseEnterpriseCustomerView (class in enterprise.admin.views), 37
- basename(enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryViewSet attribute), 57
- basename(enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet attribute), 58
- basename(enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView attribute), 60
- basename(enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsView attribute), 62
- basename(enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsView attribute), 63
- basename(enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationView attribute), 65
- basename(enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView attribute), 65
- basename(enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView attribute), 66
- basename(enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyView attribute), 68
- basename(enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingView attribute), 69
- basename(enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationView attribute), 70
- basename(enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserView attribute), 71
- basename(enterprise.api.v1.views.enterprise_group.EnterpriseGroupView attribute), 72
- basename(enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentView attribute), 74
- basename(enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedSubsidyFulfillmentView attribute), 76
- basename(enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserView attribute), 78
- basename(enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserView attribute), 78
- BaseThrottle (class in enterprise.api.throttles), 96
- basic_list() (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView method), 60
- basic_list() (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyView method), 68
- batch() (in module enterprise.utils), 249
- batch_size(enterprise.config.models.UpdateRoleAssignmentsWithCustomBatchSizeView attribute), 113
- BigTableMySQLPaginator (class in enterprise.admin), 43
- blackboardenterpriseconfiguration_set (enterprise.models.EnterpriseCustomer attribute), 149
- BODY_HELP_TEXT (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 140
- BOTH_COURSE_FIELDS_SPECIFIED (enterprise.utils.ValidationMessages attribute), 249
- BOTH_FIELDS_SPECIFIED (enterprise.utils.ValidationMessages attribute), 249
- branding_configuration (enterprise.models.EnterpriseCustomer attribute),

149
 BrazeAPIClient (class in *enterprise.api_client.braze*), 98
 build_records() (enter-
 prise.management.commands.manufacture_data.Node attribute), 100
 method), 121
 build_tree_from_field_list() (in module *enter-
 prise.management.commands.manufacture_data*), 121
 bulk_licensed_enrollments_expiration() (enter-
 prise.api.v1.views.enterprise_subsidy_fulfillment), 76
 method), 76
 BULK_LINK_FAILED (enter-
 prise.utils.ValidationMessages attribute), 249
 BULK_UPLOAD (enter-
 prise.admin.forms.ManageLearnersForm.Fields attribute), 32
 BulkCatalogQueryUpdateCommandConfiguration
 (class in *enterprise.models*), 138
 BulkCatalogQueryUpdateCommandConfiguration.DoesNotExist, 138
 BulkCatalogQueryUpdateCommandConfiguration.MultipleObjectsReturned, 138
 BulkCatalogQueryUpdateCommandConfigurationAdmin, 26
 (class in *enterprise.admin.forms*), 26
 BY_EMAIL (enter-
 prise.admin.forms.ManageLearnersForm.NotificationTypes attribute), 33

C
 camelCase() (in module *enterprise.utils*), 250
 can_delete (enter-
 prise.admin.EnterpriseCustomerBrandingConfigurationInline attribute), 47
 can_delete (enter-
 prise.admin.EnterpriseCustomerCatalogInline attribute), 48
 cancel_enrollment() (enter-
 prise.api.v1.views.enterprise_subsidy_fulfillment), 75
 method), 75
 canvasenterpriseconfiguration_set
 (enter-
 prise.models.EnterpriseCustomer attribute), 149
 career_engagement_network_message (enter-
 prise.models.EnterpriseCustomer attribute), 149
 career_engagement_network_message (enter-
 prise.models.HistoricalEnterpriseCustomer attribute), 198
 catalog_admin_role() (in module *enter-
 prise.roles_api*), 244
 catalog_contains_course() (enter-
 prise.models.EnterpriseCustomer method), 149
 CATALOG_DIFF_ENDPOINT (enter-
 prise.api_client.enterprise_catalog.EnterpriseCatalogApiClient attribute), 104
 CatalogQueryPreviewView (class in *enter-
 prise.admin.views*), 37
 CATALOGS_COURSES_ENDPOINT (enter-
 prise.api_client.discovery.CourseCatalogApiClient attribute), 100
 CertificatesApiClient (class in *enter-
 prise.api_client.lms*), 106
 change_actions (enter-
 prise.admin.EnrollmentNotificationEmailTemplateAdmin attribute), 43
 change_actions (enter-
 prise.admin.EnterpriseCustomerAdmin attribute), 45
 change_actions (enter-
 prise.admin.EnterpriseCustomerSsoConfigurationAdmin attribute), 50
 change_date (enter-
 prise.config.models.UpdateRoleAssignmentsWithCustom attribute), 113
 change_date (enter-
 prise.models.BulkCatalogQueryUpdateCommandConf attribute), 138
 change_list_template (enter-
 prise.admin.EnterpriseCourseEnrollmentAdmin attribute), 44
 change_view() (enter-
 prise.admin.EnterpriseCustomerAdmin attribute), 45
 changed_by (enter-
 prise.config.models.UpdateRoleAssignmentsWithCustom attribute), 113
 changed_by (enter-
 prise.models.BulkCatalogQueryUpdateCommandConf attribute), 138
 changed_by_id (enter-
 prise.config.models.UpdateRoleAssignmentsWithCustom attribute), 113
 changed_by_id (enter-
 prise.models.BulkCatalogQueryUpdateCommandConfiguration attribute), 138
 changelist_view() (enter-
 prise.admin.EnterpriseCourseEnrollmentAdmin method), 44
 chat_completion() (in module *enter-
 prise.api_client.open_ai*), 112
 chat_gpt_prompts (enter-
 prise.models.EnterpriseCustomer attribute), 149
 ChatGPTResponse (class in *enterprise.models*), 138
 ChatGPTResponse.DoesNotExist, 139
 ChatGPTResponse.MultipleObjectsReturned, 139
 ChatGPTResponseAdmin (class in *enterprise.admin*), 43
 check_discovery() (in module *enter-
 prise.heartbeat.checks*), 114
 check_ecommerce() (in module *enter-
 prise.heartbeat.checks*), 114
 check_enterprise_catalog() (in module *enter-
 prise.heartbeat.checks*), 114

<code>check_lms()</code> (in module <code>enterprise.heartbeat.checks</code>), 115	<code>CodesAPIRequestError</code> , 131
<code>check_user_part_of_customer()</code> (in module <code>enterprise.api.v1.views.enterprise_customer_sso_configuration</code>), 71	<code>cohort_name</code> (<code>enterprise.models.HistoricalPendingEnrollment</code> attribute), 225
<code>CHOICES</code> (<code>enterprise.constants.FulfillmentTypes</code> attribute), 129	<code>cohort_name</code> (<code>enterprise.models.PendingEnrollment</code> attribute), 236
<code>clean()</code> (<code>enterprise.admin.forms.AdminNotificationForm</code> method), 26	<code>Command</code> (class in <code>enterprise.management.commands.add_executed_exclusion_to_catalog</code>), 117
<code>clean()</code> (<code>enterprise.admin.forms.EnterpriseCustomerIdentityProviderForm</code> method), 30	<code>Command</code> (class in <code>enterprise.management.commands.backfill_learner_role_assignments</code>), 117
<code>clean()</code> (<code>enterprise.admin.forms.EnterpriseCustomerReportingConfigurationForm</code> method), 31	<code>Command</code> (class in <code>enterprise.management.commands.bulk_update_catalog_query_id</code>), 118
<code>clean()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.compare_discovery_and_enterprise</code>), 118
<code>clean()</code> (<code>enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentForm</code> method), 34	<code>Command</code> (class in <code>enterprise.management.commands.create_enterprise_course_enrollment</code>), 118
<code>clean()</code> (<code>enterprise.forms.EnterpriseLoginForm</code> method), 132	<code>Command</code> (class in <code>enterprise.management.commands.create_missing_dsc_records</code>), 119
<code>clean()</code> (<code>enterprise.forms.EnterpriseSelectionForm</code> method), 132	<code>Command</code> (class in <code>enterprise.management.commands.create_missing_dsc_records</code>), 119
<code>clean()</code> (<code>enterprise.models.EnterpriseCustomerReportingConfiguration</code> method), 170	<code>Command</code> (class in <code>enterprise.management.commands.email_drip_for_missing_dsc_records</code>), 119
<code>clean()</code> (<code>enterprise.models.EnterpriseGroupMembership</code> method), 189	<code>Command</code> (class in <code>enterprise.management.commands.ensure_singular_active_enterprise</code>), 120
<code>clean_channel_worker_username()</code> (<code>enterprise.admin.forms.TransmitEnterpriseCoursesForm</code> method), 34	<code>Command</code> (class in <code>enterprise.management.commands.ensure_singular_active_enterprise</code>), 120
<code>clean_course()</code> (<code>enterprise.admin.forms.ManageLearnersDataSharingForm</code> method), 32	<code>Command</code> (class in <code>enterprise.management.commands.fix_dsc_records</code>), 120
<code>clean_course()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.manufacture_data</code>), 120
<code>clean_discount()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.migrate_enterprise_catalogs</code>), 122
<code>clean_email_or_username()</code> (<code>enterprise.admin.forms.ManageLearnersDataSharingForm</code> method), 32	<code>Command</code> (class in <code>enterprise.management.commands.monthly_impact_report</code>), 122
<code>clean_email_or_username()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.nudge_dormant_enrolled_enterpris</code>), 123
<code>clean_html_for_template_rendering()</code> (in module <code>enterprise.utils</code>), 250	<code>Command</code> (class in <code>enterprise.management.commands.reencrypt_enterprise_customer_rep</code>), 123
<code>clean_notify()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.remove_expired_pending_group_m</code>), 123
<code>clean_reason()</code> (<code>enterprise.admin.forms.ManageLearnersForm</code> method), 33	<code>Command</code> (class in <code>enterprise.management.commands.revert_enrollment_objects</code>), 124
<code>clear_pending_registration()</code> (<code>enterprise.models.EnterpriseCustomer</code> method), 150	<code>Command</code> (class in <code>enterprise</code>), 124

[prise.management.commands.save_enterprise_customer_content_filter](#) (enterprise-
[124](#) [prise.models.HistoricalEnterpriseCustomerCatalog](#)
 Command (class in enterprise- [attribute](#)), 203
[prise.management.commands.seed_enterprise_dev_content_filter_ids](#) (enterprise-
[125](#) [prise.models.EnterpriseCustomerCatalog](#)
 Command (class in enterprise- [attribute](#)), 163
[prise.management.commands.unlink_enterprise_content_metadata_identifier_endpoint](#) (enterprise-
[125](#) [prise.api_client.enterprise_catalog.EnterpriseCatalogApiClient](#)
 Command (class in enterprise- [attribute](#)), 104
[prise.management.commands.update_config_last_content_metadata_item_transmission_set](#) (enterprise-
[126](#) [prise.models.EnterpriseCustomer](#) [attribute](#)),
 Command (class in enterprise- [150](#)
[prise.management.commands.update_role_assignment_to_unusable\(\)](#) (in module enterprise-
[126](#) [prise.management.commands.manufacture_data](#)),
[configure_edx_oauth\(\)](#) (enterprise- [121](#)
[prise.api_client.sso_orchestrator.EnterpriseSSOOrchestrator.update_auth_state\(\)](#) (in module enterprise.utils), 250
[method](#)), 112 [cornerstoneenterpriseconfiguration_set](#)
[configure_sso_orchestration_record\(\)](#) (enterprise- [\(enterprise.models.EnterpriseCustomer](#) [at-](#)
[prise.api_client.sso_orchestrator.EnterpriseSSOOrchestrator.update_auth_state\(\)](#) [150](#)
[method](#)), 112 [count](#) (enterprise.admin.BigTableMysqlPaginator prop-
[configured_at](#) (enterprise- [erty](#)), 43
[prise.models.EnterpriseCustomerSsoConfiguration](#) [count](#) (enterprise.api.v1.views.enterprise_course_enrollment.PaginatorWith
[attribute](#)), 174 [attribute](#)), 59
[configured_at](#) (enterprise- [count\(\)](#) (enterprise.admin.EnterpriseGroupAdmin
[prise.models.HistoricalEnterpriseCustomerSsoConfiguration](#) [method](#)), 52
[attribute](#)), 209 [country](#) (enterprise.models.EnterpriseCustomer [at-](#)
[connect\(\)](#) (enterprise.api_client.client.UserAPIClient [tribute](#)), 150
[method](#)), 99 [country](#) (enterprise.models.HistoricalEnterpriseCustomer
[contact_email](#) (enterprise.models.EnterpriseCustomer [attribute](#)), 199
[attribute](#)), 150 [country_attribute](#) (enterprise-
[contact_email](#) (enterprise- [prise.models.EnterpriseCustomerSsoConfiguration](#)
[prise.models.HistoricalEnterpriseCustomer](#) [attribute](#)), 174
[attribute](#)), 198 [country_attribute](#) (enterprise-
[contains_content_items\(\)](#) (enterprise- [prise.models.HistoricalEnterpriseCustomerSsoConfiguration](#)
[prise.api.v1.views.enterprise_customer.EnterpriseCustomerWithSet](#) [attribute](#)), 209
[method](#)), 60 [COUPON_CODE](#) (enterprise.constants.FulfillmentTypes [at-](#)
[contains_content_items\(\)](#) (enterprise- [tribute](#)), 129
[prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView](#) [in](#) enterprise-
[method](#)), 65 [prise.api.v1.views.coupon_codes](#)), 56
[contains_content_items\(\)](#) (enterprise- [COURSE](#) (enterprise.admin.forms.ManageLearnersDataSharingConsentForm
[prise.api_client.enterprise_catalog.EnterpriseCatalogApiClient](#) [attribute](#)), 32
[method](#)), 104 [COURSE](#) (enterprise.admin.forms.ManageLearnersForm.Fields
[contains_courses\(\)](#) (enterprise- [attribute](#)), 32
[prise.models.EnterpriseCustomerCatalog](#) [COURSE_COMPLETED](#) (enterprise-
[method](#)), 162 [prise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterprise](#)
[contains_programs\(\)](#) (enterprise- [attribute](#)), 75
[prise.models.EnterpriseCustomerCatalog](#) [course_detail\(\)](#) (enterprise-
[method](#)), 162 [prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomer](#)
[content_filter](#) (enterprise- [method](#)), 65
[prise.models.EnterpriseCatalogQuery](#) [attribute](#)), 142 [course_enrollment](#) (enterprise-
[content_filter](#) (enterprise- [prise.models.EnterpriseCourseEnrollment](#)
[prise.models.EnterpriseCustomerCatalog](#) [attribute](#)), 144
[attribute](#)), 162 [COURSE_ENROLLMENT_VIEW_URL](#) (enterprise-
[prise.views.RouterView](#) [attribute](#)), 271

`course_enrollments()` (enterprise.api.v1.serializers), 81
`course_enrollments()` (enterprise.api.v1.views.enterprise_customer.EnterpriseCourseEnrollmentDowngradeError, 248
method), 60
`course_entitlement` (enterprise.CourseEnrollmentPermissionError, 248
`enterprise.models.EnterpriseCourseEntitlement`
attribute), 146
`COURSE_ID` (enterprise.admin.forms.ManageLearnersForm.CsvColumn, 129
attribute), 32
`course_id` (enterprise.models.EnterpriseCourseEnrollment, 81
attribute), 144
`course_id` (enterprise.models.HistoricalEnterpriseCourseEnrollment, 81
attribute), 193
`course_id` (enterprise.models.HistoricalPendingEnrollment, 81
attribute), 225
`course_id` (enterprise.models.PendingEnrollment, 81
attribute), 236
`course_modal()` (in module enterprise, 127
`enterprise.template_tags.enterprise`), 127
`COURSE_MODE` (enterprise.admin.forms.ManageLearnersForm.CsvColumn, 129
attribute), 32
`course_mode` (enterprise.models.HistoricalPendingEnrollment, 81
attribute), 225
`course_mode` (enterprise.models.PendingEnrollment, 81
attribute), 237
`COURSE_MODE_INVALID_FOR_COURSE` (enterprise, 249
attribute), 249
`COURSE_NOT_EXIST_IN_CATALOG` (enterprise, 249
attribute), 249
`course_or_program_exist()` (enterprise, 269
`enterprise.views.GrantDataSharingPermissions`
method), 269
`course_run_detail()` (enterprise, 66
`enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCourseRunDetail`
method), 66
`COURSE_RUNS_ENDPOINT` (enterprise, 100
`enterprise.api_client.discovery.CourseCatalogApiClient`
attribute), 100
`course_uuid` (enterprise.models.EnterpriseCourseEntitlement, 146
attribute), 146
`course_uuid` (enterprise.models.HistoricalEnterpriseCourseEntitlement, 195
attribute), 195
`COURSE_WITHOUT_COURSE_MODE` (enterprise, 249
attribute), 249
`CourseApiClient` (class in enterprise.api_client.lms), 107
`CourseCatalogApiClient` (class in enterprise, 100
`enterprise.api_client.discovery`), 100
`CourseCatalogApiError`, 248
`CourseCatalogApiServiceClient` (class in enterprise, 102
`enterprise.api_client.discovery`), 102
`CourseDetailSerializer` (class in enterprise, 81
`enterprise.api.v1.serializers`), 81

prise.api.utils), 97

create_order_for_enrollment() (enterprise.models.EnterpriseCustomerUser method), 178

create_pending_enterprise_admin_user() (in module enterprise.signals), 245

create_recipient() (enterprise.api_client.braze.BrazeAPIClient method), 98

create_recipient_no_external_id() (enterprise.api_client.braze.BrazeAPIClient method), 98

created (enterprise.models.AdminNotification attribute), 135

created (enterprise.models.AdminNotificationFilter attribute), 136

created (enterprise.models.AdminNotificationRead attribute), 137

created (enterprise.models.ChatGPTResponse attribute), 139

created (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 141

created (enterprise.models.EnterpriseCatalogQuery attribute), 142

created (enterprise.models.EnterpriseCourseEnrollment attribute), 144

created (enterprise.models.EnterpriseCourseEntitlement attribute), 147

created (enterprise.models.EnterpriseCustomer attribute), 150

created (enterprise.models.EnterpriseCustomerBrandingConfiguration attribute), 161

created (enterprise.models.EnterpriseCustomerCatalog attribute), 163

created (enterprise.models.EnterpriseCustomerIdentityProvider attribute), 166

created (enterprise.models.EnterpriseCustomerInviteKey attribute), 167

created (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170

created (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 174

created (enterprise.models.EnterpriseCustomerType attribute), 177

created (enterprise.models.EnterpriseCustomerUser attribute), 178

created (enterprise.models.EnterpriseEnrollmentSource attribute), 182

created (enterprise.models.EnterpriseFeatureRole attribute), 183

created (enterprise.models.EnterpriseFeatureUserRoleAssignment attribute), 184

created (enterprise.models.EnterpriseFulfillmentSource attribute), 186

created (enterprise.models.EnterpriseGroup attribute), 187

created (enterprise.models.EnterpriseGroupMembership attribute), 189

created (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 191

created (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 193

created (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196

created (enterprise.models.HistoricalEnterpriseCustomer attribute), 199

created (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 204

created (enterprise.models.HistoricalEnterpriseCustomerInviteKey attribute), 206

created (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 209

created (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 212

created (enterprise.models.HistoricalEnterpriseGroup attribute), 215

created (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 217

created (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 220

created (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 222

created (enterprise.models.HistoricalPendingEnrollment attribute), 225

created (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser attribute), 227

created (enterprise.models.HistoricalPendingEnterpriseCustomerUser attribute), 229

created (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 231

created (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234

created (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 235

created (enterprise.models.PendingEnrollment attribute), 237

created (enterprise.models.PendingEnterpriseCustomerAdminUser attribute), 238

created (enterprise.models.PendingEnterpriseCustomerUser attribute), 239

created (enterprise.models.SystemWideEnterpriseRole attribute), 241

created (enterprise.models.SystemWideEnterpriseUserRoleAssignment attribute), 242

CREDIT (enterprise.constants.CourseModes attribute), 129

CUSTOMER_ADMIN (enterprise.models.EnterpriseEnrollmentSource

attribute), 182
 customer_admin_enroll_user() (in module *enterprise.utils*), 250
 customer_admin_enroll_user_with_status() (in module *enterprise.utils*), 250
 customer_success_email (enterprise.apps.EnterpriseConfig attribute), 128
 customer_type (enterprise.models.EnterpriseCustomer attribute), 151
 customer_type (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
 customer_type_id (enterprise.models.EnterpriseCustomer attribute), 151
 customer_type_id (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
 CustomPaginator (class in *enterprise.admin.paginator*), 35

D

dashboard_list() (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet method), 60
 DATA_SHARING_CONSENT_CHOICES (enterprise.models.EnterpriseCustomer attribute), 148
 data_sharing_consent_page (enterprise.models.EnterpriseCustomer attribute), 151
 data_sharing_consent_records (enterprise.models.EnterpriseCustomerUser property), 178
 data_type (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
 DATA_TYPE_CATALOG (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_CHOICES (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_COMPLETION (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_COURSE_STRUCTURE (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_ENGAGEMENT (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_GRADE (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 DATA_TYPE_PROGRESS_V3 (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 169
 day_of_month (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
 day_of_week (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
 DAYS_OF_WEEK (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
 declared_fields (enterprise.admin.forms.AdminNotificationForm attribute), 26
 declared_fields (enterprise.admin.forms.EnterpriseCustomerAdminForm attribute), 28
 declared_fields (enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm attribute), 29
 declared_fields (enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm attribute), 30
 declared_fields (enterprise.admin.forms.EnterpriseCustomerReportingConfigAdminForm attribute), 31
 declared_fields (enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentForm attribute), 31
 declared_fields (enterprise.admin.forms.ManageLearnersDataSharingConsentForm attribute), 32
 declared_fields (enterprise.admin.forms.ManageLearnersForm attribute), 34
 declared_fields (enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentForm attribute), 34
 declared_fields (enterprise.admin.forms.TransmitEnterpriseCoursesForm attribute), 35
 declared_fields (enterprise.forms.EnterpriseLoginForm attribute), 132
 declared_fields (enterprise.forms.EnterpriseSelectionForm attribute), 132
 decrypted_password (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
 decrypted_sftp_password (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 171
 DEFAULT (enterprise.admin.forms.ManageLearnersForm.NotificationTypes attribute), 33
 default_content_filter() (in module *enterprise*

- prise.signals*), 245
- `default_contract_discount` (enterprise.models.EnterpriseCustomer attribute), 151
- `default_contract_discount` (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- `default_language` (enterprise.models.EnterpriseCustomer attribute), 151
- `default_language` (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- `default_provider` (enterprise.models.EnterpriseCustomerIdentityProvider attribute), 166
- `default_provider_idp` (enterprise.models.EnterpriseCustomer property), 151
- `DEFAULT_VALUE_SAFEGUARD` (enterprise.api_client.discovery.CourseCatalogApiClient attribute), 100
- `DEFAULT_VALUE_SAFEGUARD` (enterprise.api_client.enterprise.EnterpriseApiClient attribute), 103
- `DefaultColors` (class in *enterprise.constants*), 129
- `degreed2enterprisecustomerconfiguration_set` (enterprise.models.EnterpriseCustomer attribute), 151
- `degreedenterprisecustomerconfiguration_set` (enterprise.models.EnterpriseCustomer attribute), 151
- `delete()` (enterprise.admin.views.EnterpriseCustomerManageLearnersView method), 38
- `delete()` (enterprise.models.EnterpriseCatalogQuery method), 142
- `delete_admin_role_assignment()` (in module *enterprise.roles_api*), 244
- `delete_data_sharing_consent()` (in module *enterprise.utils*), 251
- `delete_enterprise_admin_role_assignment()` (in module *enterprise.signals*), 245
- `delete_enterprise_catalog()` (enterprise.api_client.enterprise_catalog.EnterpriseCatalogClient method), 104
- `delete_enterprise_catalog_data()` (in module *enterprise.signals*), 245
- `delete_enterprise_learner_role_assignment()` (in module *enterprise.signals*), 245
- `delete_learner_role_assignment()` (in module *enterprise.roles_api*), 244
- `delete_pending_enterprise_admin_user()` (in module *enterprise.signals*), 245
- `delete_role_assignment()` (in module *enterprise.roles_api*), 244
- `delivery_method` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 171
- `DELIVERY_METHOD_CHOICES` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
- `DELIVERY_METHOD_EMAIL` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
- `DELIVERY_METHOD_SFTP` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
- `delta_format()` (in module *enterprise.api.utils*), 97
- `deprecated()` (in module *enterprise.decorators*), 130
- `description` (enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQuery attribute), 57
- `description` (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollment attribute), 58
- `description` (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomer attribute), 60
- `description` (enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentials attribute), 62
- `description` (enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentials attribute), 63
- `description` (enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfiguration attribute), 65
- `description` (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalog attribute), 66
- `description` (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalog attribute), 67
- `description` (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKey attribute), 68
- `description` (enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReporting attribute), 69
- `description` (enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfiguration attribute), 70
- `description` (enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUser attribute), 71
- `description` (enterprise.api.v1.views.enterprise_group.EnterpriseGroupView attribute), 72
- `description` (enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillment attribute), 75
- `description` (enterprise.api.v1.views.enterprise_subsidy_fulfillment.License attribute), 76
- `description` (enterprise.api.v1.views.pending_enterprise_customer_user.EnterprisePendingCustomerUser attribute), 78
- `description` (enterprise.api.v1.views.pending_enterprise_customer_user.EnterprisePendingCustomerUser attribute), 78
- `description` (enterprise.models.EnterpriseFeatureRole attribute), 183
- `description` (enterprise.models.SystemWideEnterpriseRole attribute), 241
- `destroy()` (enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentials attribute), 76

- method), 63
- destroy() (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet method), 68
- destroy() (enterprise.api.v1.views.enterprise_customer_report.EnterpriseCustomerReportingConfigurationViewSet method), 69
- destroy() (enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationViewSet method), 70
- detail(enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryAdminView method), 57
- detail(enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentView method), 58
- detail(enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView method), 60
- detail(enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentialsView method), 62
- detail(enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentialsView method), 64
- detail(enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationView method), 65
- detail(enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView method), 66
- detail(enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogWriteView method), 67
- detail(enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyView method), 68
- detail(enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingConfigurationView method), 69
- detail(enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationView method), 70
- detail(enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserView method), 71
- detail(enterprise.api.v1.views.enterprise_group.EnterpriseGroupView method), 72
- detail(enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentView method), 75
- detail(enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseCourseEnrollmentView method), 76
- detail(enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserView method), 78
- detail(enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserView method), 78
- disable_for_loaddata() (in module enterprise.decorators), 130
- DISCOUNT (enterprise.admin.forms.ManageLearnersForm.Fields attribute), 32
- discount_percentage (enterprise.models.HistoricalPendingEnrollment attribute), 225
- discount_percentage (enterprise.models.PendingEnrollment attribute), 237
- discovery_query_url() (enterprise.admin.EnterpriseCatalogQueryAdmin method), 44
- discovery_query_url() (in module enterprise.utils), 51
- EnterpriseCustomerInviteKeyViewSet, 68
- DiscoveryNotAvailable, 115
- display() (enterprise.api.v1.views.EnterpriseCatalogQueryAdminView method), 268
- dispatch() (in module enterprise.api_client.lms), 270
- display_game_query_views() (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 174
- display() (in module enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 209
- django_pagination_class (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentView method), 115
- APIClient (class in enterprise.api_client.lms), 103
- dry_run (enterprise.config.models.UpdateRoleAssignmentsWithCustomers method), 115
- APIClient (class in enterprise.api_client.lms), 103
- EnterpriseCustomerBrandingConfigurationViewSet, 65
- EcommerceApiClient (class in enterprise.api_client.lms), 103
- EcommerceNotAvailable, 115
- EnterpriseCustomerCatalogWriteView method, 67
- EnterpriseCustomerReportingConfigurationView method, 69
- EnterpriseCustomerSsoConfigurationView method, 70
- EnterpriseCustomerUserView method, 71
- email_attribute (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 209
- EnterpriseSubsidyFulfillmentView method, 75
- EnterpriseCourseEnrollmentView method, 76
- EMAIL_OR_USERNAME (enterprise.admin.forms.ManageLearnersForm.Fields attribute), 32
- PendingEnterpriseCustomerUserView method, 78
- PendingEnterpriseCustomerUserView method, 78
- email_or_username_to_email() (in module enterprise.admin.utils), 35
- EmbargoApiClient (class in enterprise.api_client.lms), 107
- emit_event() (enterprise.management.commands.email_drip_for_missing method), 119
- emit_event() (enterprise.management.commands.monthly_impact_report method), 122
- emit_event() (enterprise.management.commands.nudge_dormant_enrollment method), 123
- enable_academies (enterprise.models.EnterpriseCustomer attribute), 152
- enable_academies (enterprise.models.HistoricalEnterpriseCustomer attribute), 209

- attribute), 199
- enable_analytics_screen (enterprise.models.EnterpriseCustomer attribute), 152
- enable_analytics_screen (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- enable_audit_data_reporting (enterprise.models.EnterpriseCustomer attribute), 152
- enable_audit_data_reporting (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- enable_audit_enrollment (enterprise.models.EnterpriseCustomer attribute), 152
- enable_audit_enrollment (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- enable_autocohorting (enterprise.models.EnterpriseCustomer attribute), 152
- enable_autocohorting (enterprise.models.HistoricalEnterpriseCustomer attribute), 199
- enable_browse_and_request (enterprise.models.EnterpriseCustomer attribute), 152
- enable_browse_and_request (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_career_engagement_network_on_learner_portal (enterprise.models.EnterpriseCustomer attribute), 152
- enable_career_engagement_network_on_learner_portal (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_compression (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 171
- enable_data_sharing_consent (enterprise.models.EnterpriseCustomer attribute), 152
- enable_data_sharing_consent (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_demo_data_for_analytics_and_lpr (enterprise.models.EnterpriseCustomer attribute), 152
- enable_demo_data_for_analytics_and_lpr (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_dsc() (enterprise.admin.EnterpriseCustomerAdmin method), 45
- enable_executive_education_2U_fulfillment (enterprise.models.EnterpriseCustomer attribute), 152
- enable_executive_education_2U_fulfillment (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_generation_of_api_credentials (enterprise.models.EnterpriseCustomer attribute), 152
- enable_generation_of_api_credentials (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_integrated_customer_learner_portal_search (enterprise.models.EnterpriseCustomer attribute), 152
- enable_integrated_customer_learner_portal_search (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_learner_portal (enterprise.models.EnterpriseCustomer attribute), 152
- enable_learner_portal (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_learner_portal_offers (enterprise.models.EnterpriseCustomer attribute), 152
- enable_learner_portal_offers (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_one_academy (enterprise.models.EnterpriseCustomer attribute), 153
- enable_one_academy (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_pathways (enterprise.models.EnterpriseCustomer attribute), 153
- enable_pathways (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_portal_code_management_screen (enterprise.models.EnterpriseCustomer attribute), 153
- enable_portal_code_management_screen (enterprise.models.HistoricalEnterpriseCustomer attribute), 200
- enable_portal_learner_credit_management_screen (enterprise.models.EnterpriseCustomer attribute), 153
- enable_portal_learner_credit_management_screen (enterprise.models.HistoricalEnterpriseCustomer attribute), 200

`enable_portal_lms_configurations_screen` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_portal_lms_configurations_screen` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 200
`enable_portal_reporting_config_screen` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_portal_reporting_config_screen` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enable_portal_saml_configuration_screen` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_portal_saml_configuration_screen` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enable_portal_subscription_management_screen` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_portal_subscription_management_screen` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enable_programs` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_programs` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enable_slug_login` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_slug_login` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enable_universal_link` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enable_universal_link` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enabled` (*enterprise.config.models.UpdateRoleAssignmentsWithCustomPermissionsConfig* attribute), 113
`enabled` (*enterprise.models.BulkCatalogQueryUpdateCommand* attribute), 138
`enabled_course_modes` (*enterprise.models.EnterpriseCustomerCatalog* attribute), 163
`enabled_course_modes` (*enterprise.models.HistoricalEnterpriseCustomerCatalog* attribute), 204
`enables_audit_data_reporting` (*enterprise.models.EnterpriseCustomer* property), 153
`encrypted_password` (*enterprise.models.EnterpriseCustomerReportingConfiguration* property), 171
`encrypted_sftp_password` (*enterprise.models.EnterpriseCustomerReportingConfiguration* property), 171
`enforce_data_sharing_consent` (*enterprise.models.EnterpriseCustomer* attribute), 153
`enforce_data_sharing_consent` (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 201
`enforces_data_sharing_consent()` (*enterprise.models.EnterpriseCustomer* method), 153
`enroll()` (*enterprise.models.EnterpriseCustomerUser* method), 178
`enroll_learners_in_courses()` (*enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView* method), 60
`enroll_subsidy_users_in_courses()` (in module *enterprise.utils*), 251
`enroll_user()` (in module *enterprise.utils*), 252
`enroll_user_in_course()` (*enterprise.api_client.lms.EnrollmentApiClient* method), 107
`enroll_user_pending_registration()` (*enterprise.models.EnterpriseCustomer* method), 154
`enroll_user_pending_registration_with_status()` (*enterprise.models.EnterpriseCustomer* method), 154
`enroll_users_in_course()` (in module *enterprise.utils*), 253
`enrollment_client_error` (*enterprise.views.FailedEnrollmentReason* attribute), 268
`ENROLLMENT_TASK` (*enterprise.models.EnterpriseEnrollmentSource* attribute), 182
`ENROLLMENT_URL` (*enterprise.config.models.EnterpriseCustomerManageLearnersView.Config* attribute), 38
`ENROLLMENT_URL` (*enterprise.models.EnterpriseEnrollmentSource* attribute), 182
`EnrollmentApiClient` (class in *enterprise.api_client.lms*), 107
`EnrollmentModificationException`, 74, 131
`EnrollmentNotificationEmailTemplate` (class in *enterprise.models*), 140
`EnrollmentNotificationEmailTemplate.DoesNotExist`, 141

EnrollmentNotificationEmailTemplate.MultipleObjectsReturned (class in *enterprise.api*), 141
 EnrollmentNotificationEmailTemplateAdmin (class in *enterprise.admin*), 43
 EnrollmentNotificationEmailTemplateAdmin.Meta (class in *enterprise.admin*), 43
 enrollments_for_user() (enter-
 prise.models.EnterpriseFulfillmentSource
 class method), 186
 EnrollmentsInfoSerializer (class in enter-
 prise.api.v1.serializers), 82
 ensure_data_exists() (enter-
 prise.api.v1.views.base_views.EnterpriseViewSet
 method), 56
 enterprise
 module, 273
 enterprise.admin
 module, 42
 enterprise.admin.actions
 module, 25
 enterprise.admin.forms
 module, 25
 enterprise.admin.paginator
 module, 35
 enterprise.admin.utils
 module, 35
 enterprise.admin.views
 module, 37
 enterprise.admin.widgets
 module, 42
 enterprise.api
 module, 98
 enterprise.api.filters
 module, 94
 enterprise.api.pagination
 module, 95
 enterprise.api.throttles
 module, 96
 enterprise.api.urls
 module, 97
 enterprise.api.utils
 module, 97
 enterprise.api.v1
 module, 94
 enterprise.api.v1.decorators
 module, 79
 enterprise.api.v1.fields
 module, 80
 enterprise.api.v1.permissions
 module, 80
 enterprise.api.v1.serializers
 module, 80
 enterprise.api.v1.urls
 module, 94
 enterprise.api.v1.views
 module, 79
 enterprise.api.v1.views.analytics_summary
 module, 55
 enterprise.api.v1.views.base_views
 module, 55
 enterprise.api.v1.views.coupon_codes
 module, 56
 enterprise.api.v1.views.enterprise_catalog_query
 module, 57
 enterprise.api.v1.views.enterprise_course_enrollment
 module, 58
 enterprise.api.v1.views.enterprise_customer
 module, 60
 enterprise.api.v1.views.enterprise_customer_api_credentials
 module, 62
 enterprise.api.v1.views.enterprise_customer_branding_config
 module, 64
 enterprise.api.v1.views.enterprise_customer_catalog
 module, 65
 enterprise.api.v1.views.enterprise_customer_invite_key
 module, 68
 enterprise.api.v1.views.enterprise_customer_reporting
 module, 69
 enterprise.api.v1.views.enterprise_customer_sso_configuration
 module, 70
 enterprise.api.v1.views.enterprise_customer_user
 module, 71
 enterprise.api.v1.views.enterprise_group
 module, 72
 enterprise.api.v1.views.enterprise_subsidy_fulfillment
 module, 74
 enterprise.api.v1.views.notifications
 module, 77
 enterprise.api.v1.views.pending_enterprise_customer_user
 module, 78
 enterprise.api.v1.views.plotly_auth
 module, 79
 enterprise.api_client
 module, 113
 enterprise.api_client.braze
 module, 98
 enterprise.api_client.client
 module, 99
 enterprise.api_client.discovery
 module, 100
 enterprise.api_client.ecommerce
 module, 103
 enterprise.api_client.enterprise
 module, 103
 enterprise.api_client.enterprise_catalog
 module, 104
 enterprise.api_client.lms
 module, 106

<code>enterprise.api_client.open_ai</code>	<code>enterprise.management.commands.manufacture_data</code>
<code>module, 112</code>	<code>module, 120</code>
<code>enterprise.api_client.sso_orchestrator</code>	<code>enterprise.management.commands.migrate_enterprise_catalogs</code>
<code>module, 112</code>	<code>module, 122</code>
<code>enterprise.apps</code>	<code>enterprise.management.commands.monthly_impact_report</code>
<code>module, 128</code>	<code>module, 122</code>
<code>enterprise.config</code>	<code>enterprise.management.commands.nudge_dormant_enrolled_enterprise</code>
<code>module, 114</code>	<code>module, 123</code>
<code>enterprise.config.models</code>	<code>enterprise.management.commands.reencrypt_enterprise_customers</code>
<code>module, 113</code>	<code>module, 123</code>
<code>enterprise.constants</code>	<code>enterprise.management.commands.remove_expired_pending_group_members</code>
<code>module, 129</code>	<code>module, 123</code>
<code>enterprise.decorators</code>	<code>enterprise.management.commands.revert_enrollment_objects</code>
<code>module, 130</code>	<code>module, 124</code>
<code>enterprise.errors</code>	<code>enterprise.management.commands.save_enterprise_customer_user</code>
<code>module, 131</code>	<code>module, 124</code>
<code>enterprise.forms</code>	<code>enterprise.management.commands.seed_enterprise_devstack_data</code>
<code>module, 132</code>	<code>module, 125</code>
<code>enterprise.heartbeat</code>	<code>enterprise.management.commands.unlink_enterprise_customer_user</code>
<code>module, 117</code>	<code>module, 125</code>
<code>enterprise.heartbeat.checks</code>	<code>enterprise.management.commands.update_config_last_errored</code>
<code>module, 114</code>	<code>module, 126</code>
<code>enterprise.heartbeat.exceptions</code>	<code>enterprise.management.commands.update_role_assignments_with_roles</code>
<code>module, 115</code>	<code>module, 126</code>
<code>enterprise.heartbeat.throttles</code>	<code>enterprise.messages</code>
<code>module, 116</code>	<code>module, 133</code>
<code>enterprise.heartbeat.utils</code>	<code>enterprise.middleware</code>
<code>module, 116</code>	<code>module, 134</code>
<code>enterprise.heartbeat.views</code>	<code>enterprise.models</code>
<code>module, 116</code>	<code>module, 134</code>
<code>enterprise.logging</code>	<code>enterprise.roles_api</code>
<code>module, 132</code>	<code>module, 244</code>
<code>enterprise.management</code>	<code>enterprise.rules</code>
<code>module, 127</code>	<code>module, 245</code>
<code>enterprise.management.commands</code>	<code>enterprise.settings</code>
<code>module, 127</code>	<code>module, 127</code>
<code>enterprise.management.commands.add_executed_extensions</code>	<code>enterprise.settings.test</code>
<code>module, 117</code>	<code>module, 127</code>
<code>enterprise.management.commands.backfill_learned_roles</code>	<code>enterprise.settings.test</code>
<code>module, 117</code>	<code>module, 127</code>
<code>enterprise.management.commands.bulk_update_catalogs</code>	<code>enterprise.settings.test</code>
<code>module, 118</code>	<code>module, 127</code>
<code>enterprise.management.commands.compare_discovered_and_enrolled_catalogs</code>	<code>enterprise.settings.test</code>
<code>module, 118</code>	<code>module, 127</code>
<code>enterprise.management.commands.create_enterprise_enrollment_tags</code>	<code>enterprise.settings.test</code>
<code>module, 118</code>	<code>module, 127</code>
<code>enterprise.management.commands.create_missing_enterprise_stoggles</code>	<code>enterprise.settings.test</code>
<code>module, 119</code>	<code>module, 127</code>
<code>enterprise.management.commands.email_drip_for_missing_disconnects</code>	<code>enterprise.settings.test</code>
<code>module, 119</code>	<code>module, 127</code>
<code>enterprise.management.commands.ensure_singular_enterprise_customer_user</code>	<code>enterprise.settings.test</code>
<code>module, 120</code>	<code>module, 127</code>
<code>enterprise.management.commands.fix_dsc_records</code>	<code>enterprise.settings.test</code>
<code>module, 120</code>	<code>module, 127</code>

enterprise.validators	enterprise_course_enrollment_model() (in module enterprise.utils), 253
enterprise.views	enterprise_course_entitlement (enterprise.models.EnterpriseFulfillmentSource attribute), 186
ENTERPRISE_CATALOG_ENDPOINT	enterprise_course_entitlement (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 220
enterprise_catalog_query	enterprise_course_entitlement (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
enterprise_catalog_query	enterprise_course_entitlement (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234
enterprise_catalog_query_id	enterprise_course_entitlement (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 235
enterprise_catalog_query_id	enterprise_course_entitlement_id (enterprise.models.EnterpriseFulfillmentSource attribute), 186
enterprise_contains_content_items()	enterprise_course_entitlement_id (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 220
enterprise_course_enrollment	enterprise_course_entitlement_id (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
enterprise_course_enrollment	enterprise_course_entitlement_id (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234
enterprise_course_enrollment	enterprise_course_entitlement_id (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 235
enterprise_course_enrollment	ENTERPRISE_CUSTOMER (enterprise.admin.views.EnterpriseCustomerManageLearnerDataSharingView attribute), 37
enterprise_course_enrollment	ENTERPRISE_CUSTOMER (enterprise.admin.views.EnterpriseCustomerManageLearnersView.Constructor attribute), 38
enterprise_course_enrollment_id	ENTERPRISE_CUSTOMER (enterprise.admin.views.EnterpriseCustomerTransmitCoursesView.Constructor attribute), 41
enterprise_course_enrollment_id	enterprise_customer (enterprise.models.ChatGPTResponse attribute), 139
enterprise_course_enrollment_id	enterprise_customer (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 141
enterprise_course_enrollment_id	enterprise_customer (enterprise.models.EnterpriseCustomerBrandingConfiguration attribute), 161
enterprise_course_enrollment_id	enterprise_customer (enterprise.models.EnterpriseCustomerCatalog attribute), 163
enterprise_course_enrollment_model() (in module enterprise.tasks), 246	enterprise_customer (enterprise.models.EnterpriseCustomerCatalog attribute), 163

<i>prise.models.EnterpriseCustomerIdentityProvider</i> attribute), 166		<i>prise.models.EnterpriseCatalogQuery</i> attribute), 142	
<i>enterprise_customer</i> <i>prise.models.EnterpriseCustomerInviteKey</i> attribute), 167	(enter-	<i>enterprise_customer_catalogs</i> <i>prise.models.EnterpriseCustomer</i> 154	(enter- attribute),
<i>enterprise_customer</i> <i>prise.models.EnterpriseCustomerReportingConfiguration</i> attribute), 171	(enter-	<i>enterprise_customer_catalogs</i> <i>prise.models.EnterpriseCustomerReportingConfiguration</i> attribute), 171	(enter-
<i>enterprise_customer</i> <i>prise.models.EnterpriseCustomerSsoConfiguration</i> attribute), 174	(enter-	ENTERPRISE_CUSTOMER_CATALOGS_ENDPOINT <i>prise.api_client.enterprise.EnterpriseApiClient</i> attribute), 103	(enter-
<i>enterprise_customer</i> <i>prise.models.EnterpriseCustomerUser</i> attribute), 178	(enter- attribute),	<i>enterprise_customer_consent</i> <i>prise.models.EnterpriseCustomer</i> 154	(enter- attribute),
<i>enterprise_customer</i> <i>prise.models.EnterpriseGroup</i> 187	(enter- attribute),	ENTERPRISE_CUSTOMER_ENDPOINT <i>prise.api_client.enterprise.EnterpriseApiClient</i> attribute), 103	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalEnrollmentNotificationEmailTemplate</i> attribute), 191	(enter-	ENTERPRISE_CUSTOMER_ENDPOINT <i>prise.api_client.enterprise_catalog.EnterpriseCatalogApiClient</i> attribute), 104	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalEnterpriseCustomerCatalog</i> attribute), 204	(enter-	<i>enterprise_customer_id</i> <i>prise.models.ChatGPTResponse</i> 139	(enter- attribute),
<i>enterprise_customer</i> <i>prise.models.HistoricalEnterpriseCustomerInviteKey</i> attribute), 206	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnrollmentNotificationEmailTemplate</i> attribute), 141	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalEnterpriseCustomerSsoConfiguration</i> attribute), 209	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerBrandingConfiguration</i> attribute), 161	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalEnterpriseCustomerUser</i> attribute), 213	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerCatalog</i> attribute), 163	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalEnterpriseGroup</i> attribute), 215	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerIdentityProvider</i> attribute), 167	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalPendingEnterpriseCustomerAdminUser</i> attribute), 227	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerInviteKey</i> attribute), 168	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalPendingEnterpriseCustomerUser</i> attribute), 229	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerReportingConfiguration</i> attribute), 171	(enter-
<i>enterprise_customer</i> <i>prise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</i> attribute), 231	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerSsoConfiguration</i> attribute), 174	(enter-
<i>enterprise_customer</i> <i>prise.models.PendingEnterpriseCustomerAdminUser</i> attribute), 238	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseCustomerUser</i> attribute), 179	(enter- attribute),
<i>enterprise_customer</i> <i>prise.models.PendingEnterpriseCustomerUser</i> attribute), 239	(enter-	<i>enterprise_customer_id</i> <i>prise.models.EnterpriseGroup</i> 188	(enter- attribute),
<i>enterprise_customer</i> <i>prise.models.SystemWideEnterpriseUserRoleAssignment</i> attribute), 242	(enter-	<i>enterprise_customer_id</i> <i>prise.models.HistoricalEnrollmentNotificationEmailTemplate</i> attribute), 191	(enter-
<i>enterprise_customer_catalogs</i>	(enter-	<i>enterprise_customer_id</i>	(enter-

prise.models.HistoricalEnterpriseCustomerCatalog (enterprise_customer_user attribute), 204

prise.models.HistoricalEnterpriseCustomerInviteKey (enterprise_customer_user attribute), 206

prise.models.HistoricalEnterpriseCustomerSsoConnect (enterprise_customer_user attribute), 209

prise.models.HistoricalEnterpriseCustomerUser (enterprise_customer_user attribute), 213

prise.models.HistoricalEnterpriseGroup (enterprise_customer_user attribute), 215

prise.models.HistoricalPendingEnterpriseCustomerAdminUser (enterprise_customer_user_id attribute), 228

prise.models.HistoricalPendingEnterpriseCustomerUser (enterprise_customer_user_id attribute), 229

prise.models.HistoricalSystemWideEnterpriseUserRoleAssignment (enterprise_customer_user_id attribute), 232

prise.models.PendingEnterpriseCustomerAdminUser (enterprise_customer_user_id attribute), 238

prise.models.PendingEnterpriseCustomerUser (enterprise_customer_user_id attribute), 239

prise.models.SystemWideEnterpriseUserRoleAssignment (enterprise_customer_user_id attribute), 242

prise.models.EnterpriseCustomer (enterprise_customer_identity_provider attribute), 155

prise.models.EnterpriseCustomer (enterprise_customer_identity_providers attribute), 155

enterprise_customer_invite_key_model() (in module *enterprise.utils*), 253

enterprise_customer_model() (in module *enterprise.utils*), 253

enterprise_customer_user (enterprise.api.v1.serializers.EnterpriseCourseEnrollmentSerializer attribute), 83

enterprise_customer_user (enterprise.models.AdminNotificationRead attribute), 137

enterprise_customer_user (enterprise.models.EnterpriseCourseEnrollment attribute), 144

enterprise_customer_user (enterprise.models.EnterpriseCourseEntitlement attribute), 147

enterprise_customer_user (enterprise.models.EnterpriseFulfillmentSource property), 186

enterprise_customer_user (enterprise.models.EnterpriseGroupMembership attribute), 189

enterprise_customer_user (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 193

enterprise_customer_user (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196

enterprise_customer_user (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 217

enterprise_customer_user_id (enterprise.models.AdminNotificationRead attribute), 137

enterprise_customer_user_id (enterprise.models.EnterpriseCourseEnrollment attribute), 144

enterprise_customer_user_id (enterprise.models.EnterpriseCourseEntitlement attribute), 147

enterprise_customer_user_id (enterprise.models.EnterpriseGroupMembership attribute), 189

enterprise_customer_user_id (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 193

enterprise_customer_user_id (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196

enterprise_customer_user_id (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 217

enterprise_customer_user_model() (in module *enterprise.tasks*), 246

enterprise_customer_user_model() (in module *enterprise.utils*), 253

enterprise_customer_users (enterprise.models.EnterpriseCustomer attribute), 155

enterprise_customer_uuid (enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfiguration attribute), 113

enterprise_customer_uuids (enterprise.models.EnterpriseFeatureUserRoleAssignment property), 185

enterprise_enrollment_source_model() (in module *enterprise.tasks*), 246

enterprise_enrollment_source_model() (in module *enterprise.utils*), 253

enterprise_enrollment_template (enterprise)

`prise.models.EnterpriseCustomer` attribute), 155

`enterprise_enrollments` (`prise.models.EnterpriseCustomerUser` attribute), 179

`enterprise_enrollments()` (`enterprise.admin.EnterpriseCustomerUserAdmin` method), 51

`enterprise_entitlements` (`prise.models.EnterpriseCustomerUser` attribute), 179

`enterprise_features()` (in module `enterprise.toggles`), 247

`enterprise_group_membership_model()` (in module `enterprise.tasks`), 246

`enterprise_groups_v1()` (in module `enterprise.toggles`), 247

`enterprise_integrations_email` (`enterprise.apps.EnterpriseConfig` attribute), 128

`enterprise_login_required()` (in module `enterprise.decorators`), 130

`enterprise_unenrollment_receiver()` (in module `enterprise.signals`), 245

`EnterpriseApiClient` (class in `enterprise.api_client.enterprise`), 103

`EnterpriseCatalogApiClient` (class in `enterprise.api_client.enterprise_catalog`), 104

`EnterpriseCatalogNotAvailable`, 115

`EnterpriseCatalogQuery` (class in `enterprise.models`), 142

`EnterpriseCatalogQuery.DoesNotExist`, 142

`EnterpriseCatalogQuery.MultipleObjectsReturned`, 142

`EnterpriseCatalogQueryAdmin` (class in `enterprise.admin`), 43

`EnterpriseCatalogQueryAdmin.Meta` (class in `enterprise.admin`), 44

`EnterpriseCatalogQuerySerializer` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCatalogQuerySerializer.Meta` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCatalogQueryViewSet` (class in `enterprise.api.v1.views.enterprise_catalog_query`), 57

`EnterpriseConfig` (class in `enterprise.apps`), 128

`EnterpriseCourseEnrollment` (class in `enterprise.models`), 143

`EnterpriseCourseEnrollment.DoesNotExist`, 143

`EnterpriseCourseEnrollment.MultipleObjectsReturned`, 144

`enterprisecourseenrollment_set` (`enterprise.models.EnterpriseEnrollmentSource` attribute), 182

`EnterpriseCourseEnrollmentAdmin` (class in `enterprise.admin`), 44

`EnterpriseCourseEnrollmentAdmin.Meta` (class in `enterprise.admin`), 44

`EnterpriseCourseEnrollmentFilterBackend` (class in `enterprise.api.filters`), 94

`EnterpriseCourseEnrollmentManager` (class in `enterprise.models`), 146

`EnterpriseCourseEnrollmentPagination` (class in `enterprise.api.v1.views.enterprise_course_enrollment`), 58

`EnterpriseCourseEnrollmentReadOnlySerializer` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEnrollmentReadOnlySerializer.Meta` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEnrollmentViewSet` (class in `enterprise.api.v1.views.enterprise_course_enrollment`), 58

`EnterpriseCourseEnrollmentWithAdditionalFieldsManager` (class in `enterprise.models`), 146

`EnterpriseCourseEnrollmentWithAdditionalFieldsReadOnlySerializer` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEnrollmentWithAdditionalFieldsReadOnlySerializer.Meta` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEnrollmentWriteSerializer` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEnrollmentWriteSerializer.Meta` (class in `enterprise.api.v1.serializers`), 82

`EnterpriseCourseEntitlement` (class in `enterprise.models`), 146

`EnterpriseCourseEntitlement.DoesNotExist`, 146

`EnterpriseCourseEntitlement.MultipleObjectsReturned`, 146

`EnterpriseCourseEntitlementManager` (class in `enterprise.models`), 148

`EnterpriseCustomer` (class in `enterprise.models`), 148

`EnterpriseCustomer.DoesNotExist`, 148

`EnterpriseCustomer.MultipleObjectsReturned`, 148

`enterprisecustomer_set` (`enterprise.models.EnterpriseCustomerType` attribute), 177

`EnterpriseCustomerAdmin` (class in `enterprise.admin`), 45

`EnterpriseCustomerAdmin.Meta` (class in `enterprise.admin`), 45

`EnterpriseCustomerAdminForm` (class in `enterprise.admin.forms`), 26

`EnterpriseCustomerAdminForm.Meta` (class in `enterprise.admin.forms`), 26

`EnterpriseCustomerApiCredentialRegeneratePatchSerializer` (class in `enterprise.api.v1.serializers`), 83

`EnterpriseCustomerApiCredentialRegeneratePatchSerializer.Meta` (class in `enterprise.api.v1.serializers`), 83

(class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerApiCredentialSerializer (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerApiCredentialSerializer.Meta (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerBasicSerializer (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerBasicSerializer.Meta (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerBrandingConfiguration (class in *enterprise.models*), 161

EnterpriseCustomerBrandingConfiguration.DoesNotExist, 161

EnterpriseCustomerBrandingConfiguration.MultipleObjectsReturned, 161

EnterpriseCustomerBrandingConfigurationInline (class in *enterprise.admin*), 47

EnterpriseCustomerBrandingConfigurationSerializer (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerBrandingConfigurationSerializer.Meta (class in *enterprise.api.v1.serializers*), 83

EnterpriseCustomerBrandingConfigurationViewSet (class in *enterprise.api.v1.views.enterprise_customer_branding_configuration*), 64

EnterpriseCustomerBulkEnrollmentsSerializer (class in *enterprise.api.v1.serializers*), 84

EnterpriseCustomerBulkSubscriptionEnrollmentsSerializer (class in *enterprise.api.v1.serializers*), 84

EnterpriseCustomerCatalog (class in *enterprise.models*), 162

EnterpriseCustomerCatalog.DoesNotExist, 162

EnterpriseCustomerCatalog.MultipleObjectsReturned, 162

EnterpriseCustomerCatalogAdmin (class in *enterprise.admin*), 47

EnterpriseCustomerCatalogAdmin.Media (class in *enterprise.admin*), 47

EnterpriseCustomerCatalogAdmin.Meta (class in *enterprise.admin*), 48

EnterpriseCustomerCatalogAdminForm (class in *enterprise.admin.forms*), 28

EnterpriseCustomerCatalogAdminForm.Meta (class in *enterprise.admin.forms*), 29

EnterpriseCustomerCatalogDetailSerializer (class in *enterprise.api.v1.serializers*), 84

EnterpriseCustomerCatalogInline (class in *enterprise.admin*), 48

EnterpriseCustomerCatalogSerializer (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCatalogSerializer.Meta (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCatalogViewSet (class in *enterprise.api.v1.views.enterprise_customer_catalog*), 65

EnterpriseCustomerCatalogWriteOnlySerializer (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCatalogWriteOnlySerializer.Meta (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCatalogWriteViewSet (class in *enterprise.api.v1.views.enterprise_customer_catalog*), 66

EnterpriseCustomerCourseEnrollmentsListSerializer (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCourseEnrollmentsSerializer (class in *enterprise.api.v1.serializers*), 85

EnterpriseCustomerCourseEnrollmentsSerializer.Meta (class in *enterprise.api.v1.serializers*), 86

EnterpriseCustomerIdentityProvider (class in *enterprise.models*), 166

EnterpriseCustomerIdentityProvider.DoesNotExist, 166

EnterpriseCustomerIdentityProvider.MultipleObjectsReturned, 166

EnterpriseCustomerIdentityProviderAdminForm (class in *enterprise.admin.forms*), 29

EnterpriseCustomerIdentityProviderAdminForm.Meta (class in *enterprise.admin.forms*), 29

EnterpriseCustomerIdentityProviderInline (class in *enterprise.admin*), 49

EnterpriseCustomerIdentityProviderSerializer (class in *enterprise.api.v1.serializers*), 86

EnterpriseCustomerIdentityProviderSerializer.Meta (class in *enterprise.api.v1.serializers*), 86

EnterpriseCustomerInactiveException, 70

EnterpriseCustomerInviteKey (class in *enterprise.models*), 167

EnterpriseCustomerInviteKey.DoesNotExist, 167

EnterpriseCustomerInviteKey.MultipleObjectsReturned, 167

EnterpriseCustomerInviteKeyAdmin (class in *enterprise.admin*), 49

EnterpriseCustomerInviteKeyAdmin.Meta (class in *enterprise.admin*), 49

EnterpriseCustomerInviteKeyFilterBackend (class in *enterprise.api.filters*), 94

EnterpriseCustomerInviteKeyPartialUpdateSerializer (class in *enterprise.api.v1.serializers*), 86

EnterpriseCustomerInviteKeyPartialUpdateSerializer.Meta (class in *enterprise.api.v1.serializers*), 86

EnterpriseCustomerInviteKeyReadOnlySerializer (class in *enterprise.api.v1.serializers*), 87

EnterpriseCustomerInviteKeyReadOnlySerializer.Meta (class in *enterprise.api.v1.serializers*), 87

EnterpriseCustomerInviteKeyViewSet (class in *enterprise.api.v1.views.enterprise_customer_invite_key*), 67

68	EnterpriseCustomerInviteKeyWriteSerializer (class in enterprise.api.v1.serializers), 87	EnterpriseCustomerSsoConfiguration.Meta (class in enterprise.api.v1.serializers), 89
EnterpriseCustomerManageLearnerDataSharingConsentView	173 (class in enterprise.admin.views), 37	EnterpriseCustomerSsoConfiguration.MultipleObjectsReturned
EnterpriseCustomerManageLearnerDataSharingConsentView	173 (class in enterprise.admin.views), 37	EnterpriseCustomerSsoConfigurationAdmin (class in enterprise.admin.views), 40
EnterpriseCustomerManageLearnersView (class in enterprise.admin.views), 38		EnterpriseCustomerSsoConfigurationViewSet (class in enterprise.api.v1.views.enterprise_customer_sso_configuration), 50
EnterpriseCustomerManageLearnersView.ContextParameters	50 (class in enterprise.admin.views), 38	EnterpriseCustomerToggleUniversalLinkSerializer (class in enterprise.api.v1.serializers), 89
EnterpriseCustomerManager (class in enterprise.models), 169		EnterpriseCustomerTransmitCoursesView (class in enterprise.admin.views), 40
EnterpriseCustomerReportingConfigAdminForm (class in enterprise.admin.forms), 30		EnterpriseCustomerTransmitCoursesView.ContextParameters (class in enterprise.admin.views), 41
EnterpriseCustomerReportingConfigAdminForm.Meta (class in enterprise.admin.forms), 30		EnterpriseCustomerType (class in enterprise.models), 177
EnterpriseCustomerReportingConfiguration (class in enterprise.models), 169		EnterpriseCustomerType.DoesNotExist, 177
EnterpriseCustomerReportingConfiguration.DoesNotExist, 170		EnterpriseCustomerType.MultipleObjectsReturned, 177
EnterpriseCustomerReportingConfiguration.MultipleObjectsReturned, 170		EnterpriseCustomerTypeAdmin (class in enterprise.admin), 50
enterprisecustomerreportingconfiguration_set (enterprise.models.EnterpriseCustomerCatalog attribute), 163		EnterpriseCustomerTypeAdmin.Meta (class in enterprise.admin), 50
EnterpriseCustomerReportingConfigurationAdmin (class in enterprise.admin), 49		EnterpriseCustomerUnlinkUsersSerializer (class in enterprise.api.v1.serializers), 89
EnterpriseCustomerReportingConfigurationAdmin.Meta (class in enterprise.admin), 49		EnterpriseCustomerUser (class in enterprise.models), 178
EnterpriseCustomerReportingConfigurationSerializer (class in enterprise.api.v1.serializers), 87		EnterpriseCustomerUser.DoesNotExist, 178
EnterpriseCustomerReportingConfigurationSerializer.Meta (class in enterprise.api.v1.serializers), 87		EnterpriseCustomerUser.MultipleObjectsReturned, 178
EnterpriseCustomerReportingConfigurationViewSet (class in enterprise.api.v1.views.enterprise_customer_reporting), 69		EnterpriseCustomerUserAdmin (class in enterprise.admin), 50
EnterpriseCustomerReportTypesView (class in enterprise.api.v1.views.enterprise_customer_reporting), 69		EnterpriseCustomerUserAdmin.Meta (class in enterprise.admin), 50
EnterpriseCustomerSerializer (class in enterprise.api.v1.serializers), 88		EnterpriseCustomerUserFilterBackend (class in enterprise.api.filters), 94
EnterpriseCustomerSerializer.Meta (class in enterprise.api.v1.serializers), 88		EnterpriseCustomerUserManager (class in enterprise.models), 181
EnterpriseCustomerSetupAuthOrgIDView (class in enterprise.admin.views), 40		EnterpriseCustomerUserReadOnlySerializer (class in enterprise.api.v1.serializers), 89
EnterpriseCustomerSsoConfiguration (class in enterprise.api.v1.serializers), 89		EnterpriseCustomerUserReadOnlySerializer.Meta (class in enterprise.api.v1.serializers), 89
EnterpriseCustomerSsoConfiguration (class in enterprise.models), 173		EnterpriseCustomerUserViewSet (class in enterprise.api.v1.views.enterprise_customer_user), 71
EnterpriseCustomerSsoConfiguration.DoesNotExist, 173		EnterpriseCustomerUserWriteSerializer (class in enterprise.api.v1.serializers), 90
		EnterpriseCustomerUserWriteSerializer.Meta (class in enterprise.api.v1.serializers), 90
		EnterpriseCustomerViewSet (class in enterprise.api.v1.views.enterprise_customer), 60

[EnterpriseEnrollmentSource](#) (class in *enterprise.models*), 182
[EnterpriseEnrollmentSource.DoesNotExist](#), 182
[EnterpriseEnrollmentSource.MultipleObjectsReturned](#), 182
[EnterpriseFeatureRole](#) (class in *enterprise.models*), 183
[EnterpriseFeatureRole.DoesNotExist](#), 183
[EnterpriseFeatureRole.MultipleObjectsReturned](#), 183
[EnterpriseFeatureUserRoleAssignment](#) (class in *enterprise.models*), 184
[EnterpriseFeatureUserRoleAssignment.DoesNotExist](#), 184
[EnterpriseFeatureUserRoleAssignment.MultipleObjectsReturned](#), 184
[enterprisefeatureuserroleassignment_set](#) (*enterprise.models.EnterpriseFeatureRole* attribute), 184
[EnterpriseFeatureUserRoleAssignmentAdmin](#) (class in *enterprise.admin*), 51
[EnterpriseFeatureUserRoleAssignmentAdmin.Meta](#) (class in *enterprise.admin*), 51
[EnterpriseFeatureUserRoleAssignmentForm](#) (class in *enterprise.admin.forms*), 31
[EnterpriseFeatureUserRoleAssignmentForm.Meta](#) (class in *enterprise.admin.forms*), 31
[EnterpriseFulfillmentSource](#) (class in *enterprise.models*), 186
[EnterpriseFulfillmentSource.Meta](#) (class in *enterprise.models*), 186
[EnterpriseGroup](#) (class in *enterprise.models*), 187
[EnterpriseGroup.DoesNotExist](#), 187
[EnterpriseGroup.MultipleObjectsReturned](#), 187
[EnterpriseGroupAdmin](#) (class in *enterprise.admin*), 52
[EnterpriseGroupLearnersRequestQuerySerializer](#) (class in *enterprise.api.v1.serializers*), 90
[EnterpriseGroupMembership](#) (class in *enterprise.models*), 189
[EnterpriseGroupMembership.DoesNotExist](#), 189
[EnterpriseGroupMembership.MultipleObjectsReturned](#), 189
[EnterpriseGroupMembershipAdmin](#) (class in *enterprise.admin*), 52
[EnterpriseGroupMembershipSerializer](#) (class in *enterprise.api.v1.serializers*), 90
[EnterpriseGroupMembershipSerializer.Meta](#) (class in *enterprise.api.v1.serializers*), 90
[EnterpriseGroupRequestDataSerializer](#) (class in *enterprise.api.v1.serializers*), 91
[EnterpriseGroupSerializer](#) (class in *enterprise.api.v1.serializers*), 91
[EnterpriseGroupSerializer.Meta](#) (class in *enterprise.api.v1.serializers*), 91
[EnterpriseGroupViewSet](#) (class in *enterprise.api.v1.views.enterprise_group*), 72
[EnterpriseLanguagePreferenceMiddleware](#) (class in *enterprise.middleware*), 134
[EnterpriseLinkedUserFilterBackend](#) (class in *enterprise.api.filters*), 94
[EnterpriseLoginForm](#) (class in *enterprise.forms*), 132
[EnterpriseLoginView](#) (class in *enterprise.views*), 268
[EnterpriseModelViewSet](#) (class in *enterprise.api.v1.views.base_views*), 55
[EnterpriseProxyLoginView](#) (class in *enterprise.views*), 268
[EnterpriseReadOnlyModelViewSet](#) (class in *enterprise.api.v1.views.base_views*), 55
[EnterpriseReadWriteModelViewSet](#) (class in *enterprise.api.v1.views.base_views*), 55
[EnterpriseRequestIdLoggerAdapter](#) (class in *enterprise.logging*), 132
[EnterpriseRoleAssignmentContextMixin](#) (class in *enterprise.models*), 191
[EnterpriseSelectionForm](#) (class in *enterprise.forms*), 132
[EnterpriseSelectionView](#) (class in *enterprise.views*), 268
[EnterpriseSSOOrchestratorApiClient](#) (class in *enterprise.api_client.sso_orchestrator*), 112
[EnterpriseSubsidyFulfillmentViewSet](#) (class in *enterprise.api.v1.views.enterprise_subsidy_fulfillment*), 74
[EnterpriseViewSet](#) (class in *enterprise.api.v1.views.base_views*), 56
[EnterpriseWrapperApiViewSet](#) (class in *enterprise.api.v1.views.base_views*), 56
[EnterpriseWriteOnlyModelViewSet](#) (class in *enterprise.api.v1.views.base_views*), 56
[entity_id](#) (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 174
[entity_id](#) (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 209
[EntityIdNotFoundError](#), 70
[errored_at](#) (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 174
[errored_at](#) (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 209
[expand_button\(\)](#) (in module *enterprise.template_tags.enterprise*), 127
[ExpandDefaultPageSize](#) (class in *enterprise.api.v1.views.enterprise_catalog_query*), 58
[expiration_date](#) (*enterprise.models.AdminNotification* attribute), 135
[expiration_date](#) (

<code>prise.models.EnterpriseCustomerInviteKey</code> <code>attribute</code>), 168	<code>fields</code> (<code>enterprise.admin.EnterpriseCustomerUserAdmin</code> <code>attribute</code>), 51
<code>expiration_date</code> (<code>enterprise.models.HistoricalEnterpriseCustomerInviteKey</code> <code>attribute</code>), 206	<code>fields</code> (<code>enterprise.admin.forms.AdminNotificationForm.Meta</code> <code>attribute</code>), 26
<code>export_as_csv_action()</code> (in module <code>enterprise.admin.actions</code>), 25	<code>fields</code> (<code>enterprise.admin.forms.EnterpriseCustomerAdminForm.Meta</code> <code>attribute</code>), 26
<code>EXPORT_AS_CSV_FIELDS</code> (<code>enterprise.admin.EnterpriseCustomerAdmin</code> <code>attribute</code>), 45	<code>fields</code> (<code>enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm.Meta</code> <code>attribute</code>), 29
<code>extend_course()</code> (<code>enterprise.views.ProgramEnrollmentView</code> <code>static method</code>), 270	<code>fields</code> (<code>enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm.Meta</code> <code>attribute</code>), 29
<code>EXTERNALLY_MANAGED</code> (<code>enterprise.models.EnterpriseCustomer</code> <code>attribute</code>), 148	<code>fields</code> (<code>enterprise.admin.forms.EnterpriseCustomerReportingConfigAdminForm.Meta</code> <code>attribute</code>), 30
<code>extra</code> (<code>enterprise.admin.EnterpriseCustomerCatalogInline</code> <code>attribute</code>), 48	<code>fields</code> (<code>enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentForm.Meta</code> <code>attribute</code>), 31
<code>extra</code> (<code>enterprise.admin.EnterpriseCustomerIdentityProviderAdminForm</code> <code>attribute</code>), 49	<code>fields</code> (<code>enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentForm.Meta</code> <code>attribute</code>), 34
<code>extra</code> (<code>enterprise.admin.PendingEnterpriseCustomerAdminForm</code> <code>attribute</code>), 53	<code>fields</code> (<code>enterprise.admin.PendingEnterpriseCustomerAdminUserAdminForm.Meta</code> <code>attribute</code>), 53
<code>extra_kwargs</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerCatalogQuerySerializer.Meta</code> <code>attribute</code>), 85	<code>fields</code> (<code>enterprise.admin.PendingEnterpriseCustomerUserAdminForm.Meta</code> <code>attribute</code>), 54
F	<code>fields</code> (<code>enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdminForm.Meta</code> <code>attribute</code>), 54
<code>fa_icon()</code> (in module <code>enterprise.templatetags.enterprise</code>), 127	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCatalogWriteOnlySerializer.Meta</code> <code>attribute</code>), 80
<code>FailedEnrollmentReason</code> (class in <code>enterprise.views</code>), 268	<code>fields</code> (<code>enterprise.api.v1.serializers.BaseEnterpriseCustomerInviteKeySerializer.Meta</code> <code>attribute</code>), 81
<code>failure_reason_message</code> (<code>enterprise.views.FailedEnrollmentReason</code> <code>attribute</code>), 268	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCatalogQuerySerializer.Meta</code> <code>attribute</code>), 82
<code>feature_prequery_search_suggestions()</code> (in module <code>enterprise.toggles</code>), 247	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCourseEnrollmentReadOnlySerializer.Meta</code> <code>attribute</code>), 82
<code>fetch_configuration_record()</code> (in module <code>enterprise.api.v1.views.enterprise_customer_sso_configuration</code>), 71	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWithAdditionSerializer.Meta</code> <code>attribute</code>), 82
<code>fetch_entity_id_from_metadata_xml()</code> (in module <code>enterprise.api.v1.views.enterprise_customer_sso_configuration</code>), 71	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWriteOnlySerializer.Meta</code> <code>attribute</code>), 83
<code>fetch_request_data_from_request()</code> (in module <code>enterprise.api.v1.views.enterprise_customer_sso_configuration</code>), 71	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerBasicSerializer.Meta</code> <code>attribute</code>), 83
<code>fields</code> (<code>enterprise.admin.EnterpriseCatalogQueryAdminForm</code> <code>attribute</code>), 44	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationSerializer.Meta</code> <code>attribute</code>), 84
<code>fields</code> (<code>enterprise.admin.EnterpriseCustomerCatalogAdminForm</code> <code>attribute</code>), 48	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerCatalogSerializer.Meta</code> <code>attribute</code>), 85
<code>fields</code> (<code>enterprise.admin.EnterpriseCustomerInviteKeyAdminForm</code> <code>attribute</code>), 49	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerCatalogWriteOnlySerializer.Meta</code> <code>attribute</code>), 85
<code>fields</code> (<code>enterprise.admin.EnterpriseCustomerTypeAdminForm</code> <code>attribute</code>), 50	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerIdentityProviderSerializer.Meta</code> <code>attribute</code>), 86
	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyPartialUpdateSerializer.Meta</code> <code>attribute</code>), 86
	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyReadOnlySerializer.Meta</code> <code>attribute</code>), 87
	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerReportingConfigurationSerializer.Meta</code> <code>attribute</code>), 87
	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerSerializer.Meta</code> <code>attribute</code>), 88
	<code>fields</code> (<code>enterprise.api.v1.serializers.EnterpriseCustomerSsoConfigurationSerializer.Meta</code> <code>attribute</code>), 89

`prise.models.EnterpriseCustomerSsoConfiguration` (attribute), 174
`first_name_attribute` (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 209
`force_fresh_session()` (in module `enterprise.decorators`), 130
`form` (enterprise.admin.AdminNotificationAdmin attribute), 42
`form` (enterprise.admin.EnterpriseCustomerAdmin attribute), 46
`form` (enterprise.admin.EnterpriseCustomerCatalogInline attribute), 48
`form` (enterprise.admin.EnterpriseCustomerIdentityProvider attribute), 49
`form` (enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin attribute), 50
`form` (enterprise.admin.EnterpriseFeatureUserRoleAssignmentAdmin attribute), 51
`form` (enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin attribute), 54
`form_class` (enterprise.views.EnterpriseLoginView attribute), 268
`form_class` (enterprise.views.EnterpriseSelectionView attribute), 268
`form_invalid()` (enterprise.views.EnterpriseLoginView method), 268
`form_invalid()` (enterprise.views.EnterpriseSelectionView method), 268
`form_valid()` (enterprise.views.EnterpriseLoginView method), 268
`form_valid()` (enterprise.views.EnterpriseSelectionView method), 268
`format_price()` (in module `enterprise.utils`), 254
`frequency` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 171
`FREQUENCY_CHOICES` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
`FREQUENCY_TYPE_DAILY` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
`FREQUENCY_TYPE_MONTHLY` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
`FREQUENCY_TYPE_WEEKLY` (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 170
`from_db()` (enterprise.models.EnterpriseCustomerInviteKey class method), 168
`fulfill_pending_course_enrollments()` (enterprise.models.PendingEnterpriseCustomerUser method), 239
`fulfill_pending_group_memberships()` (enterprise.models.PendingEnterpriseCustomerUser method), 240
`fulfillment_type` (enterprise.models.EnterpriseFulfillmentSource attribute), 186
`fulfillment_type` (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 220
`fulfillment_type` (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
`fulfillment_type` (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234
`fulfillment_type` (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 235
`FullfillmentTypes` (class in `enterprise.constants`), 129
`full_name_attribute` (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 174
`full_name_attribute` (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 209

G

`GENERAL_ERRORS` (enterprise.admin.forms.ManageLearnersForm.Fields attribute), 32
`generate_default_orchestration_record_display_name()` (in module `enterprise.signals`), 245
`generate_mailto_link()` (enterprise.api_client.braze.BrazeAPIClient method), 99
`generate_prompt_for_learner_engagement_summary()` (in module `enterprise.api.utils`), 97
`generate_prompt_for_learner_progress_summary()` (in module `enterprise.api.utils`), 97
`genericenterprisepluginconfiguration_set` (enterprise.models.EnterpriseCustomer attribute), 155
`get()` (enterprise.admin.views.CatalogQueryPreviewView method), 37
`get()` (enterprise.admin.views.EnterpriseCustomerManageLearnerDataSh method), 37
`get()` (enterprise.admin.views.EnterpriseCustomerManageLearnersView method), 38
`get()` (enterprise.admin.views.EnterpriseCustomerSetupAuthOrgIDView method), 40
`get()` (enterprise.admin.views.EnterpriseCustomerTransmitCoursesView method), 41

`get()` (*enterprise.admin.views.TemplatePreviewView* method), 41
`get()` (*enterprise.api.v1.views.enterprise_customer_reporting_assignment_types()* method), 69
`get()` (*enterprise.api.v1.views.plotly_auth.PlotlyAuthView* method), 79
`get()` (*enterprise.models.EnterpriseCustomerUserManager* method), 181
`get()` (*enterprise.views.CourseEnrollmentView* method), 267
`get()` (*enterprise.views.EnterpriseProxyLoginView* method), 268
`get()` (*enterprise.views.GrantDataSharingPermissions* method), 269
`get()` (*enterprise.views.HandleConsentEnrollment* method), 269
`get()` (*enterprise.views.ProgramEnrollmentView* method), 270
`get()` (*enterprise.views.RouterView* method), 271
`get_actions()` (*enterprise.admin.EnterpriseCustomerCatalogAdmin* method), 48
`get_active_course_runs()` (in module *enterprise.utils*), 254
`get_active_enterprise_users()` (*enterprise.models.EnterpriseCustomerUser* class method), 179
`get_active_integrations()` (*enterprise.api.v1.serializers.EnterpriseCustomerSerializer* method), 89
`get_admin_registration_url()` (*enterprise.admin.PendingEnterpriseCustomerAdminUserAdmin* method), 53
`get_admin_registration_url()` (*enterprise.admin.PendingEnterpriseCustomerAdminUserInline* method), 53
`get_admin_users()` (*enterprise.api.v1.serializers.EnterpriseCustomerSerializer* method), 89
`get_advertised_course_run()` (in module *enterprise.utils*), 254
`get_advertised_or_closest_course_run()` (in module *enterprise.utils*), 254
`get_all_field_names()` (in module *enterprise.utils*), 254
`get_all_learners()` (*enterprise.models.EnterpriseGroup* method), 188
`get_api_url()` (*enterprise.api_client.client.APIClientMixin* method), 99
`get_app_config()` (in module *enterprise.validators*), 266
`get_args_from_database()` (*enterprise.management.commands.bulk_update_catalog_query_id* method), 118
`get_assignment_types()` (*enterprise.models.SystemWideEnterpriseUserRoleAssignment* class method), 242
`get_available_course_modes()` (*enterprise.views.CourseEnrollmentView* method), 267
`get_base_details()` (*enterprise.views.CourseEnrollmentView* method), 267
`get_best_mode_from_course_key()` (in module *enterprise.utils*), 254
`get_branding_configuration()` (*enterprise.api.v1.serializers.EnterpriseCustomerSerializer* method), 89
`get_cache_key()` (in module *enterprise.utils*), 254
`get_catalog_admin_url()` (in module *enterprise.utils*), 254
`get_catalog_admin_url_template()` (in module *enterprise.utils*), 255
`get_catalog_content_count()` (*enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient* method), 105
`get_catalog_diff()` (*enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient* method), 105
`get_catalog_preview_uuid()` (*enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm* static method), 29
`get_catalog_results()` (*enterprise.api_client.discovery.CourseCatalogApiClient* method), 100
`get_catalog_results_from_discovery()` (*enterprise.api_client.discovery.CourseCatalogApiClient* method), 100
`get_change_actions()` (*enterprise.admin.EnterpriseCustomerAdmin* method), 46
`get_closest_course_run()` (in module *enterprise.utils*), 255
`get_configuration_value()` (in module *enterprise.utils*), 255
`get_configuration_value_for_site()` (in module *enterprise.utils*), 255
`get_content_filter()` (*enterprise.models.EnterpriseCustomerCatalog* method), 163
`get_content_metadata()` (*enterprise.api_client.enterprise.EnterpriseApiClient* method), 104
`get_content_metadata()` (*enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient* method), 105

<code>get_content_metadata_content_identifier()</code> (<i>enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient</i> method), 105	<code>get_course_details()</code> (<i>enterprise.api_client.lms.CourseApiClient</i> method), 107
<code>GET_CONTENT_METADATA_ENDPOINT</code> (<i>enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient</i> attribute), 104	<code>get_course_details()</code> (<i>enterprise.api_client.lms.EnrollmentApiClient</i> method), 108
<code>get_content_metadata_item_id()</code> (in module <i>enterprise.utils</i>), 255	<code>get_course_enrollment()</code> (<i>enterprise.api_client.lms.EnrollmentApiClient</i> method), 108
<code>GET_CONTENT_METADATA_PAGE_SIZE</code> (<i>enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient</i> attribute), 104	<code>get_course_enrollment_url()</code> (<i>enterprise.models.EnterpriseCustomer</i> method), 155
<code>get_content_metadata_url()</code> (<i>enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient</i> static method), 105	<code>get_course_enrollment_url()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> method), 164
<code>get_context()</code> (<i>enterprise.models.EnterpriseFeatureUserRoleAssignment</i> method), 185	<code>get_course_final_price()</code> (<i>enterprise.api_client.ecommerce.EcommerceApiClient</i> method), 103
<code>get_context()</code> (<i>enterprise.models.SystemWideEnterpriseUserRoleAssignment</i> method), 243	<code>get_course_grade()</code> (<i>enterprise.api_client.lms.GradesApiClient</i> method), 110
<code>get_context_data()</code> (<i>enterprise.views.EnterpriseLoginView</i> method), 268	<code>get_course_id()</code> (<i>enterprise.api_client.discovery.CourseCatalogApiClient</i> method), 101
<code>get_context_data()</code> (<i>enterprise.views.EnterpriseSelectionView</i> method), 268	<code>get_course_modes()</code> (<i>enterprise.api_client.lms.EnrollmentApiClient</i> method), 108
<code>get_context_from_db()</code> (<i>enterprise.views.GrantDataSharingPermissions</i> method), 269	<code>get_course_or_program_context()</code> (<i>enterprise.views.GrantDataSharingPermissions</i> method), 269
<code>get_country_display()</code> (<i>enterprise.models.EnterpriseCustomer</i> method), 155	<code>get_course_run()</code> (<i>enterprise.api_client.discovery.CourseCatalogApiClient</i> method), 101
<code>get_country_display()</code> (<i>enterprise.models.HistoricalEnterpriseCustomer</i> method), 201	<code>get_course_run()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> method), 164
<code>get_course()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> method), 163	<code>get_course_run_duration_info()</code> (in module <i>enterprise.utils</i>), 255
<code>get_course_and_course_run()</code> (<i>enterprise.api_client.discovery.CourseCatalogApiClient</i> method), 100	<code>get_course_run_enrollment_url()</code> (<i>enterprise.models.EnterpriseCustomer</i> method), 156
<code>get_course_and_course_run()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> method), 164	<code>get_course_run_enrollment_url()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> method), 164
<code>get_course_assessment_grades()</code> (<i>enterprise.api_client.lms.GradesApiClient</i> method), 110	<code>get_course_run_id()</code> (<i>enterprise.views.RouterView</i> static method), 271
<code>get_course_catalog_api_service_client()</code> (in module <i>enterprise.api_client.discovery</i>), 102	<code>get_course_run_identifiers()</code> (<i>enterprise.api_client.discovery.CourseCatalogApiClient</i> method), 101
<code>get_course_certificate()</code> (<i>enterprise.api_client.lms.CertificatesApiClient</i> method), 106	<code>get_course_run_start()</code> (in module <i>enterprise.utils</i>), 256
<code>get_course_details()</code> (<i>enterprise.api_client.discovery.CourseCatalogApiClient</i> method), 100	<code>get_course_track_selection_url()</code> (in module <i>enterprise.utils</i>), 256
	<code>get_create_ent_enrollment()</code> (in module <i>enterprise</i>), 107

prise.utils), 256
get_current_course_run() (in module *enterprise.utils*), 256
get_customer_from_request() (in module *enterprise.api.v1.views.enterprise_customer_sso_configuration*), 71
get_data_sharing_consent_records() (enter-
prise.api.v1.serializers.EnterpriseCustomerUserRegistration
method), 90
get_data_sharing_consent_text_overrides()
(enterprise.models.EnterpriseCustomer
method), 156
get_data_type_display() (enter-
prise.models.EnterpriseCustomerReportingConfiguration
method), 172
get_day_of_week_display() (enter-
prise.models.EnterpriseCustomerReportingConfiguration
method), 172
get_default_catalog_content_filter() (in mod-
ule *enterprise.utils*), 256
get_default_context() (enter-
prise.views.GrantDataSharingPermissions
method), 269
get_default_customer_type() (in module *enterprise.models*), 244
get_default_history_user() (enter-
prise.models.HistoricalEnrollmentNotificationEmail
static method), 191
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCourseEnrollment
static method), 194
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCourseEntitlement
static method), 196
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCustomer
static method), 201
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCustomerCatalog
static method), 204
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCustomerInviteKey
static method), 206
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCustomerSsoConfiguration
static method), 209
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseCustomerUserRegistration
static method), 213
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseGroup
static method), 215
get_default_history_user() (enter-
prise.models.HistoricalEnterpriseGroupMemberships
static method), 217
get_default_history_user() (enter-
prise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment
static method), 220
get_default_history_user() (enter-
prise.models.HistoricalLicensedEnterpriseCourseEnrollment
static method), 223
get_default_history_user() (enter-
prise.models.HistoricalPendingEnrollment
static method), 225
get_default_history_user() (enter-
prise.models.HistoricalPendingEnterpriseCustomerAdminUser
static method), 228
get_default_history_user() (enter-
prise.models.HistoricalPendingEnterpriseCustomerUser
static method), 230
get_default_history_user() (enter-
prise.models.HistoricalSystemWideEnterpriseUserRoleAssignment
static method), 232
get_default_idp_user_social_auth() (in module
enterprise.tpa_pipeline), 247
get_default_invite_key_expiration_date() (in
module *enterprise.utils*), 256
get_default_language_display() (enter-
prise.models.EnterpriseCustomer method),
156
get_default_language_display() (enter-
prise.models.HistoricalEnterpriseCustomer
method), 201
get_default_site() (in module *enterprise.models*),
244
get_delivery_method_display() (enter-
prise.models.EnterpriseCustomerReportingConfiguration
method), 172
get_distinct_assignments_by_role_name() (en-
terprise.models.SystemWideEnterpriseUserRoleAssignment
class method), 243
get_duration_of_course_or_courserun() (in mod-
ule *enterprise.utils*), 256
get_ecommerce_worker_user() (in module *enterprise.utils*), 257
get_enforce_data_sharing_consent_display()
(enterprise.models.EnterpriseCustomer
method), 156
get_enforce_data_sharing_consent_display()
(enterprise.models.HistoricalEnterpriseCustomer
method), 201
get_enrolled_course_string() (enter-
prise.admin.EnterpriseCustomerUserAdmin
method), 51
get_enrolled_courses() (enter-
prise.api_client.lms.EnrollmentApiClient
method), 108
get_ent_cust_from_enterprise_customer_key()

(in module *enterprise.api.utils*), 97

get_ent_cust_from_report_config_uuid() (in module *enterprise.api.utils*), 97

get_enterprise_catalog() (enterprise.api_client.enterprise_catalog.EnterpriseCatalogAdminClient method), 105

get_enterprise_course_enrollment_id() (enterprise.models.EnterpriseCourseEnrollment class method), 144

get_enterprise_course_enrollment_page() (enterprise.views.CourseEnrollmentView method), 267

get_enterprise_course_enrollments() (enterprise.management.commands.email_drip_for_missed_emails.Command method), 119

get_enterprise_customer() (enterprise.admin.EnterpriseCustomerUserAdmin class method), 51

get_enterprise_customer() (enterprise.admin.PendingEnterpriseCustomerAdminUser class method), 53

get_enterprise_customer() (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationSerializer method), 84

get_enterprise_customer() (enterprise.api.v1.serializers.EnterpriseCustomerSsoConfigurationSerializer method), 89

get_enterprise_customer() (in module *enterprise.utils*), 257

get_enterprise_customer_by_invite_key_or_404() (in module *enterprise.utils*), 257

get_enterprise_customer_by_slug_or_404() (in module *enterprise.utils*), 257

get_enterprise_customer_catalogs() (enterprise.api.v1.serializers.EnterpriseCustomerSerializer method), 89

get_enterprise_customer_for_running_pipeline() (in module *enterprise.tpa_pipeline*), 247

get_enterprise_customer_for_sso() (in module *enterprise.tpa_pipeline*), 247

get_enterprise_customer_for_user() (in module *enterprise.utils*), 257

get_enterprise_customer_from_catalog_id() (in module *enterprise.api.utils*), 97

get_enterprise_customer_from_enterprise_group_id() (in module *enterprise.api.utils*), 97

get_enterprise_customer_from_user_id() (in module *enterprise.api.utils*), 97

get_enterprise_customer_name() (enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyReadOnlySerializer method), 87

get_enterprise_customer_or_404() (in module *enterprise.utils*), 257

get_enterprise_customer_user() (in module *enterprise.utils*), 258

get_enterprise_customer_user_queryset() (enterprise.admin.views.EnterpriseCustomerManageLearnersView class method), 38

get_enterprise_customer_uuid() (enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyReadOnlySerializer method), 87

get_enterprise_group() (enterprise.api.v1.serializers.EnterpriseCustomerUserReadOnlySerializer method), 90

get_enterprise_notification_banner() (enterprise.api.v1.serializers.EnterpriseCustomerSerializer method), 89

get_enterprise_program_enrollment_page() (enterprise.views.ProgramEnrollmentView method), 271

get_enterprise_slug() (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationSerializer method), 84

get_enterprise_utm_context() (in module *enterprise.utils*), 258

get_enterprise_uuids_for_user_and_course() (in module *enterprise.utils*), 258

get_enterprise_uuids_with_user_and_course() (enterprise.models.EnterpriseCourseEnrollment class method), 144

get_enterprise_worker_user() (in module *enterprise.utils*), 258

get_failed_enrollment_message() (enterprise.admin.views.EnterpriseCustomerManageLearnersView class method), 39

get_fields() (enterprise.admin.EnterpriseCustomerReportingConfiguration class method), 50

get_form() (enterprise.admin.EnterpriseCustomerAdmin class method), 46

get_form() (enterprise.admin.EnterpriseCustomerCatalogAdmin class method), 48

get_form_view() (enterprise.admin.views.BaseEnterpriseCustomerView method), 37

get_formset() (enterprise.admin.EnterpriseCustomerCatalogInline class method), 48

get_frequency_display() (enterprise.models.EnterpriseCustomerReportingConfiguration class method), 172

get_fulfillment_type_display() (enterprise.models.EnterpriseFulfillmentSource class method), 186

get_fulfillment_type_display() (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment class method), 221

get_fulfillment_type_display() (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment class method), 221

method), 223

get_fulfillment_type_display() (enterprise.models.LearnerCreditEnterpriseCourseEnrollment method), 234

get_fulfillment_type_display() (enterprise.models.LicensedEnterpriseCourseEnrollment method), 236

get_global_context() (in module enterprise.views), 272

get_groups() (enterprise.api.v1.serializers.EnterpriseCustomerOnlySerializer method), 90

get_health() (enterprise.api_client.client.NoAuthAPIClient method), 99

get_health() (enterprise.api_client.lms.NoAuthLMSClient method), 111

get_history_type_display() (enterprise.models.HistoricalEnrollmentNotificationEmail method), 191

get_history_type_display() (enterprise.models.HistoricalEnterpriseCourseEnrollment method), 194

get_history_type_display() (enterprise.models.HistoricalEnterpriseCourseEntitlement method), 196

get_history_type_display() (enterprise.models.HistoricalEnterpriseCustomer method), 201

get_history_type_display() (enterprise.models.HistoricalEnterpriseCustomerCatalog method), 204

get_history_type_display() (enterprise.models.HistoricalEnterpriseCustomerInvite method), 206

get_history_type_display() (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration method), 210

get_history_type_display() (enterprise.models.HistoricalEnterpriseCustomerUser method), 213

get_history_type_display() (enterprise.models.HistoricalEnterpriseGroup method), 215

get_history_type_display() (enterprise.models.HistoricalEnterpriseGroupMemberships method), 217

get_history_type_display() (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment method), 221

get_history_type_display() (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment method), 223

get_history_type_display() (enterprise.models.HistoricalPendingEnrollment method), 225

get_history_type_display() (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser method), 228

get_history_type_display() (enterprise.models.HistoricalPendingEnterpriseCustomerUser method), 230

get_history_type_display() (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment method), 232

get_id_by_email() (in module enterprise.utils), 258

get_idiff_list() (in module enterprise.utils), 258

get_idp_choices() (in module enterprise.utils), 259

get_initial() (enterprise.views.EnterpriseSelectionView method), 268

get_integrated_channel_choices() (in module enterprise.utils), 259

get_integrations_for_customers() (in module enterprise.utils), 259

get_is_pending_configuration() (enterprise.api.v1.serializers.EnterpriseCustomerSsoConfiguration method), 89

get_language_code() (in module enterprise.utils), 259

get_last_course_run_end_date() (in module enterprise.utils), 259

get_learner_portal_url() (in module enterprise.utils), 259

get_learners() (enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet method), 72

get_link_by_email() (enterprise.models.EnterpriseCustomerUserManager method), 181

get_logo() (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfiguration method), 84

get_md5_hash() (in module enterprise.utils), 259

get_member_details() (enterprise.api.v1.serializers.EnterpriseGroupMembershipSerializer method), 91

get_metadata_xml_from_url() (in module enterprise.api.v1.views.enterprise_customer_sso_configuration), 71

get_missing_params_message() (enterprise.api.v1.views.coupon_codes.CouponCodesView method), 56

get_missing_params_message() (enterprise.api.v1.views.notifications.NotificationReadView method), 77

get_next_by_change_date() (enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfiguration method), 114

get_next_by_change_date() (enterprise.models.BulkCatalogQueryUpdateCommandConfiguration method), 114

<i>method</i>), 138		<i>method</i>), 182	
<code>get_next_by_created()</code> (<i>enterprise.models.AdminNotification</i> <i>method</i>), 135		<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseFeatureRole</i> <i>method</i>), 184	
<code>get_next_by_created()</code> (<i>enterprise.models.AdminNotificationFilter</i> <i>method</i>), 136		<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseFeatureUserRoleAssignment</i> <i>method</i>), 185	
<code>get_next_by_created()</code> (<i>enterprise.models.AdminNotificationRead</i> <i>method</i>), 137		<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseFulfillmentSource</i> <i>method</i>), 186	
<code>get_next_by_created()</code> (<i>enterprise.models.ChatGPTResponse</i> <i>method</i>), 139		<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseGroup</i> <i>method</i>), 188	
<code>get_next_by_created()</code> (<i>enterprise.models.EnrollmentNotificationEmailTemplate</i> <i>method</i>), 141		<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseGroupMembership</i> <i>method</i>), 189	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCatalogQuery</i> <i>method</i>), 143		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnrollmentNotificationEmailTemplate</i> <i>method</i>), 191	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCourseEnrollment</i> <i>method</i>), 144		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEnrollment</i> <i>method</i>), 194	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCourseEntitlement</i> <i>method</i>), 147		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEntitlement</i> <i>method</i>), 196	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomer</i> <i>method</i>), 156		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCustomer</i> <i>method</i>), 201	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerBrandingConfiguration</i> <i>method</i>), 161		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerCatalog</i> <i>method</i>), 204	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> <i>method</i>), 164		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerInviteKey</i> <i>method</i>), 206	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerIdentityProvider</i> <i>method</i>), 167		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration</i> <i>method</i>), 210	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerInviteKey</i> <i>method</i>), 168		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerUser</i> <i>method</i>), 213	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerReportingConfiguration</i> <i>method</i>), 172		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseGroup</i> <i>method</i>), 215	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerSsoConfiguration</i> <i>method</i>), 175		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalEnterpriseGroupMembership</i> <i>method</i>), 217	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerType</i> <i>method</i>), 177		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 221	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseCustomerUser</i> <i>method</i>), 179		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment</i> <i>method</i>), 223	
<code>get_next_by_created()</code> (<i>enterprise.models.EnterpriseEnrollmentSource</i> <i>method</i>), 182		<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalPendingEnrollment</i> <i>method</i>), 225	

<i>method</i>), 225	<i>method</i>), 204
<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser</i> <i>method</i>), 228	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerInviteKey</i> <i>method</i>), 206
<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerUser</i> <i>method</i>), 230	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration</i> <i>method</i>), 210
<code>get_next_by_created()</code> (<i>enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</i> <i>method</i>), 232	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerUser</i> <i>method</i>), 213
<code>get_next_by_created()</code> (<i>enterprise.models.LearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 234	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseGroup</i> <i>method</i>), 215
<code>get_next_by_created()</code> (<i>enterprise.models.LicensedEnterpriseCourseEnrollment</i> <i>method</i>), 236	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseGroupMembership</i> <i>method</i>), 217
<code>get_next_by_created()</code> (<i>enterprise.models.PendingEnrollment</i> <i>method</i>), 237	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 221
<code>get_next_by_created()</code> (<i>enterprise.models.PendingEnterpriseCustomerAdminUser</i> <i>method</i>), 238	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment</i> <i>method</i>), 223
<code>get_next_by_created()</code> (<i>enterprise.models.PendingEnterpriseCustomerUser</i> <i>method</i>), 240	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalPendingEnrollment</i> <i>method</i>), 225
<code>get_next_by_created()</code> (<i>enterprise.models.SystemWideEnterpriseRole</i> <i>method</i>), 241	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser</i> <i>method</i>), 228
<code>get_next_by_created()</code> (<i>enterprise.models.SystemWideEnterpriseUserRoleAssignment</i> <i>method</i>), 243	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerUser</i> <i>method</i>), 230
<code>get_next_by_expiration_date()</code> (<i>enterprise.models.AdminNotification</i> <i>method</i>), 135	<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</i> <i>method</i>), 232
<code>get_next_by_expiration_date()</code> (<i>enterprise.models.EnterpriseCustomerInviteKey</i> <i>method</i>), 168	<code>get_next_by_modified()</code> (<i>enterprise.models.AdminNotification</i> <i>method</i>), 135
<code>get_next_by_expiration_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerInviteKey</i> <i>method</i>), 206	<code>get_next_by_modified()</code> (<i>enterprise.models.AdminNotificationFilter</i> <i>method</i>), 136
<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnrollmentNotificationEmailTemplate</i> <i>method</i>), 191	<code>get_next_by_modified()</code> (<i>enterprise.models.AdminNotificationRead</i> <i>method</i>), 137
<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEnrollment</i> <i>method</i>), 194	<code>get_next_by_modified()</code> (<i>enterprise.models.ChatGPTResponse</i> <i>method</i>), 139
<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEntitlement</i> <i>method</i>), 196	<code>get_next_by_modified()</code> (<i>enterprise.models.EnrollmentNotificationEmailTemplate</i> <i>method</i>), 141
<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomer</i> <i>method</i>), 201	<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCatalogQuery</i> <i>method</i>), 143
<code>get_next_by_history_date()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerCatalog</i> <i>method</i>), 204	<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCourseEnrollment</i> <i>method</i>), 204

<i>method</i>), 144		<i>method</i>), 194	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCourseEntitlement</i> <i>method</i>), 147		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEntitlement</i> <i>method</i>), 196	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomer</i> <i>method</i>), 156		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCustomer</i> <i>method</i>), 201	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerBrandingConfiguration</i> <i>method</i>), 161		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerCatalog</i> <i>method</i>), 204	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerCatalog</i> <i>method</i>), 165		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerInviteKey</i> <i>method</i>), 206	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerIdentityProvider</i> <i>method</i>), 167		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration</i> <i>method</i>), 210	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerInviteKey</i> <i>method</i>), 168		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCustomerUser</i> <i>method</i>), 213	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerReportingConfiguration</i> <i>method</i>), 172		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseGroup</i> <i>method</i>), 215	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerSsoConfiguration</i> <i>method</i>), 175		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseGroupMembership</i> <i>method</i>), 217	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerType</i> <i>method</i>), 177		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 221	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseCustomerUser</i> <i>method</i>), 179		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment</i> <i>method</i>), 223	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseEnrollmentSource</i> <i>method</i>), 183		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalPendingEnrollment</i> <i>method</i>), 225	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseFeatureRole</i> <i>method</i>), 184		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser</i> <i>method</i>), 228	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseFeatureUserRoleAssignment</i> <i>method</i>), 185		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalPendingEnterpriseCustomerUser</i> <i>method</i>), 230	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseFulfillmentSource</i> <i>method</i>), 186		<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</i> <i>method</i>), 232	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseGroup</i> <i>method</i>), 188		<code>get_next_by_modified()</code> (<i>enterprise.models.LearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 234	
<code>get_next_by_modified()</code> (<i>enterprise.models.EnterpriseGroupMembership</i> <i>method</i>), 189		<code>get_next_by_modified()</code> (<i>enterprise.models.LicensedEnterpriseCourseEnrollment</i> <i>method</i>), 236	
<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnrollmentNotificationEmailTemplate</i> <i>method</i>), 191		<code>get_next_by_modified()</code> (<i>enterprise.models.PendingEnrollment</i> <i>method</i>), 237	
<code>get_next_by_modified()</code> (<i>enterprise.models.HistoricalEnterpriseCourseEnrollment</i> <i>method</i>), 191		<code>get_next_by_modified()</code> (<i>enterprise.models.PendingEnterpriseCustomerAdminUser</i> <i>method</i>), 237	

- method*), 238
- `get_next_by_modified()` (*enterprise.models.PendingEnterpriseCustomerUser method*), 240
- `get_next_by_modified()` (*enterprise.models.SystemWideEnterpriseRole method*), 241
- `get_next_by_modified()` (*enterprise.models.SystemWideEnterpriseUserRoleAssignment method*), 243
- `get_next_by_start_date()` (*enterprise.models.AdminNotification method*), 135
- `get_notification_subject_line()` (*in module enterprise.utils*), 259
- `get_oauth2authentication_class()` (*in module enterprise.utils*), 260
- `get_or_create()` (*enterprise.models.ChatGPTResponse class method*), 139
- `get_or_create_system_wide_role()` (*in module enterprise.roles_api*), 244
- `get_page_language_context_data()` (*enterprise.views.GrantDataSharingPermissions method*), 269
- `get_paginated_content()` (*enterprise.models.EnterpriseCustomerCatalog method*), 165
- `get_paginated_response()` (*enterprise.api.pagination.PaginationWithFeatureFlags method*), 95
- `get_paginated_response()` (*in module enterprise.api.pagination*), 95
- `get_path_variables()` (*enterprise.views.RouterView static method*), 272
- `get_pending_enrollment_message()` (*enterprise.admin.views.EnterpriseCustomerManageLearnersView class method*), 39
- `get_pending_users_queryset()` (*enterprise.admin.views.EnterpriseCustomerManageLearnersView method*), 39
- `get_permissions()` (*enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet method*), 61
- `get_platform_logo_url()` (*in module enterprise.utils*), 260
- `get_previous_by_change_date()` (*enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfig method*), 114
- `get_previous_by_change_date()` (*enterprise.models.BulkCatalogQueryUpdateCommandConfiguration method*), 138
- `get_previous_by_created()` (*enterprise.models.AdminNotification method*), 135
- `get_previous_by_created()` (*enterprise.models.AdminNotificationFilter method*), 136
- `get_previous_by_created()` (*enterprise.models.AdminNotificationRead method*), 137
- `get_previous_by_created()` (*enterprise.models.ChatGPTResponse method*), 139
- `get_previous_by_created()` (*enterprise.models.EnrollmentNotificationEmailTemplate method*), 141
- `get_previous_by_created()` (*enterprise.models.EnterpriseCatalogQuery method*), 143
- `get_previous_by_created()` (*enterprise.models.EnterpriseCourseEnrollment method*), 144
- `get_previous_by_created()` (*enterprise.models.EnterpriseCourseEntitlement method*), 147
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomer method*), 156
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerBrandingConfiguration method*), 161
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerCatalog method*), 165
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerIdentityProvider method*), 167
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerInviteKey method*), 168
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerReportingConfiguration method*), 172
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerSsoConfiguration method*), 175
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerType method*), 177
- `get_previous_by_created()` (*enterprise.models.EnterpriseCustomerUser method*), 179
- `get_previous_by_created()` (*enterprise.models.EnterpriseEnrollmentSource method*), 183
- `get_previous_by_created()` (*enterprise.models.EnterpriseFeatureRole method*), 185

184		method), 228	
get_previous_by_created()	(enterprise.models.EnterpriseFeatureUserRoleAssignment method), 185	get_previous_by_created()	(enterprise.models.HistoricalPendingEnterpriseCustomerUser method), 230
get_previous_by_created()	(enterprise.models.EnterpriseFulfillmentSource method), 186	get_previous_by_created()	(enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment method), 232
get_previous_by_created()	(enterprise.models.EnterpriseGroup method), 188	get_previous_by_created()	(enterprise.models.LearnerCreditEnterpriseCourseEnrollment method), 234
get_previous_by_created()	(enterprise.models.EnterpriseGroupMembership method), 189	get_previous_by_created()	(enterprise.models.LicensedEnterpriseCourseEnrollment method), 236
get_previous_by_created()	(enterprise.models.HistoricalEnrollmentNotificationEmailTemplate method), 191	get_previous_by_created()	(enterprise.models.PendingEnrollment method), 237
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCourseEnrollment method), 194	get_previous_by_created()	(enterprise.models.PendingEnterpriseCustomerAdminUser method), 238
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCourseEntitlement method), 196	get_previous_by_created()	(enterprise.models.PendingEnterpriseCustomerUser method), 240
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCustomer method), 201	get_previous_by_created()	(enterprise.models.SystemWideEnterpriseRole method), 241
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCustomerCatalog method), 204	get_previous_by_created()	(enterprise.models.SystemWideEnterpriseUserRoleAssignment method), 243
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCustomerInviteKey method), 207	get_previous_by_expiration_date()	(enterprise.models.AdminNotification method), 135
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration method), 210	get_previous_by_expiration_date()	(enterprise.models.EnterpriseCustomerInviteKey method), 168
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseCustomerUser method), 213	get_previous_by_expiration_date()	(enterprise.models.HistoricalEnterpriseCustomerInviteKey method), 207
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseGroup method), 215	get_previous_by_history_date()	(enterprise.models.HistoricalEnrollmentNotificationEmailTemplate method), 192
get_previous_by_created()	(enterprise.models.HistoricalEnterpriseGroupMembership method), 218	get_previous_by_history_date()	(enterprise.models.HistoricalEnterpriseCourseEnrollment method), 194
get_previous_by_created()	(enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment method), 221	get_previous_by_history_date()	(enterprise.models.HistoricalEnterpriseCourseEntitlement method), 196
get_previous_by_created()	(enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment method), 223	get_previous_by_history_date()	(enterprise.models.HistoricalEnterpriseCustomer method), 201
get_previous_by_created()	(enterprise.models.HistoricalPendingEnrollment method), 225	get_previous_by_history_date()	(enterprise.models.HistoricalEnterpriseCustomerCatalog method), 204
get_previous_by_created()	(enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser method), 228	get_previous_by_history_date()	(enterprise.models.HistoricalEnterpriseCustomerInviteKey method), 207

<i>method</i>), 207	<i>method</i>), 147
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalEnterpriseCustomerSsoConfiguration</i> <i>method</i>), 210	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomer</i> <i>method</i>), 156
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalEnterpriseCustomerUser</i> <i>method</i>), 213	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerBrandingConfiguration</i> <i>method</i>), 162
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalEnterpriseGroup</i> <i>method</i>), 215	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerCatalog</i> <i>method</i>), 165
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalEnterpriseGroupMembership</i> <i>method</i>), 218	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerIdentityProvider</i> <i>method</i>), 167
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</i> <i>method</i>), 221	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerInviteKey</i> <i>method</i>), 168
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalLicensedEnterpriseCourseEnrollment</i> <i>method</i>), 223	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerReportingConfiguration</i> <i>method</i>), 172
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalPendingEnrollment</i> <i>method</i>), 225	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerSsoConfiguration</i> <i>method</i>), 175
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalPendingEnterpriseCustomerAdminUser</i> <i>method</i>), 228	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerType</i> <i>method</i>), 177
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalPendingEnterpriseCustomerUser</i> <i>method</i>), 230	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCustomerUser</i> <i>method</i>), 179
<code>get_previous_by_history_date()</code> (enter- <i>prise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</i> <i>method</i>), 232	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseEnrollmentSource</i> <i>method</i>), 183
<code>get_previous_by_modified()</code> (enter- <i>prise.models.AdminNotification</i> <i>method</i>), 135	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseFeatureRole</i> <i>method</i>), 184
<code>get_previous_by_modified()</code> (enter- <i>prise.models.AdminNotificationFilter</i> <i>method</i>), 136	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseFeatureUserRoleAssignment</i> <i>method</i>), 185
<code>get_previous_by_modified()</code> (enter- <i>prise.models.AdminNotificationRead</i> <i>method</i>), 137	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseFulfillmentSource</i> <i>method</i>), 186
<code>get_previous_by_modified()</code> (enter- <i>prise.models.ChatGPTResponse</i> <i>method</i>), 140	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseGroup</i> <i>method</i>), 188
<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnrollmentNotificationEmailTemplate</i> <i>method</i>), 141	<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseGroupMembership</i> <i>method</i>), 189
<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCatalogQuery</i> <i>method</i>), 143	<code>get_previous_by_modified()</code> (enter- <i>prise.models.HistoricalEnrollmentNotificationEmailTemplate</i> <i>method</i>), 192
<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCourseEnrollment</i> <i>method</i>), 144	<code>get_previous_by_modified()</code> (enter- <i>prise.models.HistoricalEnterpriseCourseEnrollment</i> <i>method</i>), 194
<code>get_previous_by_modified()</code> (enter- <i>prise.models.EnterpriseCourseEntitlement</i> <i>method</i>), 147	<code>get_previous_by_modified()</code> (enter- <i>prise.models.HistoricalEnterpriseCourseEntitlement</i> <i>method</i>), 197

<i>method</i>), 196		<i>method</i>), 240	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseCustomer <i>method</i>), 201		<code>get_previous_by_modified()</code> (enterprise.models.SystemWideEnterpriseRole <i>method</i>), 241	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseCustomerCatalog <i>method</i>), 204		<code>get_previous_by_modified()</code> (enterprise.models.SystemWideEnterpriseUserRoleAssignment <i>method</i>), 243	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseCustomerInviteKey <i>method</i>), 207		<code>get_previous_by_start_date()</code> (enterprise.models.AdminNotification <i>method</i>), 135	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseCustomerSsoConfig <i>method</i>), 210		<code>get_price_text()</code> (in module <code>enterprise.views</code>), 272	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseCustomerUser <i>method</i>), 213		<code>get_primary_color()</code> (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfiguration <i>method</i>), 84	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseGroup <i>method</i>), 216		<code>get_program()</code> (enterprise.models.EnterpriseCustomerCatalog <i>method</i>), 165	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseGroupMemberships <i>method</i>), 218		<code>get_program_by_uuid()</code> (enterprise.api_client.discovery.CourseCatalogApiClient <i>method</i>), 101	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalEnterpriseGroupMemberships <i>method</i>), 218		<code>get_program_course_keys()</code> (enterprise.api_client.discovery.CourseCatalogApiClient <i>method</i>), 101	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment <i>method</i>), 221		<code>get_program_details()</code> (enterprise.views.ProgramEnrollmentView <i>method</i>), 271	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment <i>method</i>), 223		<code>get_program_enrollment_url()</code> (enterprise.models.EnterpriseCustomer <i>method</i>), 156	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalPendingEnrollment <i>method</i>), 225		<code>get_program_enrollment_url()</code> (enterprise.models.EnterpriseCustomerCatalog <i>method</i>), 165	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalPendingEnterpriseCustomerEnrollment <i>method</i>), 228		<code>get_program_type_by_slug()</code> (enterprise.api_client.discovery.CourseCatalogApiClient <i>method</i>), 102	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalPendingEnterpriseCustomerEnrollment <i>method</i>), 230		<code>get_program_type_description()</code> (in module <code>enterprise.utils</code>), 260	
<code>get_previous_by_modified()</code> (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment <i>method</i>), 232		<code>get_prompt_type_display()</code> (enterprise.models.ChatGPTResponse <i>method</i>), 140	
<code>get_previous_by_modified()</code> (enterprise.models.LearnerCreditEnterpriseCourseEnrollment <i>method</i>), 234		<code>get_query_results_from_snowflake()</code> (enterprise.management.commands.monthly_impact_report.Command <i>method</i>), 122	
<code>get_previous_by_modified()</code> (enterprise.models.LicensedEnterpriseCourseEnrollment <i>method</i>), 236		<code>get_query_results_from_snowflake()</code> (enterprise.management.commands.nudge_dormant_enrolled_enterpris <i>method</i>), 123	
<code>get_previous_by_modified()</code> (enterprise.models.PendingEnrollment <i>method</i>), 237		<code>get_queryset()</code> (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollment <i>method</i>), 58	
<code>get_previous_by_modified()</code> (enterprise.models.PendingEnterpriseCustomerAdminUser <i>method</i>), 238		<code>get_queryset()</code> (enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentials <i>method</i>), 63	
<code>get_previous_by_modified()</code> (enterprise.models.PendingEnterpriseCustomerUser <i>method</i>), 238		<code>get_queryset()</code> (enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentials <i>method</i>), 63	

method), 64

get_queryset() (enterprise.views), 272

prise.api.v1.views.enterprise_group.EnterpriseGroupViewSet() (enterprise.api.throttles.BaseThrottle method), 73

get_queryset() (enterprise.models.EnterpriseCourseEnrollmentManager method), 146

get_queryset() (enterprise.models.EnterpriseCourseEnrollmentWithAdditionalFieldsManager method), 146

get_queryset() (enterprise.models.EnterpriseCourseEntitlementManager method), 148

get_queryset() (enterprise.models.EnterpriseCustomerManager method), 169

get_queryset() (enterprise.models.EnterpriseCustomerUserManager method), 181

get_readonly_fields() (enterprise.admin.EnterpriseCustomerInviteKeyAdmin method), 49

get_readonly_fields() (enterprise.admin.EnterpriseCustomerUserAdmin method), 51

get_recent_action() (enterprise.api.v1.serializers.EnterpriseGroupMembershipsSerializer method), 91

get_remote_id() (enterprise.api_client.lms.ThirdPartyAuthApiClient method), 111

get_remote_id() (enterprise.models.EnterpriseCustomerUser method), 179

get_report_type_display() (enterprise.models.EnterpriseCustomerReportingConfiguration method), 172

get_request_id() (in module enterprise.logging), 133

get_request_value() (in module enterprise.utils), 260

get_required_query_params() (enterprise.api.v1.views.coupon_codes.CouponCodesView method), 57

get_required_query_params() (enterprise.api.v1.views.notifications.NotificationReadView method), 77

get_response_from_discovery() (enterprise.admin.views.CatalogQueryPreviewView static method), 37

get_role_assignments() (enterprise.api.v1.serializers.EnterpriseCustomerUserRegistrationSerializer method), 90

get_role_display() (enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfiguration method), 114

get_safe_redirect_url() (in module enterprise.views), 272

get_scope() (enterprise.api.throttles.BaseThrottle method), 96

get_scope() (enterprise.api.throttles.HighServiceUserThrottle method), 96

get_scope() (enterprise.api.throttles.ServiceUserThrottle method), 96

get_search_results() (enterprise.admin.views.EnterpriseCustomerManageLearnersView method), 39

get_search_results() (enterprise.admin.EnterpriseCustomerAdmin method), 46

get_search_results() (enterprise.admin.EnterpriseCustomerUserAdmin method), 51

get_secondary_color() (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfiguration method), 84

get_serializer_class() (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentView method), 59

get_serializer_class() (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView method), 61

get_serializer_class() (enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsView method), 63

get_serializer_class() (enterprise.api.v1.views.enterprise_customer_api_credentials.APICredentialsView method), 64

get_serializer_class() (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView method), 66

get_serializer_class() (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyView method), 68

get_serializer_class() (enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserView method), 71

get_service_usernames() (in module enterprise.api.utils), 98

get_social_auth_from_idp() (in module enterprise.utils), 260

get_source() (enterprise.models.EnterpriseEnrollmentSource class method), 183

get_sso_orchestrator_api_base_url() (in module enterprise.utils), 260

get_sso_orchestrator_basic_auth_password() (in module enterprise.utils), 260

get_sso_orchestrator_basic_auth_username() (in module enterprise.utils), 260

get_sso_orchestrator_configure_edx_oauth_path() (in module enterprise.utils), 260

(in module `enterprise.utils`), 260

`get_sso_orchestrator_configure_path()` (in module `enterprise.utils`), 261

`get_sso_provider()` (in module `enterprise.tpa_pipeline`), 247

`get_status_display()` (`enterprise.models.EnterpriseGroupMembership` method), 189

`get_status_display()` (`enterprise.models.HistoricalEnterpriseGroupMembership` method), 218

`get_subsidy_fulfillment_queryset()` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet` method), 75

`get_subsidy_fulfillment_serializer_class()` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet` method), 75

`get_success_enrollment_message()` (`enterprise.admin.views.EnterpriseCustomerManageLead` class method), 39

`get_template_type_display()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 141

`get_template_type_display()` (`enterprise.models.HistoricalEnrollmentNotificationEmailTemplate` method), 192

`get_tertiary_color()` (`enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationSerializer` method), 84

`get_tpa_hint()` (`enterprise.models.EnterpriseCustomer` method), 156

`get_unenrolled_fulfillment_queryset()` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet` method), 75

`get_unenrolled_fulfillment_serializer_class()` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet` method), 75

`get_urls()` (`enterprise.admin.EnrollmentNotificationEmailTemplateAdmin` method), 43

`get_urls()` (`enterprise.admin.EnterpriseCatalogQueryAdmin` method), 44

`get_urls()` (`enterprise.admin.EnterpriseCourseEnrollmentAdmin` method), 44

`get_urls()` (`enterprise.admin.EnterpriseCustomerAdmin` method), 46

`get_user_from_social_auth()` (in module `enterprise.tpa_pipeline`), 247

`get_user_name()` (`enterprise.admin.views.TemplatePreviewView` static method), 41

`get_user_social_auth()` (in module `enterprise.tpa_pipeline`), 247

`get_user_valid_idp()` (in module `enterprise.utils`), 261

`get_username_from_remote_id()` (`enterprise.api_client.lms.ThirdPartyAuthApiClient` method), 111

`get_users_by_email()` (in module `enterprise.utils`), 261

`getEnterpriseLogger()` (in module `enterprise.logging`), 132

`GradesApiClient` (class in `enterprise.api_client.lms`), 110

`GrantDataSharingPermissions` (class in `enterprise.views`), 269

`EnterpriseSubsidyFulfillmentViewSet` (`EnterpriseSubsidyFulfillmentViewSet` attribute), 189

`group` (`enterprise.models.HistoricalEnterpriseGroupMembership` attribute), 189

`group_id` (`enterprise.models.EnterpriseGroupMembership` attribute), 190

`group_id` (`enterprise.models.HistoricalEnterpriseGroupMembership` attribute), 218

`groups` (`enterprise.models.EnterpriseCustomer` attribute), 157

H

`handle()` (`enterprise.management.commands.add_exec_ed_exclusion_to_` method), 117

`handle()` (`enterprise.management.commands.backfill_learner_role_assignment` method), 117

`handle()` (`enterprise.management.commands.bulk_update_catalog_query` method), 118

`handle()` (`enterprise.management.commands.compare_discovery_and_enrollment` method), 118

`handle()` (`enterprise.management.commands.create_enterprise_course_enrollment` method), 119

`handle()` (`enterprise.management.commands.create_missing_dsc_records` method), 119

`handle()` (`enterprise.management.commands.email_drip_for_missing_dsc_records` method), 119

`handle()` (`enterprise.management.commands.ensure_singular_active_enrollment` method), 120

`handle()` (`enterprise.management.commands.fix_dsc_records.Command` method), 120

`handle()` (`enterprise.management.commands.manufacture_data.Command` method), 121

`handle()` (`enterprise.management.commands.migrate_enterprise_catalogs` method), 122

`handle()` (`enterprise.management.commands.monthly_impact_report.Command` method), 122

`handle()` (`enterprise.management.commands.nudge_dormant_enrolled_enrollment` method), 123

`handle()` (`enterprise.management.commands.reencrypt_enterprise_customers` method), 123

`handle()` (`enterprise.management.commands.remove_expired_pending_group` method), 124

`handle()` (`enterprise.management.commands.revert_enrollment_objects.Command` (enter-
 prise.models.EnterpriseCustomer property),
 method), 124 157
`handle()` (`enterprise.management.commands.save_enterprise_catalog()` (`enterprise.management.commands.EnterpriseCustomerAdmin`
 method), 125 method), 46
`handle()` (`enterprise.management.commands.seed_enterprise_devstack_data()` (`enterprise.management.commands`
 method), 125 `enterprise.models.EnterpriseCustomer` property),
`handle()` (`enterprise.management.commands.unlink_enterprise_customer_learners.Command`
 method), 125 `has_permission()` (enter-
 prise.models.EnterpriseCustomer property),
`handle()` (`enterprise.management.commands.update_config_last_errored_at()` (`enterprise.management.commands.IsInEnterpriseGroup`
 method), 126 method), 80
`handle()` (`enterprise.management.commands.update_role_assignments()` (`enterprise.management.commands`
 method), 126 `enterprise.models.EnterpriseCustomer` property),
`HANDLE_CONSENT_ENROLLMENT_VIEW_URL` (enter-
 prise.views.RouterView attribute), 271 `has_single_idp` (enter-
`handle_enterprise_logistration()` (in module `enterprise.tpa_pipeline`), 247 `prise.models.EnterpriseCustomer` property),
`handle_redirect_after_social_auth_login()` (in module `enterprise.tpa_pipeline`), 248 157
`handle_user_post_save()` (in module `enterprise.signals`), 245 `heartbeat()` (in module `enterprise.heartbeat.views`),
`HandleConsentEnrollment` (class in `enterprise.views`), 269 116
`has_access_to_all_contexts()` (enter-
 prise.models.SystemWideEnterpriseUserRoleAssignment attribute), 243 `HeartbeatRateThrottle` (class in enter-
 prise.heartbeat.throttles), 116
`has_add_permission()` (enter-
 prise.admin.EnterpriseCourseEnrollmentAdmin method), 44 `help` (`enterprise.management.commands.backfill_learner_role_assignment`
 attribute), 117
`has_add_permission()` (enter-
 prise.admin.PendingEnrollmentAdmin attribute), 118
 method), 52 `help` (`enterprise.management.commands.bulk_update_catalog_query_id.C`
 attribute), 119
`has_api_credentials_enabled()` (in module `enterprise.api.utils`), 98 `help` (`enterprise.management.commands.create_enterprise_course_enrollm`
 attribute), 122
`has_course_mode()` (enter-
 prise.api_client.lms.EnrollmentApiClient attribute), 124
 method), 109 `help` (`enterprise.management.commands.remove_expired_pending_group_`
 attribute), 124
`has_course_run_available_for_enrollment()` (in module `enterprise.utils`), 261 `help` (`enterprise.management.commands.revert_enrollment_objects.Comm`
 attribute), 124
`has_delete_permission()` (enter-
 prise.admin.EnterpriseCatalogQueryAdmin attribute), 125
 method), 44 `help` (`enterprise.management.commands.save_enterprise_customer_users.C`
 attribute), 125
`has_delete_permission()` (enter-
 prise.admin.EnterpriseCourseEnrollmentAdmin attribute), 125
 method), 44 `help` (`enterprise.management.commands.seed_enterprise_devstack_data.C`
 attribute), 125
`has_delete_permission()` (enter-
 prise.admin.PendingEnrollmentAdmin attribute), 126
 method), 52 `help` (`enterprise.management.commands.update_config_last_errored_at.C`
 attribute), 126
`has_enterprise_customer_catalog()` (enter-
 prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogView attribute), 67 `help` (`enterprise.management.commands.update_role_assignments_with_c`
 attribute), 126
`has_identity_provider()` (enter-
 prise.admin.EnterpriseCustomerAdmin here()) (in module `enterprise.settings.test`), 127
 method), 46 `hide_course_original_price` (enter-
 prise.models.EnterpriseCustomer attribute),
`has_identity_providers` (enter- 157
`hide_course_original_price` (enter-
 prise.models.HistoricalEnterpriseCustomer attribute), 202
 attribute), 202
`HighServiceUserThrottle` (class in enter-
 prise.models.HistoricalEnterpriseCustomer attribute), 202
 class in enter-

prise.api.throttles), 96

`HistoricalEnrollmentNotificationEmailTemplate` (class in *enterprise.models*), 191

`HistoricalEnrollmentNotificationEmailTemplate.DoesNotExist`, 191

`HistoricalEnrollmentNotificationEmailTemplate.MultipleObjectsReturned`, 191

`HistoricalEnterpriseCourseEnrollment` (class in *enterprise.models*), 193

`HistoricalEnterpriseCourseEnrollment.DoesNotExist`, 193

`HistoricalEnterpriseCourseEnrollment.MultipleObjectsReturned`, 193

`HistoricalEnterpriseCourseEntitlement` (class in *enterprise.models*), 195

`HistoricalEnterpriseCourseEntitlement.DoesNotExist`, 195

`HistoricalEnterpriseCourseEntitlement.MultipleObjectsReturned`, 195

`HistoricalEnterpriseCustomer` (class in *enterprise.models*), 197

`HistoricalEnterpriseCustomer.DoesNotExist`, 198

`HistoricalEnterpriseCustomer.MultipleObjectsReturned`, 198

`HistoricalEnterpriseCustomerCatalog` (class in *enterprise.models*), 203

`HistoricalEnterpriseCustomerCatalog.DoesNotExist`, 203

`HistoricalEnterpriseCustomerCatalog.MultipleObjectsReturned`, 203

`HistoricalEnterpriseCustomerInviteKey` (class in *enterprise.models*), 206

`HistoricalEnterpriseCustomerInviteKey.DoesNotExist`, 206

`HistoricalEnterpriseCustomerInviteKey.MultipleObjectsReturned`, 206

`HistoricalEnterpriseCustomerSsoConfiguration` (class in *enterprise.models*), 208

`HistoricalEnterpriseCustomerSsoConfiguration.DoesNotExist`, 208

`HistoricalEnterpriseCustomerSsoConfiguration.MultipleObjectsReturned`, 209

`HistoricalEnterpriseCustomerUser` (class in *enterprise.models*), 212

`HistoricalEnterpriseCustomerUser.DoesNotExist`, 212

`HistoricalEnterpriseCustomerUser.MultipleObjectsReturned`, 212

`HistoricalEnterpriseGroup` (class in *enterprise.models*), 215

`HistoricalEnterpriseGroup.DoesNotExist`, 215

`HistoricalEnterpriseGroup.MultipleObjectsReturned`, 215

`HistoricalEnterpriseGroupMembership` (class in *enterprise.models*), 217

`HistoricalEnterpriseGroupMembership.DoesNotExist`, 217

`HistoricalEnterpriseGroupMembership.MultipleObjectsReturned`, 217

`HistoricalLearnerCreditEnterpriseCourseEnrollment` (class in *enterprise.models*), 219

`HistoricalLearnerCreditEnterpriseCourseEnrollment.DoesNotExist`, 220

`HistoricalLearnerCreditEnterpriseCourseEnrollment.MultipleObjectsReturned`, 220

`HistoricalLicensedEnterpriseCourseEnrollment` (class in *enterprise.models*), 222

`HistoricalLicensedEnterpriseCourseEnrollment.DoesNotExist`, 222

`HistoricalLicensedEnterpriseCourseEnrollment.MultipleObjectsReturned`, 222

`HistoricalPendingEnrollment` (class in *enterprise.models*), 225

`HistoricalPendingEnrollment.DoesNotExist`, 225

`HistoricalPendingEnrollment.MultipleObjectsReturned`, 225

`HistoricalPendingEnterpriseCustomerAdminUser` (class in *enterprise.models*), 227

`HistoricalPendingEnterpriseCustomerAdminUser.DoesNotExist`, 227

`HistoricalPendingEnterpriseCustomerAdminUser.MultipleObjectsReturned`, 227

`HistoricalPendingEnterpriseCustomerUser` (class in *enterprise.models*), 229

`HistoricalPendingEnterpriseCustomerUser.DoesNotExist`, 229

`HistoricalPendingEnterpriseCustomerUser.MultipleObjectsReturned`, 229

`HistoricalSystemWideEnterpriseUserRoleAssignment` (class in *enterprise.models*), 231

`HistoricalSystemWideEnterpriseUserRoleAssignment.DoesNotExist`, 231

`HistoricalSystemWideEnterpriseUserRoleAssignment.MultipleObjectsReturned`, 231

`HistoricalEnrollmentNotificationEmailTemplate` (attribute), 141

`history` (*enterprise.models.EnterpriseCourseEnrollment* attribute), 145

`history` (*enterprise.models.EnterpriseCourseEntitlement* attribute), 147

`history` (*enterprise.models.EnterpriseCustomer* attribute), 157

`history` (*enterprise.models.EnterpriseCustomerCatalog* attribute), 165

`history` (*enterprise.models.EnterpriseCustomerInviteKey* attribute), 168

`history` (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 168

attribute), 175
 history (enterprise.models.EnterpriseCustomerUser attribute), 180
 history (enterprise.models.EnterpriseGroup attribute), 188
 history (enterprise.models.EnterpriseGroupMembership attribute), 190
 history (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234
 history (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 236
 history (enterprise.models.PendingEnrollment attribute), 237
 history (enterprise.models.PendingEnterpriseCustomerAdminUser attribute), 239
 history (enterprise.models.PendingEnterpriseCustomerUser attribute), 240
 history (enterprise.models.SystemWideEnterpriseUserRoleAssignment attribute), 243
 history_change_reason (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 192
 history_change_reason (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 194
 history_change_reason (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196
 history_change_reason (enterprise.models.HistoricalEnterpriseCustomer attribute), 202
 history_change_reason (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 205
 history_change_reason (enterprise.models.HistoricalEnterpriseCustomerInviteKey attribute), 207
 history_change_reason (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 210
 history_change_reason (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 213
 history_change_reason (enterprise.models.HistoricalEnterpriseGroup attribute), 216
 history_change_reason (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 218
 history_change_reason (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 221
 history_change_reason (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
 history_change_reason (enterprise.models.HistoricalPendingEnrollment attribute), 226
 history_change_reason (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser attribute), 228
 history_change_reason (enterprise.models.HistoricalPendingEnterpriseCustomerUser attribute), 230
 history_change_reason (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 232
 history_id (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 192
 history_id (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 194
 history_id (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196
 history_id (enterprise.models.HistoricalEnterpriseCustomer attribute), 202
 history_id (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 205
 history_id (enterprise.models.HistoricalEnterpriseCustomerInviteKey attribute), 207
 history_id (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 210
 history_id (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 213
 history_id (enterprise.models.HistoricalEnterpriseGroup attribute), 216
 history_id (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 218
 history_id (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 221
 history_id (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
 history_id (enterprise.models.HistoricalPendingEnrollment attribute), 226
 history_id (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser attribute), 228
 history_id (enterprise.models.HistoricalPendingEnterpriseCustomerUser attribute), 230
 history_id (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 232
 history_id (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 192
 history_id (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 194
 history_id (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 196
 history_id (enterprise.models.HistoricalEnterpriseCustomer attribute), 202
 history_id (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 205
 history_id (enterprise.models.HistoricalEnterpriseCustomerInviteKey attribute), 207
 history_id (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 210
 history_id (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 213
 history_id (enterprise.models.HistoricalEnterpriseGroup attribute), 216
 history_id (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 218
 history_id (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 221
 history_id (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 223
 history_id (enterprise.models.HistoricalPendingEnrollment attribute), 226
 history_id (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser attribute), 228
 history_id (enterprise.models.HistoricalPendingEnterpriseCustomerUser attribute), 230
 history_id (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 232

attribute), 205	prise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment
history_id(enterprise.models.HistoricalEnterpriseCustomerInviteKey	attribute), 221
attribute), 207	history_object (enter-
history_id(enterprise.models.HistoricalEnterpriseCustomerSsoConfiguratio	prise.models.HistoricalLicensedEnterpriseCourseEnrollment
attribute), 210	attribute), 224
history_id(enterprise.models.HistoricalEnterpriseCustomerUser	history_object (enter-
attribute), 213	prise.models.HistoricalPendingEnrollment
history_id(enterprise.models.HistoricalEnterpriseGroup	attribute), 226
attribute), 216	history_object (enter-
history_id(enterprise.models.HistoricalEnterpriseGroupMembership	prise.models.HistoricalPendingEnterpriseCustomerAdminUser
attribute), 218	attribute), 228
history_id(enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment	history_object (enter-
attribute), 221	prise.models.HistoricalPendingEnterpriseCustomerUser
history_id(enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment)	attribute), 230
attribute), 224	history_object (enter-
history_id(enterprise.models.HistoricalPendingEnrollment	prise.models.HistoricalSystemWideEnterpriseUserRoleAssignment
attribute), 226	attribute), 232
history_id(enterprise.models.HistoricalPendingEnterpriseCustomerTypeLookup	prise.models.HistoricalEnrollmentNotificationEmailTemplate
attribute), 228	attribute), 192
history_id(enterprise.models.HistoricalPendingEnterpriseCustomerTypeLookup	history_type(enterprise.models.HistoricalEnterpriseCourseEnrollment
attribute), 230	attribute), 194
history_id(enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment	history_type(enterprise.models.HistoricalEnterpriseCourseEntitlement
attribute), 232	attribute), 196
history_object (enter-	history_type(enterprise.models.HistoricalEnterpriseCustomer
prise.models.HistoricalEnrollmentNotificationEmailTemplate	attribute), 202
attribute), 192	history_type(enterprise.models.HistoricalEnterpriseCustomerCatalog
history_object (enter-	attribute), 205
prise.models.HistoricalEnterpriseCourseEnrollment	history_type(enterprise.models.HistoricalEnterpriseCustomerInviteKey
attribute), 194	attribute), 207
history_object (enter-	history_type(enterprise.models.HistoricalEnterpriseCustomerSsoConfiguratio
prise.models.HistoricalEnterpriseCourseEntitlement	attribute), 210
attribute), 196	history_type(enterprise.models.HistoricalEnterpriseCustomerUser
history_object (enter-	attribute), 213
prise.models.HistoricalEnterpriseCustomer	history_type(enterprise.models.HistoricalEnterpriseGroup
attribute), 202	attribute), 216
history_object (enter-	history_type(enterprise.models.HistoricalEnterpriseGroupMembership
prise.models.HistoricalEnterpriseCustomerCatalog	attribute), 218
attribute), 205	history_type(enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment
history_object (enter-	attribute), 221
prise.models.HistoricalEnterpriseCustomerInviteKey	history_type(enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment
attribute), 207	attribute), 224
history_object (enter-	history_type(enterprise.models.HistoricalPendingEnrollment
prise.models.HistoricalEnterpriseCustomerSsoConfiguration	attribute), 226
attribute), 210	history_type(enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser
history_object (enter-	attribute), 228
prise.models.HistoricalEnterpriseCustomerUser	history_type(enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser
attribute), 213	attribute), 230
history_object (enter-	history_type(enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment
prise.models.HistoricalEnterpriseGroup	attribute), 232
attribute), 216	history_user(enterprise.models.HistoricalEnrollmentNotificationEmailTemplate
history_object (enter-	attribute), 192
prise.models.HistoricalEnterpriseGroupMembership	history_user(enterprise.models.HistoricalEnterpriseCourseEnrollment
attribute), 218	attribute), 194
history_object (enter-	history_user(enterprise.models.HistoricalEnterpriseCourseEntitlement

<code>id</code> (<code>enterprise.models.EnterpriseCourseEnrollment</code> attribute), 145	<code>prise.models.EnterpriseCustomer</code> property), 157
<code>id</code> (<code>enterprise.models.EnterpriseCourseEntitlement</code> attribute), 147	<code>identity_provider</code> (<code>enterprise.models.EnterpriseCustomerIdProvider</code> property), 167
<code>id</code> (<code>enterprise.models.EnterpriseCustomerBrandingConfiguration</code> attribute), 162	<code>identity_provider</code> (<code>enterprise.models.EnterpriseCustomerSsoConfiguration</code> attribute), 175
<code>id</code> (<code>enterprise.models.EnterpriseCustomerIdProvider</code> attribute), 167	<code>identity_provider</code> (<code>enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration</code> attribute), 210
<code>id</code> (<code>enterprise.models.EnterpriseCustomerReportingConfiguration</code> attribute), 172	<code>identity_provider_ids</code> (<code>enterprise.models.EnterpriseCustomer</code> property), 157
<code>id</code> (<code>enterprise.models.EnterpriseCustomerType</code> attribute), 177	<code>identity_providers</code> (<code>enterprise.models.EnterpriseCustomer</code> property), 157
<code>id</code> (<code>enterprise.models.EnterpriseCustomerUser</code> attribute), 180	<code>ignore_warning()</code> (in module <code>enterprise.decorators</code>), 131
<code>id</code> (<code>enterprise.models.EnterpriseEnrollmentSource</code> attribute), 183	<code>ImmutableStateSerializer</code> (class in <code>enterprise.api.v1.serializers</code>), 91
<code>id</code> (<code>enterprise.models.EnterpriseFeatureRole</code> attribute), 184	<code>inactivate_other_customers()</code> (<code>enterprise.models.EnterpriseCustomerUser</code> class method), 180
<code>id</code> (<code>enterprise.models.EnterpriseFeatureUserRoleAssignment</code> attribute), 185	<code>include_date</code> (<code>enterprise.models.EnterpriseCustomerReportingConfiguration</code> attribute), 172
<code>id</code> (<code>enterprise.models.HistoricalEnrollmentNotificationEmailTemplate</code> attribute), 192	<code>include_exec_ed_2u_courses</code> (<code>enterprise.models.EnterpriseCatalogQuery</code> attribute), 143
<code>id</code> (<code>enterprise.models.HistoricalEnterpriseCourseEnrollment</code> attribute), 194	<code>inlines</code> (<code>enterprise.admin.EnterpriseCustomerAdmin</code> attribute), 47
<code>id</code> (<code>enterprise.models.HistoricalEnterpriseCourseEntitlement</code> attribute), 197	<code>input_type</code> (<code>enterprise.admin.widgets.SubmitInput</code> attribute), 42
<code>id</code> (<code>enterprise.models.HistoricalEnterpriseCustomerUser</code> attribute), 214	<code>instance</code> (<code>enterprise.models.HistoricalEnrollmentNotificationEmailTemplate</code> property), 192
<code>id</code> (<code>enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</code> attribute), 221	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCourseEnrollment</code> property), 194
<code>id</code> (<code>enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment</code> attribute), 224	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCourseEntitlement</code> property), 197
<code>id</code> (<code>enterprise.models.HistoricalPendingEnrollment</code> attribute), 226	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCustomer</code> property), 202
<code>id</code> (<code>enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser</code> attribute), 229	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCustomerCatalog</code> property), 205
<code>id</code> (<code>enterprise.models.HistoricalPendingEnterpriseCustomerUser</code> attribute), 230	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCustomerInviteKey</code> property), 207
<code>id</code> (<code>enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment</code> attribute), 232	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration</code> property), 210
<code>id</code> (<code>enterprise.models.LearnerCreditEnterpriseCourseEnrollment</code> attribute), 234	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseCustomerUser</code> property), 214
<code>id</code> (<code>enterprise.models.LicensedEnterpriseCourseEnrollment</code> attribute), 236	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseGroup</code> property), 216
<code>id</code> (<code>enterprise.models.PendingEnrollment</code> attribute), 237	<code>instance</code> (<code>enterprise.models.HistoricalEnterpriseGroupMembership</code> property), 218
<code>id</code> (<code>enterprise.models.PendingEnterpriseCustomerAdminUser</code> attribute), 239	<code>instance</code> (<code>enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment</code> property), 218
<code>id</code> (<code>enterprise.models.PendingEnterpriseCustomerUser</code> attribute), 240	
<code>id</code> (<code>enterprise.models.SystemWideEnterpriseRole</code> attribute), 241	
<code>id</code> (<code>enterprise.models.SystemWideEnterpriseUserRoleAssignment</code> attribute), 243	
<code>identity_provider</code>	

261
 is_course_run_available_for_enrollment() (in module *enterprise.utils*), 261
 is_course_run_enrollable() (in module *enterprise.utils*), 261
 is_course_run_id() (in module *enterprise.views.GrantDataSharingPermissions* method), 269
 is_course_run_published() (in module *enterprise.utils*), 262
 is_course_run_upgradeable() (in module *enterprise.utils*), 262
 is_enrolled() (in module *enterprise.api_client.lms.EnrollmentApiClient* method), 109
 is_pending_configuration() (in module *enterprise.models.EnterpriseCustomerSsoConfiguration* method), 175
 is_pending_user() (in module *enterprise.utils*), 262
 is_read (in module *enterprise.models.AdminNotificationRead* attribute), 137
 is_relinkable (in module *enterprise.models.EnterpriseCustomerUser* attribute), 180
 is_relinkable (in module *enterprise.models.HistoricalEnterpriseCustomerUser* attribute), 214
 is_removed (in module *enterprise.models.EnterpriseCustomerInviteKey* attribute), 168
 is_removed (in module *enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 175
 is_removed (in module *enterprise.models.EnterpriseGroup* attribute), 188
 is_removed (in module *enterprise.models.EnterpriseGroupMembership* attribute), 190
 is_removed (in module *enterprise.models.HistoricalEnterpriseCustomerInviteKey* attribute), 207
 is_removed (in module *enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 211
 is_removed (in module *enterprise.models.HistoricalEnterpriseGroup* attribute), 216
 is_removed (in module *enterprise.models.HistoricalEnterpriseGroupMembership* attribute), 219
 is_revoked (in module *enterprise.models.EnterpriseFulfillmentSource* attribute), 187
 is_revoked (in module *enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment* attribute), 222
 is_revoked (in module *enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment* attribute), 224
 is_revoked (in module *enterprise.models.LearnerCreditEnterpriseCourseEnrollment* attribute), 234
 is_revoked (in module *enterprise.models.LicensedEnterpriseCourseEnrollment* attribute), 236
 is_user_enrolled() (in module *enterprise.utils*), 262
 is_user_linked() (in module *enterprise.admin.forms.ManageLearnersDataSharingConsentForm* method), 32
 is_valid (in module *enterprise.models.EnterpriseCustomerInviteKey* property), 168
 is_valid_url() (in module *enterprise.utils*), 262
 IsInEnterpriseGroup (in module *enterprise.api.v1.permissions*), 80

J

js (in module *enterprise.admin.EnterpriseCustomerCatalogAdmin* attribute), 47
 json_serialized_course_modes() (in module *enterprise.constants*), 129

L

last_name_attribute (in module *enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 175
 last_name_attribute (in module *enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 211
 LEARNER_CREDIT (in module *enterprise.constants.FulfillmentTypes* attribute), 129
 LEARNER_ENGAGEMENT (in module *enterprise.models.ChatGPTResponse* attribute), 139
 LEARNER_PROGRESS (in module *enterprise.models.ChatGPTResponse* attribute), 139
 learner_role() (in module *enterprise.roles_api*), 244
 LearnerCreditEnterpriseCourseEnrollment (in module *enterprise.models*), 233
 LearnerCreditEnterpriseCourseEnrollment.DoesNotExist, 233
 LearnerCreditEnterpriseCourseEnrollment.MultipleObjectsReturned, 233
 learnercreditenterpriseourseenrollment_enrollment_fulfillment (in module *enterprise.models.EnterpriseCourseEnrollment* attribute), 145
 learnercreditenterpriseourseenrollment_entitlement_fulfillment (in module *enterprise.models.EnterpriseCourseEntitlement* attribute), 147
 LearnerCreditEnterpriseCourseEnrollmentReadOnlySerializer (in module *enterprise.api.v1.serializers*), 91
 LearnerCreditEnterpriseCourseEnrollmentReadOnlySerializer (in module *enterprise.api.v1.serializers*), 91
 LEARNERS (in module *enterprise.admin.views.EnterpriseCustomerManageLearnersView* attribute), 38
 LICENSE (in module *enterprise.constants.FulfillmentTypes* attribute), 129
 license (in module *enterprise.models.EnterpriseCourseEnrollment* property), 145

[license_revoke\(\)](#) (enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseCourseEnrollmentViewSet method), 76

[license_uuid](#) (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollmentEnterpriseCustomerInviteKey attribute), 224

[license_uuid](#) (enterprise.models.HistoricalPendingEnrollmentLinkLearnersSerializer (class in enterprise.api.v1.serializers)), 226

[license_uuid](#) (enterprise.models.LicensedEnterpriseCourseEnrollmentEnterpriseCustomerInviteKey attribute), 236

[license_uuid](#) (enterprise.models.PendingEnrollmentLinkLearners attribute), 237

[license_uuid\(\)](#) (enterprise.admin.EnterpriseCourseEnrollmentAdmin method), 44

[licensed_enterprise_course_enrollment_model\(\)](#) (in module enterprise.utils), 262

[LicensedEnterpriseCourseEnrollment](#) (class in enterprise.models), 235

[LicensedEnterpriseCourseEnrollment.DoesNotExist](#), 235

[LicensedEnterpriseCourseEnrollment.MultipleObjectsReturned](#), 235

[licensedenterpriseenrollment_enrollment_fulfillment](#) (enterprise.models.EnterpriseCourseEnrollment attribute), 145

[licensedenterpriseenrollment_entitlement_fulfillment](#) (enterprise.models.EnterpriseCourseEntitlement attribute), 147

[LicensedEnterpriseCourseEnrollmentReadOnlySerialized](#) (class in enterprise.api.v1.serializers), 91

[LicensedEnterpriseCourseEnrollmentReadOnlySerializedMeta](#) (class in enterprise.api.v1.serializers), 91

[LicensedEnterpriseCourseEnrollmentViewSet](#) (class in enterprise.api.v1.views.enterprise_subsidy_fulfillment), 75

[LicensedEnterpriseCourseEnrollmentViewSet.EnrollmentActionStatus](#) (class in enterprise.api.v1.views.enterprise_subsidy_fulfillment), 75

[link_learners\(\)](#) (enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserEnterpriseAdminViewSet method), 78

[link_pending_enterprise_user\(\)](#) (enterprise.models.PendingEnterpriseCustomerUser method), 240

[link_to_modal\(\)](#) (in module enterprise.template_tags.enterprise), 128

[link_user\(\)](#) (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerAdminUserViewSet method), 68

[link_user\(\)](#) (enterprise.models.EnterpriseCustomerUserViewSet method), 181

[linked](#) (enterprise.models.EnterpriseCustomerUser attribute), 180

[linked](#) (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 180

[linked_enterprise_customer_users](#) (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerAdminUserViewSet method), 68

[list\(\)](#) (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogAdmin method), 66

[list\(\)](#) (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyAdmin method), 68

[list\(\)](#) (enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingAdmin method), 69

[list\(\)](#) (enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationAdmin method), 70

[list_display](#) (enterprise.admin.AdminNotificationAdmin attribute), 42

[list_display](#) (enterprise.admin.AdminNotificationFilterAdmin attribute), 42

[list_display](#) (enterprise.admin.AdminNotificationReadAdmin attribute), 42

[list_display](#) (enterprise.admin.ChatGPTResponseAdmin attribute), 43

[list_display](#) (enterprise.admin.EnterpriseCatalogQueryAdmin attribute), 44

[list_display](#) (enterprise.admin.EnterpriseCourseEnrollmentAdmin attribute), 44

[list_display](#) (enterprise.admin.EnterpriseCustomerAdmin attribute), 47

[list_display](#) (enterprise.admin.EnterpriseCustomerCatalogAdmin attribute), 48

[list_display](#) (enterprise.admin.EnterpriseCustomerInviteKeyAdmin attribute), 49

[list_display](#) (enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin attribute), 50

[list_display](#) (enterprise.admin.EnterpriseCustomerSsoConfigurationAdmin attribute), 50

[list_display](#) (enterprise.admin.EnterpriseCustomerTypeAdmin attribute), 50

[list_display](#) (enterprise.admin.EnterpriseCustomerUserAdmin attribute), 51

[list_display](#) (enterprise.admin.EnterpriseGroupAdmin attribute), 52

[list_display](#) (enterprise.admin.EnterpriseGroupMembershipAdmin attribute), 52

[list_display](#) (enterprise.admin.PendingEnrollmentAdmin attribute), 53

[list_display](#) (enterprise.admin.EnterpriseCourseEnrollmentAdmin attribute), 53

[list_display](#) (enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin attribute), 54

[list_filter](#) (enterprise.admin.ChatGPTResponseAdmin attribute), 43

list_filter(*enterprise.admin.EnterpriseCustomerAdmin* *ManageLearnersForm.CsvColumns* (class in *enterprise.admin.forms*), 32
 list_filter(*enterprise.admin.EnterpriseCustomerInviteAdmin* *ManageLearnersForm.Fields* (class in *enterprise.admin.forms*), 32
 list_filter(*enterprise.admin.EnterpriseCustomerReportingAdmin* *ManageLearnersForm.Modes* (class in *enterprise.admin.forms*), 33
 list_filter(*enterprise.admin.EnterpriseGroupAdmin* *ManageLearnersForm.NotificationTypes* (class in *enterprise.admin.forms*), 33
 list_per_page (enter- *MANAGEMENT_COMMAND* (enter-
prise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin *prise.models.EnterpriseEnrollmentSource*
 attribute), 54 attribute), 182
 list_select_related (enter- *MANUAL* (*enterprise.models.EnterpriseEnrollmentSource*
prise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin attribute), 182
 attribute), 54 *MANUAL_REPORTS* (enter-
 list_serializer_class (enter- *prise.models.EnterpriseCustomerReportingConfiguration*
prise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer *Meta*
 attribute), 86 *mark_configured()* (enter-
 LMSNotAvailable, 115 *prise.admin.EnterpriseCustomerSsoConfigurationAdmin*
 localized_utcnow() (in module *enterprise.utils*), 262 method), 50
 logo(*enterprise.models.EnterpriseCustomerBrandingConfiguration* *marked_authored* (enter-
 attribute), 162 *prise.models.EnterpriseCustomerSsoConfiguration*
 logo_path() (in module *enterprise.utils*), 262 attribute), 175
 lookup_field(*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* *generatePatchSerializer.Meta* (enter-
 attribute), 83 *prise.models.HistoricalEnterpriseCustomerSsoConfiguration*
 lookup_field(*enterprise.api.v1.serializers.EnterpriseCustomerApiClientSerializer* *Meta*
 attribute), 83 *MAX_RETRIES* (*enterprise.api_client.lms.GradesApiClient*
 lookup_field(*enterprise.api.v1.views.enterprise_customer_branding_configuration* *EnterpriseCustomerBrandingConfigurationView*
 attribute), 65 *media* (*enterprise.admin.AdminNotificationAdmin* prop-
 lookup_field(*enterprise.api.v1.views.enterprise_customer_reporting_configuration* *EnterpriseCustomerReportingConfigurationViewSet*
 attribute), 69 *media* (*enterprise.admin.AdminNotificationFilterAdmin*
 property), 42
media (*enterprise.admin.AdminNotificationReadAdmin*
 property), 42
media (*enterprise.admin.ChatGPTResponseAdmin* prop-
 erty), 43
media (*enterprise.admin.EnrollmentNotificationEmailTemplateAdmin*
 property), 43
media (*enterprise.admin.EnterpriseCatalogQueryAdmin*
 property), 44
media (*enterprise.admin.EnterpriseCourseEnrollmentAdmin*
 property), 45
media (*enterprise.admin.EnterpriseCustomerAdmin*
 property), 47
media (*enterprise.admin.EnterpriseCustomerBrandingConfigurationInline*
 property), 47
media (*enterprise.admin.EnterpriseCustomerCatalogAdmin*
 property), 48
media (*enterprise.admin.EnterpriseCustomerCatalogInline*
 property), 48
media (*enterprise.admin.EnterpriseCustomerIdentityProviderInline*
 property), 49
media (*enterprise.admin.EnterpriseCustomerInviteKeyAdmin*
 property), 49
media (*enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin*
 property), 49

M

MANAGE_LEARNERS (*enterprise.admin.utils.UrlNames* at-
 tribute), 35
 manage_learners() (enter-
prise.admin.EnterpriseCustomerAdmin
 method), 47
 manage_learners_data_sharing_consent() (en-
terprise.admin.EnterpriseCustomerAdmin
 method), 47
MANAGE_LEARNERS_DSC (enter-
prise.admin.utils.UrlNames attribute), 35
MANAGE_LEARNERS_DSC_FORM (enter-
prise.admin.views.EnterpriseCustomerManageLearnersForm *EnterpriseCustomerBrandingConfigurationInline*
 attribute), 37 property), 47
MANAGE_LEARNERS_FORM (enter-
prise.admin.views.EnterpriseCustomerManageLearnersView *PROXY_PARAMETERS*
 attribute), 38 *media* (*enterprise.admin.EnterpriseCustomerCatalogInline*
 property), 48
 ManageLearnersDataSharingConsentForm (class in
enterprise.admin.forms), 31
 ManageLearnersDataSharingConsentForm.Fields
 (class in *enterprise.admin.forms*), 31
 ManageLearnersForm (class in *enterprise.admin.forms*),
 32

property), 50

media (enterprise.admin.EnterpriseCustomerSsoConfigurationAdmin property), 50

media (enterprise.admin.EnterpriseCustomerTypeAdmin property), 50

media (enterprise.admin.EnterpriseCustomerUserAdmin property), 51

media (enterprise.admin.EnterpriseFeatureUserRoleAssignmentAdmin property), 51

media (enterprise.admin.EnterpriseGroupAdmin property), 52

media (enterprise.admin.EnterpriseGroupMembershipAdmin property), 52

media (enterprise.admin.forms.AdminNotificationForm property), 26

media (enterprise.admin.forms.BulkCatalogQueryUpdateCommandConfigurationAdmin property), 26

media (enterprise.admin.forms.EnterpriseCustomerAdminForm property), 28

media (enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm property), 29

media (enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm property), 30

media (enterprise.admin.forms.EnterpriseCustomerReportingConfigurationAdminForm property), 31

media (enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentAdminForm property), 31

media (enterprise.admin.forms.ManageLearnersDataSharingConfigurationAdminForm property), 32

media (enterprise.admin.forms.ManageLearnersForm property), 34

media (enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentAdminForm property), 34

media (enterprise.admin.forms.TransmitEnterpriseCoursesForm property), 35

media (enterprise.admin.PendingEnrollmentAdmin property), 53

media (enterprise.admin.PendingEnterpriseCustomerAdmin property), 53

media (enterprise.admin.PendingEnterpriseCustomerAdminForm property), 53

media (enterprise.admin.PendingEnterpriseCustomerUserAdmin property), 54

media (enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdminForm property), 54

media (enterprise.admin.widgets.SubmitInput property), 42

media (enterprise.forms.EnterpriseLoginForm property), 132

media (enterprise.forms.EnterpriseSelectionForm property), 132

member_email (enterprise.models.EnterpriseGroupMembership attribute), 190

members (enterprise.models.EnterpriseGroup attribute), 188

members() (enterprise.admin.EnterpriseGroupAdmin method), 52

membership_user (enterprise.models.EnterpriseGroupMembership attribute), 190

memberships (enterprise.models.EnterpriseCustomerUser attribute), 180

memberships (enterprise.models.PendingEnterpriseCustomerUser attribute), 240

message (enterprise.api.v1.permissions.IsInEnterpriseGroup attribute), 80

message (enterprise.heartbeat.exceptions.ServiceNotAvailable attribute), 115

metadata_url (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 211

metadata_url (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

metadata_xml (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 175

metadata_xml (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

MISSING_EXPECTED_COLUMNS (enterprise.admin.ValidationMessages attribute), 249

MISSING_REASON (enterprise.utils.ValidationMessages attribute), 249

MISSING_REQUIRED_PARAMS_MSG (enterprise.api.v1.views.coupon_codes.CouponCodesView attribute), 56

MISSING_REQUIRED_PARAMS_MSG (enterprise.api.v1.views.notifications.NotificationReadView attribute), 77

MODE (enterprise.admin.forms.ManageLearnersForm.Fields attribute), 32

mode (enterprise.models.EnterpriseCourseEnrollment property), 145

MODE_PLURAL (enterprise.admin.forms.ManageLearnersForm.Modes attribute), 33

MODE_SINGULAR (enterprise.admin.forms.ManageLearnersForm.Modes attribute), 33

model (enterprise.admin.AdminNotificationAdmin attribute), 42

model (enterprise.admin.AdminNotificationFilterAdmin attribute), 42

model (enterprise.admin.AdminNotificationReadAdmin attribute), 42

model (enterprise.admin.ChatGPTResponseAdmin attribute), 43

model (enterprise.admin.EnrollmentNotificationEmailTemplateAdmin.Meta attribute), 43

model (enterprise.admin.EnterpriseCatalogQueryAdmin.Meta attribute), 44

<code>model (enterprise.admin.EnterpriseCourseEnrollmentAdmin.Meta attribute), 44</code>	<code>model (enterprise.api.v1.serializers.BaseEnterpriseCustomerInviteKeySerializer attribute), 81</code>
<code>model (enterprise.admin.EnterpriseCustomerAdmin.Meta attribute), 45</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCatalogQuerySerializer.Meta attribute), 82</code>
<code>model (enterprise.admin.EnterpriseCustomerBrandingConfigurationAdmin.Meta attribute), 47</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCourseEnrollmentReadOnlySerializer attribute), 82</code>
<code>model (enterprise.admin.EnterpriseCustomerCatalogAdmin.Meta attribute), 48</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWithAdditionSerializer attribute), 82</code>
<code>model (enterprise.admin.EnterpriseCustomerCatalogInlineAdmin.Meta attribute), 48</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWriteSerializer attribute), 83</code>
<code>model (enterprise.admin.EnterpriseCustomerIdentityProviderAdmin.Meta attribute), 49</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerBasicSerializer.Meta attribute), 83</code>
<code>model (enterprise.admin.EnterpriseCustomerInviteKeyAdmin.Meta attribute), 49</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerBrandingConfigurationAdmin.Meta attribute), 84</code>
<code>model (enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin.Meta attribute), 49</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerCatalogSerializer.Meta attribute), 85</code>
<code>model (enterprise.admin.EnterpriseCustomerSsoConfigurationAdmin.Meta attribute), 50</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerCatalogWriteOnlySerializer attribute), 85</code>
<code>model (enterprise.admin.EnterpriseCustomerTypeAdmin.Meta attribute), 50</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerIdentityProviderSerializer attribute), 86</code>
<code>model (enterprise.admin.EnterpriseCustomerUserAdmin.Meta attribute), 51</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyPartialUpdateSerializer attribute), 86</code>
<code>model (enterprise.admin.EnterpriseFeatureUserRoleAssignmentAdmin.Meta attribute), 51</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerReportingConfigurationAdmin.Meta attribute), 87</code>
<code>model (enterprise.admin.EnterpriseGroupAdmin attribute), 52</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerSerializer.Meta attribute), 88</code>
<code>model (enterprise.admin.EnterpriseGroupMembershipAdmin.Meta attribute), 52</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerSsoConfigurationAdmin.Meta attribute), 89</code>
<code>model (enterprise.admin.forms.AdminNotificationForm.Meta attribute), 26</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerUserReadOnlySerializer attribute), 89</code>
<code>model (enterprise.admin.forms.EnterpriseCustomerAdminForm.Meta attribute), 27</code>	<code>model (enterprise.api.v1.serializers.EnterpriseCustomerUserWriteSerializer attribute), 90</code>
<code>model (enterprise.admin.forms.EnterpriseCustomerCatalogAdminForm.Meta attribute), 29</code>	<code>model (enterprise.api.v1.serializers.EnterpriseGroupMembershipSerializer attribute), 91</code>
<code>model (enterprise.admin.forms.EnterpriseCustomerIdentityProviderAdminForm.Meta attribute), 29</code>	<code>model (enterprise.api.v1.serializers.EnterpriseGroupSerializer.Meta attribute), 91</code>
<code>model (enterprise.admin.forms.EnterpriseCustomerReportingConfigurationAdminForm.Meta attribute), 30</code>	<code>model (enterprise.api.v1.serializers.LearnerCreditEnterpriseCourseEnrollmentSerializer attribute), 91</code>
<code>model (enterprise.admin.forms.EnterpriseFeatureUserRoleAssignmentAdminForm.Meta attribute), 31</code>	<code>model (enterprise.api.v1.serializers.LicensedEnterpriseCourseEnrollmentReadOnlySerializer attribute), 92</code>
<code>model (enterprise.admin.forms.SystemWideEnterpriseUserRoleAssignmentAdminForm.Meta attribute), 34</code>	<code>model (enterprise.api.v1.serializers.PendingEnterpriseCustomerUserSerializer attribute), 92</code>
<code>model (enterprise.admin.PendingEnrollmentAdmin.Meta attribute), 52</code>	<code>model (enterprise.api.v1.serializers.SiteSerializer.Meta attribute), 93</code>
<code>model (enterprise.admin.PendingEnterpriseCustomerAdmin.Meta attribute), 53</code>	<code>model (enterprise.api.v1.serializers.UserSerializer.Meta attribute), 93</code>
<code>model (enterprise.admin.PendingEnterpriseCustomerAdmin.Meta attribute), 53</code>	<code>model (enterprise.models.AdminNotification attribute), 135</code>
<code>model (enterprise.admin.PendingEnterpriseCustomerUserAdmin.Meta attribute), 54</code>	<code>model (enterprise.models.AdminNotificationFilter attribute), 136</code>
<code>model (enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin.Meta attribute), 54</code>	<code>model (enterprise.models.AdminNotificationRead attribute), 138</code>
<code>model (enterprise.api.v1.serializers.AdminNotificationSerializer attribute), 80</code>	<code>model (enterprise.models.ChatGPTResponse attribute), 140</code>

`modified (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 141`
`modified (enterprise.models.EnterpriseCatalogQuery attribute), 143`
`modified (enterprise.models.EnterpriseCourseEnrollment attribute), 145`
`modified (enterprise.models.EnterpriseCourseEntitlement attribute), 147`
`modified (enterprise.models.EnterpriseCustomer attribute), 158`
`modified (enterprise.models.EnterpriseCustomerBranding attribute), 162`
`modified (enterprise.models.EnterpriseCustomerCatalog attribute), 165`
`modified (enterprise.models.EnterpriseCustomerIdentityProvider attribute), 167`
`modified (enterprise.models.EnterpriseCustomerInviteKey attribute), 169`
`modified (enterprise.models.EnterpriseCustomerReporting attribute), 172`
`modified (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 175`
`modified (enterprise.models.EnterpriseCustomerType attribute), 177`
`modified (enterprise.models.EnterpriseCustomerUser attribute), 180`
`modified (enterprise.models.EnterpriseEnrollmentSource attribute), 183`
`modified (enterprise.models.EnterpriseFeatureRole attribute), 184`
`modified (enterprise.models.EnterpriseFeatureUserRoleAssignment attribute), 185`
`modified (enterprise.models.EnterpriseFulfillmentSource attribute), 187`
`modified (enterprise.models.EnterpriseGroup attribute), 188`
`modified (enterprise.models.EnterpriseGroupMembership attribute), 190`
`modified (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 192`
`modified (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 195`
`modified (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 197`
`modified (enterprise.models.HistoricalEnterpriseCustomer attribute), 202`
`modified (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 205`
`modified (enterprise.models.HistoricalEnterpriseCustomerInviteKey attribute), 207`
`modified (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211`
`modified (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 214`
`modified (enterprise.models.HistoricalEnterpriseGroup attribute), 216`
`modified (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 219`
`modified (enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 222`
`modified (enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment attribute), 224`
`modified (enterprise.models.HistoricalPendingEnrollment attribute), 226`
`modified (enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser attribute), 229`
`modified (enterprise.models.HistoricalPendingEnterpriseCustomerUser attribute), 231`
`modified (enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment attribute), 233`
`modified (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 234`
`modified (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 236`
`modified (enterprise.models.PendingEnrollment attribute), 237`
`modified (enterprise.models.PendingEnterpriseCustomerAdminUser attribute), 239`
`modified (enterprise.models.PendingEnterpriseCustomerUser attribute), 240`
`modified (enterprise.models.SystemWideEnterpriseRole attribute), 241`
`modified (enterprise.models.SystemWideEnterpriseUserRoleAssignment attribute), 243`
`module enterprise, 273`
`enterprise.admin, 42`
`enterprise.admin.actions, 25`
`enterprise.admin.forms, 25`
`enterprise.admin.paginator, 35`
`enterprise.admin.utils, 35`
`enterprise.admin.views, 37`
`enterprise.admin.widgets, 42`
`enterprise.api, 98`
`enterprise.api.filters, 94`
`enterprise.api.pagination, 95`
`enterprise.api.throttles, 96`
`enterprise.api.urls, 97`
`enterprise.api.utils, 97`
`enterprise.api.v1, 94`
`enterprise.api.v1.decorators, 79`
`enterprise.api.v1.fields, 80`
`enterprise.api.v1.permissions, 80`
`enterprise.api.v1.serializers, 80`
`enterprise.api.v1.urls, 94`
`enterprise.api.v1.views, 79`
`enterprise.api.v1.views.analytics_summary, 55`

[enterprise.api.v1.views.base_views](#), 55
[enterprise.api.v1.views.coupon_codes](#), 56
[enterprise.api.v1.views.enterprise_catalog_queries](#), 57
[enterprise.api.v1.views.enterprise_course_enrollment](#), 58
[enterprise.api.v1.views.enterprise_customer](#), 60
[enterprise.api.v1.views.enterprise_customer_apartment_details](#), 62
[enterprise.api.v1.views.enterprise_customer_branding_configuration](#), 64
[enterprise.api.v1.views.enterprise_customer_catalog](#), 65
[enterprise.api.v1.views.enterprise_customer_inventories](#), 68
[enterprise.api.v1.views.enterprise_customer_reportings](#), 69
[enterprise.api.v1.views.enterprise_customer_sso_configuration](#), 70
[enterprise.api.v1.views.enterprise_customer_users](#), 71
[enterprise.api.v1.views.enterprise_group](#), 72
[enterprise.api.v1.views.enterprise_subsidy_fulfillment](#), 74
[enterprise.api.v1.views.notifications](#), 77
[enterprise.api.v1.views.pending_enterprise_customer](#), 78
[enterprise.api.v1.views.plotly_auth](#), 79
[enterprise.api_client](#), 113
[enterprise.api_client.braze](#), 98
[enterprise.api_client.client](#), 99
[enterprise.api_client.discovery](#), 100
[enterprise.api_client.ecommerce](#), 103
[enterprise.api_client.enterprise](#), 103
[enterprise.api_client.enterprise_catalog](#), 104
[enterprise.api_client.lms](#), 106
[enterprise.api_client.open_ai](#), 112
[enterprise.api_client.sso_orchestrator](#), 112
[enterprise.apps](#), 128
[enterprise.config](#), 114
[enterprise.config.models](#), 113
[enterprise.constants](#), 129
[enterprise.decorators](#), 130
[enterprise.errors](#), 131
[enterprise.forms](#), 132
[enterprise.heartbeat](#), 117
[enterprise.heartbeat.checks](#), 114
[enterprise.heartbeat.exceptions](#), 115
[enterprise.heartbeat.throttles](#), 116
[enterprise.heartbeat.utils](#), 116
[enterprise.heartbeat.views](#), 116
[enterprise.logging](#), 132
[enterprise.management](#), 127
[enterprise.management.commands](#), 127
[enterprise.management.commands.add_executed_exclusion_token](#), 117
[enterprise.management.commands.backfill_learner_role_assignments](#), 117
[enterprise.management.commands.bulk_update_catalog_queries](#), 118
[enterprise.management.commands.compare_discovery_and_enrollment](#), 118
[enterprise.management.commands.create_enterprise_courses](#), 118
[enterprise.management.commands.create_missing_dsc_records](#), 119
[enterprise.management.commands.email_drip_for_missing_reports](#), 119
[enterprise.management.commands.ensure_singular_active_users](#), 120
[enterprise.management.commands.fix_dsc_records](#), 120
[enterprise.management.commands.manufacture_data](#), 120
[enterprise.management.commands.migrate_enterprise_catalog_queries](#), 122
[enterprise.management.commands.monthly_impact_report](#), 122
[enterprise.management.commands.new_user](#), 123
[enterprise.management.commands.nudge_dormant_enrolled_users](#), 123
[enterprise.management.commands.reencrypt_enterprise_customer_data](#), 123
[enterprise.management.commands.remove_expired_pending_enrollments](#), 123
[enterprise.management.commands.revert_enrollment_objects](#), 124
[enterprise.management.commands.save_enterprise_customer_data](#), 124
[enterprise.management.commands.seed_enterprise_devstack](#), 125
[enterprise.management.commands.unlink_enterprise_customer_data](#), 125
[enterprise.management.commands.update_config_last_error](#), 126
[enterprise.management.commands.update_role_assignments](#), 126
[enterprise.messages](#), 133
[enterprise.middleware](#), 134
[enterprise.models](#), 134
[enterprise.roles_api](#), 244
[enterprise.rules](#), 245
[enterprise.settings](#), 127
[enterprise.settings.test](#), 127
[enterprise.signals](#), 245

enterprise.tasks, 246
 enterprise.template_tags, 128
 enterprise.template_tags.enterprise, 127
 enterprise.toggles, 247
 enterprise.tpa_pipeline, 247
 enterprise.urls, 248
 enterprise.utils, 248
 enterprise.validators, 266
 enterprise.views, 267
 moodleenterpriseconfiguration_set
 (enterprise.models.EnterpriseCustomer attribute), 158
 MOVED_TO_AUDIT (enterprise.api.v1.views.enterprise_subsidy_fulfillment.
 EnterpriseCustomer attribute), 75
N
 name (enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryViewSet
 attribute), 57
 name (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet
 attribute), 59
 name (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet
 attribute), 62
 name (enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentialsViewSet
 attribute), 63
 name (enterprise.api.v1.views.enterprise_customer_api_credentials.EnterpriseCustomerApiCredentialsViewSet
 attribute), 64
 name (enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationViewSet
 attribute), 65
 name (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet
 attribute), 66
 name (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet
 attribute), 67
 name (enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet
 attribute), 68
 name (enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingConfigurationViewSet
 attribute), 69
 name (enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationViewSet
 attribute), 70
 name (enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserViewSet
 attribute), 71
 name (enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet
 attribute), 73
 name (enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet
 attribute), 75
 name (enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet
 attribute), 76
 name (enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserEnterpriseAdminViewSet
 attribute), 78
 name (enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserEnterpriseAdminViewSet
 attribute), 78
 name (enterprise.apps.EnterpriseConfig attribute), 128
 name (enterprise.models.EnterpriseCustomer attribute), 158
 name (enterprise.models.EnterpriseCustomerType attribute), 177
 name (enterprise.models.EnterpriseCustomerUser property), 180
 name (enterprise.models.EnterpriseEnrollmentSource attribute), 183
 name (enterprise.models.EnterpriseFeatureRole attribute), 184
 name (enterprise.models.EnterpriseGroup attribute), 188
 name (enterprise.models.HistoricalEnterpriseCustomer attribute), 202
 name (enterprise.models.HistoricalEnterpriseGroup attribute), 216
 name (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 241
 name (enterprise.models.HistoricalEnterpriseCourseEntitlement attribute), 241
 next_record (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate
 property), 192
 next_record (enterprise.models.HistoricalEnterpriseCourseEnrollment
 property), 195
 next_record (enterprise.models.HistoricalEnterpriseCourseEntitlement
 property), 197
 next_record (enterprise.models.HistoricalEnterpriseCustomer
 property), 202
 next_record (enterprise.models.HistoricalEnterpriseCustomerCatalog
 property), 205
 next_record (enterprise.models.HistoricalEnterpriseCustomerInviteKey
 property), 208
 next_record (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration
 property), 211
 next_record (enterprise.models.HistoricalEnterpriseCustomerUser
 property), 214
 next_record (enterprise.models.HistoricalEnterpriseGroup
 property), 216
 next_record (enterprise.models.HistoricalEnterpriseGroupMembership
 property), 219
 next_record (enterprise.models.HistoricalLearnerCreditEnterpriseCourse
 property), 222
 next_record (enterprise.models.HistoricalLearnerCreditEnterpriseCourse
 property), 224
 next_record (enterprise.models.HistoricalPendingEnrollment
 property), 226
 next_record (enterprise.models.HistoricalPendingEnterpriseCustomerAd
 property), 229
 next_record (enterprise.models.HistoricalPendingEnterpriseCustomerUs
 property), 231
 next_record (enterprise.models.HistoricalSystemWideEnterpriseUserRole
 property), 233
 NO_FIELDS_SPECIFIED (enterprise.models.HistoricalEnterpriseCustomerUserEnterpriseAdminViewSet
 attribute), 249
 NO_ID_PROFESSIONAL (enterprise.constants.CourseModes attribute), 129
 NO_NOTIFICATION (enterprise.constants.CourseModes attribute), 129

- prise.admin.forms.ManageLearnersForm.Notification* (attribute), 33
- NoAuthAPIClient* (class in *enterprise.api_client.client*), 99
- NoAuthDiscoveryClient* (class in *enterprise.api_client.discovery*), 102
- NoAuthEcommerceClient* (class in *enterprise.api_client.ecommerce*), 103
- NoAuthEnterpriseCatalogClient* (class in *enterprise.api_client.enterprise_catalog*), 106
- NoAuthLMSClient* (class in *enterprise.api_client.lms*), 111
- Node* (class in *enterprise.management.commands.manufacture*), 121
- NonAtomicView* (class in *enterprise.views*), 270
- NOT_AUTHORIZED_ERROR* (*enterprise.api.v1.serializers.LinkLearnersSerializer* attribute), 92
- NotConnectedToOpenEdX*, 248
- notification_filter* (*enterprise.models.AdminNotificationFilter* attribute), 136
- NotificationReadView* (class in *enterprise.api.v1.views.notifications*), 77
- NOTIFY* (*enterprise.admin.forms.ManageLearnersForm.Fields* attribute), 33
- notify_enrolled_learners()* (*enterprise.models.EnterpriseCustomer* method), 158
- null_decorator()* (in module *enterprise.decorators*), 131
- ## O
- oauth_orchestration_complete()* (*enterprise.api.v1.views.enterprise_customer_sso_configuration* method), 70
- oauth_user_id* (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 175
- oauth_user_id* (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 211
- objects* (*enterprise.models.AdminNotification* attribute), 135
- objects* (*enterprise.models.AdminNotificationFilter* attribute), 136
- objects* (*enterprise.models.AdminNotificationRead* attribute), 138
- objects* (*enterprise.models.ChatGPTResponse* attribute), 140
- objects* (*enterprise.models.EnrollmentNotificationEmailTemplate* attribute), 141
- objects* (*enterprise.models.EnterpriseCatalogQuery* attribute), 143
- objects* (*enterprise.models.EnterpriseCourseEnrollment* attribute), 145
- objects* (*enterprise.models.EnterpriseCourseEntitlement* attribute), 148
- objects* (*enterprise.models.EnterpriseCustomer* attribute), 158
- objects* (*enterprise.models.EnterpriseCustomerBrandingConfiguration* attribute), 162
- objects* (*enterprise.models.EnterpriseCustomerCatalog* attribute), 165
- objects* (*enterprise.models.EnterpriseCustomerIdentityProvider* attribute), 167
- objects* (*enterprise.models.EnterpriseCustomerReportingConfiguration* attribute), 172
- objects* (*enterprise.models.EnterpriseCustomerType* attribute), 177
- objects* (*enterprise.models.EnterpriseCustomerUser* attribute), 180
- objects* (*enterprise.models.EnterpriseEnrollmentSource* attribute), 183
- objects* (*enterprise.models.EnterpriseFeatureRole* attribute), 184
- objects* (*enterprise.models.EnterpriseFeatureUserRoleAssignment* attribute), 185
- objects* (*enterprise.models.HistoricalEnrollmentNotificationEmailTemplate* attribute), 193
- objects* (*enterprise.models.HistoricalEnterpriseCourseEnrollment* attribute), 195
- objects* (*enterprise.models.HistoricalEnterpriseCourseEntitlement* attribute), 197
- objects* (*enterprise.models.HistoricalEnterpriseCustomer* attribute), 202
- objects* (*enterprise.models.HistoricalEnterpriseCustomerCatalog* attribute), 205
- objects* (*enterprise.models.HistoricalEnterpriseCustomerInviteKey* attribute), 208
- objects* (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 211
- objects* (*enterprise.models.HistoricalEnterpriseCustomerUser* attribute), 214
- objects* (*enterprise.models.HistoricalEnterpriseGroup* attribute), 216
- objects* (*enterprise.models.HistoricalEnterpriseGroupMembership* attribute), 219
- objects* (*enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment* attribute), 222
- objects* (*enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment* attribute), 224
- objects* (*enterprise.models.HistoricalPendingEnrollment* attribute), 226
- objects* (*enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser* attribute), 229
- objects* (*enterprise.models.HistoricalPendingEnterpriseCustomerUser* attribute), 231

objects (enterprise.models.HistoricalSystemWideEnterpriseRoleAssignment attribute), 233

objects (enterprise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 235

objects (enterprise.models.LicensedEnterpriseCourseEnrollment attribute), 236

objects (enterprise.models.PendingEnrollment attribute), 237

objects (enterprise.models.PendingEnterpriseCustomerAdminUser attribute), 239

objects (enterprise.models.PendingEnterpriseCustomerUser attribute), 240

objects (enterprise.models.SystemWideEnterpriseRole attribute), 242

objects (enterprise.models.SystemWideEnterpriseUserRoleAssignment attribute), 243

odata_api_request_timeout (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 175

odata_api_request_timeout (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

odata_api_root_url (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 175

odata_api_root_url (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

odata_api_timeout_interval (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 176

odata_api_timeout_interval (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

odata_client_id (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 176

odata_client_id (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

odata_company_id (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 176

odata_company_id (enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration attribute), 211

OFFER_REDEMPTION (enterprise.models.EnterpriseEnrollmentSource attribute), 182

OK (enterprise.heartbeat.utils.Status attribute), 116

only_safe_html() (in module enterprise.template_tags.enterprise), 128

openedx_operator_role() (in module enterprise.roles_api), 244

OPTIONAL_IGNORE_ENROLLMENTS_MODIFIED_AFTER_PARAM (enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseCourseEnrollment attribute), 76

OPTIONAL_PARAM_NOTES (enterprise.api.v1.views.coupon_codes.CouponCodesView attribute), 56

OPTIONAL_PARAM_NUMBER_OF_CODES (enterprise.api.v1.views.coupon_codes.CouponCodesView attribute), 56

ordering (enterprise.admin.EnterpriseCustomerAdmin attribute), 47

ordering (enterprise.admin.EnterpriseCustomerCatalogAdmin attribute), 48

ordering (enterprise.admin.EnterpriseCustomerReportingConfiguration attribute), 50

ordering_fields (enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollment attribute), 59

ordering_fields (enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView attribute), 62

ordering_fields (enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfiguration attribute), 65

ordering_fields (enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalog attribute), 66

ordering_fields (enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReporting attribute), 69

ordering_fields (enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUser attribute), 71

ordering_fields (enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUser attribute), 79

other_enrollments() (enterprise.admin.EnterpriseCustomerUserAdmin attribute), 51

P

PACING_FORMAT

(enterprise.views.CourseEnrollmentView attribute), 267

page_range (enterprise.admin.paginator.CustomPaginator property), 35

page_size (enterprise.api.v1.views.enterprise_catalog_query.ExpandDefault attribute), 58

paginated_list() (in module enterprise.admin.utils), 36

pagination_class (enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQuery attribute), 58

pagination_class	(enter-prise.admin.views.EnterpriseCustomerManageLearnersViewSet prise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet attribute), 59	PendingEnrollment (class in enterprise.models), 236
pagination_class	(enter-prise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet attribute), 62	PendingEnrollment.DoesNotExist, 236 PendingEnrollment.MultipleObjectsReturned, 236
pagination_class	(enter-prise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserViewSet attribute), 71	pendingenrollment_set (enter-prise.models.PendingEnterpriseEnrollmentSource attribute), 183
PaginationWithFeatureFlags	(class in enter-prise.api.pagination), 95	pendingenrollment_set (enter-prise.models.PendingEnterpriseCustomerUser attribute), 240
paginator	(enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin, 240 attribute), 54	PendingEnrollmentAdmin (class in enterprise.admin), 52
PaginatorWithOptimizedCount	(class in enter-prise.api.v1.views.enterprise_course_enrollment), 59	PendingEnrollmentAdmin.Meta (class in enter-prise.admin), 52
pairwise()	(in module enter-prise.management.commands.manufacture_data), 121	PendingEnterpriseCustomerAdminUser (class in enter-prise.models), 238 PendingEnterpriseCustomerAdminUser.DoesNotExist, 238
parse_csv()	(in module enterprise.admin.utils), 36	PendingEnterpriseCustomerAdminUser.MultipleObjectsReturned, 238
parse_datetime_handle_invalid()	(in module enterprise.utils), 262	pendingenterprisecustomeradminuser_set (enter-prise.models.EnterpriseCustomer attribute), 158
parse_lms_api_datetime()	(in module enterprise.utils), 262	PendingEnterpriseCustomerAdminUserAdmin (class in enterprise.admin), 53
parser_classes	(enter-prise.api.v1.views.enterprise_customer_branding.EnterpriseCustomerBrandingViewSet attribute), 65	PendingEnterpriseCustomerAdminUserAdmin.Meta (class in enterprise.admin), 53 PendingEnterpriseCustomerAdminUserInline (class in enterprise.admin), 53
partial_update()	(enter-prise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet method), 62	PendingEnterpriseCustomerAdminUserSet (class in enter-prise.models), 239
partial_update()	(enter-prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet method), 67	PendingEnterpriseCustomerUser.DoesNotExist, 239 PendingEnterpriseCustomerUser.MultipleObjectsReturned, 239
partial_update()	(enter-prise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet method), 68	PendingEnterpriseCustomerRequestingCourseUser (class in enterprise.models.EnterpriseCustomer attribute), 158
partial_update()	(enter-prise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingViewSet method), 69	PendingEnterpriseCustomerUserAdmin (class in enter-prise.admin), 53
pending_enterprise_customer_user	(enter-prise.models.EnterpriseGroupMembership attribute), 190	PendingEnterpriseCustomerUserAdmin.Meta (class in enterprise.admin), 54 PendingEnterpriseCustomerUserEnterpriseAdminViewSet (class in enter-prise.api.v1.views.pending_enterprise_customer_user), 78
pending_enterprise_customer_user	(enter-prise.models.HistoricalEnterpriseGroupMembership attribute), 219	PendingEnterpriseCustomerUserSerializer (class in enterprise.api.v1.serializers), 92 PendingEnterpriseCustomerUserSerializer.Meta (class in enterprise.api.v1.serializers), 92
pending_enterprise_customer_user_id	(enter-prise.models.EnterpriseGroupMembership attribute), 190	PendingEnterpriseCustomerUserViewSet
pending_enterprise_customer_user_id	(enter-prise.models.HistoricalEnterpriseGroupMembership attribute), 219	
pending_enterprise_customer_user_model()	(in module enterprise.tasks), 246	
PENDING_LEARNERS	(enter-	(class in enter-

`prise.api.v1.views.pending_enterprise_customer_user`), `attribute`), 77
 78 `permission_classes` (enter-
`percentage_format()` (in module `enterprise.api.utils`), `prise.api.v1.views.pending_enterprise_customer_user.PendingEn`
 98 `attribute`), 78
`permission_classes` (enter- `permission_classes` (enter-
`prise.api.v1.views.analytics_summary.AnalyticsSummaryView`), `prise.api.v1.views.pending_enterprise_customer_user.PendingEn`
`attribute`), 55 `attribute`), 79
`permission_classes` (enter- `permission_classes` (enter-
`prise.api.v1.views.base_views.EnterpriseModelViewSet` `prise.api.v1.views.plotly_auth.PlotlyAuthView`
`attribute`), 55 `attribute`), 79
`permission_classes` (enter- `pgp_encryption_key` (enter-
`prise.api.v1.views.base_views.EnterpriseReadWriteModelViewSet` `prise.models.EnterpriseCustomerReportingConfiguration`
`attribute`), 56 `attribute`), 172
`permission_classes` (enter- `plaintext_template` (enter-
`prise.api.v1.views.base_views.EnterpriseViewSet` `prise.models.EnrollmentNotificationEmailTemplate`
`attribute`), 56 `attribute`), 141
`permission_classes` (enter- `plaintext_template` (enter-
`prise.api.v1.views.base_views.EnterpriseWriteOnlyModelViewSet` `prise.models.HistoricalEnrollmentNotificationEmailTemplate`
`attribute`), 56 `attribute`), 193
`permission_classes` (enter- `PlotlyAuthView` (class in enter-
`prise.api.v1.views.coupon_codes.CouponCodesView` `prise.api.v1.views.plotly_auth`), 79
`attribute`), 57 `post()` (enterprise.admin.views.EnterpriseCustomerManageLearnerDataS
`permission_classes` (enter- `method`), 37
`prise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryViewSet` `prise.views.EnterpriseCustomerManageLearnersView`
`attribute`), 58 `method`), 40
`permission_classes` (enter- `post()` (enterprise.admin.views.EnterpriseCustomerSetupAuthOrgIDView
`prise.api.v1.views.enterprise_customer_api_credentials.APICredentialsRegenerateViewSet`
`attribute`), 63 `post()` (enterprise.admin.views.EnterpriseCustomerTransmitCoursesView
`permission_classes` (enter- `method`), 41
`prise.api.v1.views.enterprise_customer_api_credentials.APICredentialsViewSet` `prise.api.v1.views.analytics_summary.AnalyticsSummaryView`
`attribute`), 64 `posts()` (enterprise.admin.views.
`permission_classes` (enter- `post()` (enterprise.api.v1.views.coupon_codes.CouponCodesView
`prise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationViewSet`
`attribute`), 65 `method`), 57
`permission_classes` (enter- `post()` (enterprise.api.v1.views.notifications.NotificationReadView
`prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet` `prise.views.CourseEnrollmentView`
`attribute`), 67 `method`), 267
`permission_classes` (enter- `post()` (enterprise.views.GrantDataSharingPermissions
`prise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet` `prise.views.EnterpriseCustomerInviteKeyView`
`attribute`), 68 `method`), 269
`permission_classes` (enter- `post()` (enterprise.views.ProgramEnrollmentView
`prise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingViewSet` `prise.views.CourseEnrollmentView`
`attribute`), 69 `method`), 271
`permission_classes` (enter- `prev_record` (enterprise.models.HistoricalEnrollmentNotificationEmailTe
`prise.api.v1.views.enterprise_customer_reporting.EnterpriseReportingTypesViewSet` `prise.models.HistoricalEnterpriseCourseEnrollment`
`attribute`), 69 `property`), 193
`permission_classes` (enter- `prev_record` (enterprise.models.HistoricalEnterpriseCourseEntitlement
`prise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationViewSet` `prise.models.HistoricalEnterpriseCustomer`
`attribute`), 70 `property`), 187
`permission_classes` (enter- `prev_record` (enterprise.models.HistoricalEnterpriseCustomerCatalog
`prise.api.v1.views.enterprise_group.EnterpriseGroupViewSet` `property`), 203
`attribute`), 73 `property`), 205
`permission_classes` (enter- `prev_record` (enterprise.models.HistoricalEnterpriseCustomerInviteKey
`prise.api.v1.views.notifications.NotificationReadView` `property`), 208

`prev_record(enterprise.models.HistoricalEnterpriseCustomerProgramExistsOn` (enter-
`property), 211` `prise.api_client.discovery.CourseCatalogApiServiceClient`
`prev_record(enterprise.models.HistoricalEnterpriseCustomerUser method), 102`
`property), 214` `PROGRAM_TYPES_ENDPOINT` (enter-
`prev_record(enterprise.models.HistoricalEnterpriseGroup` `prise.api_client.discovery.CourseCatalogApiClient`
`property), 216` `attribute), 100`
`prev_record(enterprise.models.HistoricalEnterpriseGroupProgramDetailSerializer (class in enter-`
`property), 219` `prise.api.v1.serializers), 92`
`prev_record(enterprise.models.HistoricalLearnerCreditProgramEnrollmentView (class in enterprise.views),`
`property), 222` `270`
`prev_record(enterprise.models.HistoricalLicensedEnterpriseProgramsEndpoint` (enter-
`property), 224` `prise.api_client.discovery.CourseCatalogApiClient`
`prev_record(enterprise.models.HistoricalPendingEnrollment` `attribute), 100`
`property), 226` `prompt(enterprise.models.ChatGPTResponse attribute),`
`prev_record(enterprise.models.HistoricalPendingEnterpriseCustomerAdminUser`
`property), 229` `prompt_hash(enterprise.models.ChatGPTResponse at-`
`prev_record(enterprise.models.HistoricalPendingEnterpriseCustomerUser), 140`
`property), 231` `prompt_type(enterprise.models.ChatGPTResponse at-`
`prev_record(enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment`
`property), 233` `PROMPT_TYPES(enterprise.models.ChatGPTResponse`
`preview() (enterprise.admin.EnrollmentNotificationEmailTemplateAdmin attribute), 139`
`method), 43` `provider_id(enterprise.models.EnterpriseCustomerIdentityProvider`
`preview_as_course() (enter-` `attribute), 167`
`prise.admin.EnrollmentNotificationEmailTemplateAdmin` `provider_name` (enter-
`method), 43` `prise.models.EnterpriseCustomerIdentityProvider`
`preview_as_program() (enter-` `property), 167`
`prise.admin.EnrollmentNotificationEmailTemplateAdmin` `publish_audit_enrollment_urls` (enter-
`method), 43` `prise.models.EnterpriseCustomerCatalog`
`preview_catalog_url() (enter-` `attribute), 165`
`prise.admin.EnterpriseCustomerCatalogAdmin` `publish_audit_enrollment_urls` (enter-
`method), 48` `prise.models.HistoricalEnterpriseCustomerCatalog`
`PREVIEW_EMAIL_TEMPLATE (enter-` `attribute), 205`
`prise.admin.utils.UrlNames attribute), 35`
`preview_mode(enterprise.views.GrantDataSharingPermissions`
`attribute), 269`
`PREVIEW_QUERY_RESULT (enter-`
`prise.admin.utils.UrlNames attribute), 35` `queryset(enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCa`
`PRIMARY (enterprise.constants.DefaultColors attribute),` `attribute), 58`
`129` `queryset(enterprise.api.v1.views.enterprise_course_enrollment.Enterpris`
`primary_color (enter-` `attribute), 59`
`prise.models.EnterpriseCustomerBrandingConfiguration` `queryset(enterprise.api.v1.views.enterprise_customer.EnterpriseCustome`
`attribute), 162` `attribute), 62`
`process() (enterprise.logging.EnterpriseRequestIdLoggerAdapter` `queryset(enterprise.api.v1.views.enterprise_customer_branding_configur`
`method), 132` `attribute), 65`
`process_request() (enter-` `queryset(enterprise.api.v1.views.enterprise_customer_catalog.Enterprise`
`prise.middleware.EnterpriseLanguagePreferenceMiddleware` `attribute), 66`
`method), 134` `attribute), 67`
`PROFESSIONAL (enterprise.constants.CourseModes at-` `queryset(enterprise.api.v1.views.enterprise_customer_invite_key.Enterpr`
`tribute), 129` `attribute), 68`
`program_detail() (enter-` `queryset(enterprise.api.v1.views.enterprise_customer_reporting.Enterpri`
`prise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet` `attribute), 69`
`method), 66` `queryset(enterprise.api.v1.views.enterprise_customer_sso_configuration.`
`PROGRAM_ENROLLMENT_VIEW_URL (enter-` `attribute), 70`
`prise.views.RouterView attribute), 271` `queryset(enterprise.api.v1.views.enterprise_customer_user.EnterpriseCu`
`attribute), 72`

`queryset` (`enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet` attribute), 73
`queryset` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseCourseEnrollmentViewSet` attribute), 76
`queryset` (`enterprise.api.v1.views.pending_enterprise_customer_user.EnterpriseCustomerUserViewSet` attribute), 79
`queryset_with_removed` (`enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet` attribute), 73
R
`rate` (`enterprise.heartbeat.throttles.HeartbeatRateThrottle` attribute), 116
`reactivate()` (`enterprise.models.EnterpriseFulfillmentSource` method), 187
`reactivate()` (`enterprise.models.LearnerCreditEnterpriseCourseEnrollment` method), 235
`read_replica_or_default()` (in module `enterprise.api.v1.views.enterprise_course_enrollment`), 59
`readonly_fields` (`enterprise.admin.ChatGPTResponseAdmin` attribute), 43
`readonly_fields` (`enterprise.admin.EnterpriseCatalogQueryAdmin` attribute), 44
`readonly_fields` (`enterprise.admin.EnterpriseCourseEnrollmentAdmin` attribute), 45
`readonly_fields` (`enterprise.admin.EnterpriseCustomerCatalogAdmin` attribute), 48
`readonly_fields` (`enterprise.admin.EnterpriseCustomerInviteKeyAdmin` attribute), 49
`readonly_fields` (`enterprise.admin.EnterpriseCustomerUserAdmin` attribute), 51
`readonly_fields` (`enterprise.admin.EnterpriseGroupAdmin` attribute), 52
`readonly_fields` (`enterprise.admin.PendingEnrollmentAdmin` attribute), 53
`readonly_fields` (`enterprise.admin.PendingEnterpriseCustomerAdminUserAdmin` attribute), 53
`readonly_fields` (`enterprise.admin.PendingEnterpriseCustomerAdminUserInline` attribute), 53
`readonly_fields` (`enterprise.admin.PendingEnterpriseCustomerUserAdmin` attribute), 54
`ready()` (`enterprise.apps.EnterpriseConfig` method), 128
`REASON` (`enterprise.admin.forms.ManageLearnersForm` attribute), 33
`refresh_catalog()` (in module `enterprise.admin.actions`), 25
`REFRESH_CATALOG_ENDPOINT` (`enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient` attribute), 104
`refresh_catalogs()` (`enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient` method), 106
`refresh_token()` (`enterprise.api_client.client.UserAPIClient` static method), 99
`remove_learners()` (`enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet` method), 73
`removed_at` (`enterprise.models.EnterpriseGroupMembership` attribute), 190
`removed_at` (`enterprise.models.HistoricalEnterpriseGroupMembership` attribute), 219
`render_all_templates()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 142
`render_html_template()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 142
`render_page_with_error_code_message()` (in module `enterprise.views`), 272
`render_plaintext_template()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 142
`render_template()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 142
`renderer_classes` (`enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet` attribute), 66
`replace_sensitive_sso_username` (`enterprise.models.EnterpriseCustomer` attribute), 159
`replace_sensitive_sso_username` (`enterprise.models.HistoricalEnterpriseCustomer` attribute), 203
`reply_to` (`enterprise.models.EnterpriseCustomer` attribute), 159
`reply_to` (`enterprise.models.HistoricalEnterpriseCustomer` attribute), 203

`report_type` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 172
`REPORT_TYPE_CHOICES` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 170
`REPORT_TYPE_CSV` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 170
`REPORT_TYPE_JSON` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 170
`reporting_configurations` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 159
`REQ_EXP_LICENSE_UUIDS_PARAM` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment` attribute), 76
`requests_data_sharing_consent` (`enterprise.models.EnterpriseCustomerReportingConfiguration` property), 159
`require_at_least_one_query_parameter()` (in module `enterprise.api.v1.decorators`), 79
`REQUIRED_PARAM_EMAIL` (`enterprise.api.v1.views.coupon_codes.CouponCodesView` attribute), 56
`REQUIRED_PARAM_ENTERPRISE_NAME` (`enterprise.api.v1.views.coupon_codes.CouponCodesView` attribute), 56
`REQUIRED_PARAM_ENTERPRISE_SLUG` (`enterprise.api.v1.views.notifications.NotificationReadView` attribute), 77
`REQUIRED_PARAM_NOTIFICATION_ID` (`enterprise.api.v1.views.notifications.NotificationReadView` attribute), 77
`response` (`enterprise.models.ChatGPTResponse` attribute), 140
`ResponsePaginationSerializer` (class in `enterprise.api.v1.serializers`), 92
`retrieve()` (`enterprise.api.v1.views.enterprise_customer_api_credentials` method), 64
`retrieve()` (`enterprise.api.v1.views.enterprise_customer_catalog` method), 66
`retrieve()` (`enterprise.api.v1.views.enterprise_customer_invite_key` method), 68
`retrieve()` (`enterprise.api.v1.views.enterprise_customer_reporting` method), 69
`retrieve()` (`enterprise.api.v1.views.enterprise_customer_sso_configuration` method), 70
`retrieve()` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment` method), 75
`revert_enrollment_objects()` (`enterprise.management.commands.revert_enrollment_objects` method), 124
`revert_url()` (`enterprise.models.HistoricalEnrollmentNotification` attribute), 193
`revert_url()` (`enterprise.models.HistoricalEnterpriseCourseEnrollment` attribute), 195
`revert_url()` (`enterprise.models.HistoricalEnterpriseCourseEntitlement` attribute), 197
`revert_url()` (`enterprise.models.HistoricalEnterpriseCustomer` attribute), 203
`revert_url()` (`enterprise.models.HistoricalEnterpriseCustomerCatalog` attribute), 205
`revert_url()` (`enterprise.models.HistoricalEnterpriseCustomerInviteKey` attribute), 208
`revert_url()` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212
`revert_url()` (`enterprise.models.HistoricalEnterpriseCustomerUser` attribute), 214
`revert_url()` (`enterprise.models.HistoricalEnterpriseEnrollmentNotification` attribute), 217
`revert_url()` (`enterprise.models.HistoricalEnterpriseGroupMembership` attribute), 219
`revert_url()` (`enterprise.models.HistoricalLearnerCreditEnterpriseCourse` attribute), 222
`revert_url()` (`enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment` attribute), 224
`revert_url()` (`enterprise.models.HistoricalPendingEnrollment` attribute), 226
`revert_url()` (`enterprise.models.HistoricalPendingEnterpriseCustomerAPICredentials` attribute), 229
`revert_url()` (`enterprise.models.HistoricalPendingEnterpriseCustomerUser` attribute), 231
`revert_url()` (`enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment` attribute), 233
`revoke()` (`enterprise.models.EnterpriseFulfillmentSource` attribute), 187
`role` (`enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfiguration` attribute), 114
`role` (`enterprise.models.EnterpriseFeatureUserRoleAssignment` attribute), 185
`role` (`enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment` attribute), 233
`role` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` attribute), 233
`ROLE_CHOICES` (`enterprise.config.models.UpdateRoleAssignmentsWithCustomersConfiguration` attribute), 114
`role_class` (`enterprise.models.EnterpriseFeatureUserRoleAssignment` attribute), 185
`role_id` (`enterprise.models.EnterpriseFeatureUserRoleAssignment` attribute), 185
`role_id` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212
`role_id` (`enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment` attribute), 233
`role_id` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` attribute), 233
`roles_by_name()` (in module `enterprise.roles_api`), 245
`root()` (in module `enterprise.settings.test`), 127
`RouteView` (class in `enterprise.views`), 271

`run_checks()` (in module `enterprise.heartbeat.utils`), 116
`run_from_argv()` (`enterprise.management.commands.manufacture_data.Command` method), 121
S
`safe_branding_configuration` (`enterprise.models.EnterpriseCustomer` property), 159
`safe_logo_url` (`enterprise.models.EnterpriseCustomerBrandingConfiguration` property), 162
`SALES_FORCE_ID` (`enterprise.admin.forms.ManageLearnersForm.Fields` attribute), 33
`sales_force_id` (`enterprise.models.HistoricalPendingEnrollment` attribute), 227
`sales_force_id` (`enterprise.models.PendingEnrollment` attribute), 237
`sap_config_fields` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176
`SAP_SUCCESS_FACTORS` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 173
`sapsf_oauth_root_url` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176
`sapsf_oauth_root_url` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212
`sapsf_private_key` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176
`sapsf_private_key` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212
`sapsuccessfactorsenterpriseconfigurationset` (`enterprise.models.EnterpriseCustomer` attribute), 159
`save()` (`enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWriteSerializer` method), 83
`save()` (`enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyWriteSerializer` method), 87
`save()` (`enterprise.api.v1.serializers.EnterpriseCustomerUserWriteSerializer` method), 90
`save()` (`enterprise.models.ChatGPTResponse` method), 140
`save()` (`enterprise.models.EnterpriseCustomerCatalog` method), 165
`save()` (`enterprise.models.EnterpriseCustomerInviteKey` method), 169
`save()` (`enterprise.models.EnterpriseCustomerSsoConfiguration` method), 176
`save()` (`enterprise.models.EnterpriseCustomerUser` method), 180
`save()` (`enterprise.models.EnterpriseFulfillmentSource` method), 187
`save()` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` method), 243
`save_logo_file()` (in module `enterprise.signals`), 246
`save_without_historical_record()` (`enterprise.models.EnrollmentNotificationEmailTemplate` method), 142
`save_without_historical_record()` (`enterprise.models.EnterpriseCourseEnrollment` method), 145
`save_without_historical_record()` (`enterprise.models.EnterpriseCourseEntitlement` method), 148
`save_without_historical_record()` (`enterprise.models.EnterpriseCustomer` method), 159
`save_without_historical_record()` (`enterprise.models.EnterpriseCustomerCatalog` method), 166
`save_without_historical_record()` (`enterprise.models.EnterpriseCustomerInviteKey` method), 169
`save_without_historical_record()` (`enterprise.models.EnterpriseCustomerSsoConfiguration` method), 176
`save_without_historical_record()` (`enterprise.models.EnterpriseCustomerUser` method), 181
`save_without_historical_record()` (`enterprise.models.EnterpriseFulfillmentSource` method), 187
`save_without_historical_record()` (`enterprise.models.EnterpriseGroup` method), 188
`save_without_historical_record()` (`enterprise.models.EnterpriseGroupMembership` method), 190
`save_without_historical_record()` (`enterprise.models.PendingEnrollment` method), 227
`save_without_historical_record()` (`enterprise.models.PendingEnterpriseCustomerAdminUser` method), 239
`save_without_historical_record()` (`enterprise.models.PendingEnterpriseCustomerUser` method), 241
`save_without_historical_record()` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` method), 243

saved_for_later	(enterprise.models.EnterpriseCourseEnrollment attribute), 145	tribute), 129	secondary_color	(enterprise.models.EnterpriseCustomerBrandingConfiguration attribute), 162
saved_for_later	(enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 195	select_enterprise_page_as_redirect_url()	(in module enterprise.tpa_pipeline), 248	
scope	(enterprise.heartbeat.throttles.HeartbeatRateThrottles attribute), 116	send_email_notification_message()	(in module enterprise.utils), 263	
SEARCH_ALL_ENDPOINT	(enterprise.api_client.discovery.CourseCatalogApiClient attribute), 100	send_messages()	(enterprise.admin.views.EnterpriseCustomerManageLearnersView class method), 40	
search_fields	(enterprise.admin.EnterpriseCourseEnrollmentAdmin attribute), 45	sender_alias	(enterprise.models.EnterpriseCustomer attribute), 159	
search_fields	(enterprise.admin.EnterpriseCustomerAdmin attribute), 47	sender_alias	(enterprise.models.HistoricalEnterpriseCustomer attribute), 203	
search_fields	(enterprise.admin.EnterpriseCustomerCatalogAdmin attribute), 48	serialize_notification_content()	(in module enterprise.utils), 263	
search_fields	(enterprise.admin.EnterpriseCustomerInviteKeyAdmin attribute), 49	serialized	(enterprise.models.EnterpriseCustomer property), 160	
search_fields	(enterprise.admin.EnterpriseCustomerReportingConfigurationAdmin attribute), 50	serializer_class	(enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQuery attribute), 58	
search_fields	(enterprise.admin.EnterpriseCustomerTypeAdmin attribute), 50	serializer_class	(enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView attribute), 62	
search_fields	(enterprise.admin.EnterpriseCustomerUserAdmin attribute), 51	serializer_class	(enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfiguration attribute), 65	
search_fields	(enterprise.admin.EnterpriseGroupAdmin attribute), 52	serializer_class	(enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalog attribute), 67	
search_fields	(enterprise.admin.EnterpriseGroupMembershipAdmin attribute), 52	serializer_class	(enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReporting attribute), 69	
search_fields	(enterprise.admin.PendingEnrollmentAdmin attribute), 53	serializer_class	(enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfiguration attribute), 70	
search_fields	(enterprise.admin.PendingEnterpriseCustomerAdminUserAdmin attribute), 53	serializer_class	(enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet attribute), 73	
search_fields	(enterprise.admin.PendingEnterpriseCustomerUserAdmin attribute), 54	serializer_class	(enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseSubsidyFulfillment attribute), 76	
search_fields	(enterprise.admin.SystemWideEnterpriseUserRoleAssignmentAdmin attribute), 54	serializer_class	(enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUser attribute), 78	
SEARCH_KEYWORD	(enterprise.admin.views.EnterpriseCustomerManageLearnersView attribute), 38	service_name	(enterprise.heartbeat.exceptions.DiscoveryNotAvailable attribute), 115	
SECONDARY	(enterprise.constants.DefaultColors attribute), 129	service_name	(enterprise.heartbeat.exceptions.EcommerceNotAvailable attribute), 115	
		service_name	(enterprise.heartbeat.exceptions.EnterpriseCatalogNotAvailable attribute), 115	

attribute), 115
 service_name (enterprise.heartbeat.exceptions.LMSNotAvailable (class in enterprise.api.v1.serializers), 93
 attribute), 115
 service_name (enterprise.heartbeat.exceptions.ServiceNotAvailable (class in enterprise.signals), 246
 attribute), 116
 ServiceNotAvailable, 115
 ServiceUserThrottle (class in enterprise.api.throttles), 96
 set_application_name_from_user_id() (in module enterprise.api.utils), 98
 set_final_prices() (enterprise.views.CourseEnrollmentView method), 267
 set_single_customization() (enterprise.management.commands.manufacture_data.Nuke method), 121
 SETUP_AUTH_ORG_ID (enterprise.admin.utils.UrlNames attribute), 35
 setup_auth_org_id() (enterprise.admin.EnterpriseCustomerAdmin method), 47
 sftp_file_path (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 172
 sftp_hostname (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 172
 sftp_port (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 172
 sftp_username (enterprise.models.EnterpriseCustomerReportingConfiguration attribute), 173
 should_inactivate_other_customers (enterprise.models.EnterpriseCustomerUser attribute), 181
 should_inactivate_other_customers (enterprise.models.HistoricalEnterpriseCustomerUser attribute), 214
 should_upgrade_to_licensed_enrollment() (in module enterprise.views), 272
 show_full_result_count (enterprise.admin.SystemWideEnterpriseUserRoleAssignment attribute), 54
 site (enterprise.models.EnterpriseCustomer attribute), 160
 site (enterprise.models.HistoricalEnterpriseCustomer attribute), 203
 site_id (enterprise.models.EnterpriseCustomer attribute), 160
 site_id (enterprise.models.HistoricalEnterpriseCustomer attribute), 203
 SiteField (class in enterprise.api.v1.serializers), 93
 SiteSerializer (class in enterprise.api.v1.serializers), 93
 SiteSerializer.Meta (class in enterprise.api.v1.serializers), 93
 skip_saving_logo_file() (in module enterprise.signals), 246
 slug (enterprise.models.EnterpriseCustomer attribute), 160
 slug (enterprise.models.EnterpriseEnrollmentSource attribute), 183
 slug (enterprise.models.HistoricalEnterpriseCustomer attribute), 203
 source (enterprise.models.EnterpriseCourseEnrollment attribute), 146
 source (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 195
 source (enterprise.models.HistoricalPendingEnrollment attribute), 227
 source (enterprise.models.PendingEnrollment attribute), 237
 source_id (enterprise.models.EnterpriseCourseEnrollment attribute), 146
 source_id (enterprise.models.HistoricalEnterpriseCourseEnrollment attribute), 195
 source_id (enterprise.models.HistoricalPendingEnrollment attribute), 227
 source_id (enterprise.models.PendingEnrollment attribute), 238
 split_usernames_and_emails() (in module enterprise.admin.utils), 36
 sso_orchestration_records (enterprise.models.EnterpriseCustomer attribute), 160
 SsoConfigurationApiError, 70
 SsoOrchestratorClientError, 112
 start_date (enterprise.models.AdminNotification attribute), 135
 Status (class in enterprise.heartbeat.utils), 116
 status (enterprise.models.EnterpriseGroupMembership attribute), 190
 status (enterprise.models.HistoricalEnterpriseGroupMembership attribute), 219
 strip_html_tags() (in module enterprise.utils), 264
 SUBJECT_HELP_TEXT (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 141
 subject_line (enterprise.models.EnrollmentNotificationEmailTemplate attribute), 142
 subject_line (enterprise.models.HistoricalEnrollmentNotificationEmailTemplate attribute), 193
 submit_for_configuration() (enterprise.models.EnterpriseCustomerSsoConfiguration method), 176
 SubmitInput (class in enterprise.admin.widgets), 42
 submitted_at (enterprise.models.EnterpriseCustomerSsoConfiguration attribute), 176

`submitted_at` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212
`SystemWideEnterpriseUserRoleAssignment` (class in `enterprise.models`), 242
`subsidized_enterprise_course_enrollment_model()` (in module `enterprise.utils`), 264
`SystemWideEnterpriseUserRoleAssignment.DoesNotExist`, 242
`suffix` (`enterprise.api.v1.views.enterprise_catalog_query.EnterpriseCatalogQueryViewSet` attribute), 58
`SystemWideEnterpriseUserRoleAssignment.MultipleObjectsReturned`, 242
`suffix` (`enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet` attribute), 59
`SystemWideEnterpriseUserRoleAssignmentAdmin`, 242
`suffix` (`enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet` attribute), 62
`SystemWideEnterpriseUserRoleAssignmentAdmin.Meta`, 242
`suffix` (`enterprise.api.v1.views.enterprise_customer_api_credentials.AllPKCredentialsRegenerationViewSet` attribute), 63
`SystemWideEnterpriseUserRoleAssignmentForm`, 34
`suffix` (`enterprise.api.v1.views.enterprise_customer_api_credentials.AllPKCredentialsViewSet` attribute), 64
`SystemWideEnterpriseUserRoleAssignmentForm.Meta`, 34
`suffix` (`enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationViewSet` attribute), 65
`template_name` (`enterprise.admin.views.BaseEnterpriseCustomerView` attribute), 41
`suffix` (`enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogWriteViewSet` attribute), 67
`template_name` (`enterprise.admin.views.EnterpriseCustomerManageLearnerDataView` attribute), 41
`suffix` (`enterprise.api.v1.views.enterprise_customer_invite_key.EnterpriseCustomerInviteKeyViewSet` attribute), 68
`template_name` (`enterprise.admin.views.EnterpriseCustomerManageLearnersView` attribute), 41
`suffix` (`enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingConfigurationViewSet` attribute), 69
`template_name` (`enterprise.admin.views.EnterpriseCustomerSetupAuthOrgIDView` attribute), 41
`suffix` (`enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationViewSet` attribute), 70
`template_name` (`enterprise.admin.views.EnterpriseCustomerTransmitCoursesView` attribute), 41
`suffix` (`enterprise.api.v1.views.enterprise_customer_user.EnterpriseCustomerUserViewSet` attribute), 72
`template_name` (`enterprise.admin.widgets.SubmitInput` attribute), 42
`suffix` (`enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet` attribute), 73
`template_name` (`enterprise.views.EnterpriseLoginView` attribute), 268
`suffix` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet` attribute), 75
`template_name` (`enterprise.views.EnterpriseSelectionView` attribute), 268
`suffix` (`enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseCourseEnrollmentViewSet` attribute), 76
`template_type` (`enterprise.models.PendingEnterpriseCustomerUserEnterpriseAdminViewSet` attribute), 78
`suffix` (`enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserViewSet` attribute), 79
`template_type` (`enterprise.models.EnrollmentNotificationEmailTemplate` attribute), 142
`sync_learner_profile_data` (`enterprise.models.EnterpriseCustomer` property), 160
`template_type_choices` (`enterprise.models.HistoricalEnrollmentNotificationEmailTemplate` attribute), 193
`sync_learner_profile_data` (`enterprise.models.EnterpriseCustomerIdentityProvider` property), 167
`TemplatePreviewView` (class in `enterprise.admin.views`), 41
`system_wide_role_assignments` (`enterprise.models.EnterpriseCustomer` attribute), 160
`TERTIARY` (`enterprise.constants.DefaultColors` attribute), 129
`system_wide_role_assignments` (`enterprise.models.SystemWideEnterpriseRole` attribute), 242
`tertiary_color` (`enterprise.models.EnterpriseCustomerBrandingConfiguration` attribute), 162
`SystemWideEnterpriseRole` (class in `enterprise.models`), 241
`text` (`enterprise.models.AdminNotification` attribute), 135
`SystemWideEnterpriseRole.DoesNotExist`, 241
`ThirdPartyAuthApiClient` (class in `enterprise.api.client.lms`), 111
`SystemWideEnterpriseRole.MultipleObjectsReturned`, 241

[throttle_classes](#) (enter-prise.api.v1.views.base_views.EnterpriseViewSet attribute), 56
[throttle_classes](#) (enter-prise.api.v1.views.coupon_codes.CouponCodesView attribute), 57
[throttle_classes](#) (enter-prise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet attribute), 62
[throttle_classes](#) (enter-prise.api.v1.views.notifications.NotificationReadView attribute), 77
[title](#) (enterprise.models.AdminNotification attribute), 136
[title](#) (enterprise.models.EnterpriseCatalogQuery attribute), 143
[title](#) (enterprise.models.EnterpriseCustomerCatalog attribute), 166
[title](#) (enterprise.models.HistoricalEnterpriseCustomerCatalog attribute), 205
[to_internal_value\(\)](#) (enter-prise.api.v1.fields.Base64EmailCSVField method), 80
[to_internal_value\(\)](#) (enter-prise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentListSerializer method), 85
[to_internal_value\(\)](#) (enter-prise.api.v1.serializers.SiteField method), 93
[to_representation\(\)](#) (enter-prise.api.v1.fields.Base64EmailCSVField method), 80
[to_representation\(\)](#) (enter-prise.api.v1.serializers.CourseDetailSerializer method), 81
[to_representation\(\)](#) (enter-prise.api.v1.serializers.CourseRunDetailSerializer method), 81
[to_representation\(\)](#) (enter-prise.api.v1.serializers.EnterpriseCustomerCatalogDetailSerializer method), 84
[to_representation\(\)](#) (enter-prise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsListSerializer method), 85
[to_representation\(\)](#) (enter-prise.api.v1.serializers.PendingEnterpriseCustomerCourseEnrollmentSerializer method), 92
[to_representation\(\)](#) (enter-prise.api.v1.serializers.ProgramDetailSerializer method), 92
[to_representation\(\)](#) (enter-prise.api.v1.serializers.SiteField method), 93
[toggle_universal_link\(\)](#) (enter-prise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet method), 62
[toggle_universal_link\(\)](#) (enter-prise.models.EnterpriseCustomer method), 160
[token_is_expired\(\)](#) (enter-prise.api_client.client.UserAPIClient method),
[top_down_assignment_real_time_lcm\(\)](#) (in module enterprise.toggles), 247
[track_enrollment\(\)](#) (in module enterprise.utils), 264
[track_enterprise_user_linked\(\)](#) (in module enterprise.utils), 264
[track_event\(\)](#) (in module enterprise.utils), 264
[transaction_id](#) (enter-prise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment attribute), 222
[transaction_id](#) (enter-prise.models.LearnerCreditEnterpriseCourseEnrollment attribute), 235
[TRANSMIT_COURSES_METADATA](#) (enter-prise.admin.utils.UrlNames attribute), 35
[transmit_courses_metadata\(\)](#) (enter-prise.admin.EnterpriseCustomerAdmin method), 41
[TRANSMIT_COURSES_METADATA_FORM](#) (enter-prise.admin.views.EnterpriseCustomerTransmitCoursesView.ConcreteForm attribute), 41
[TransmitEnterpriseCoursesForm](#) (class in enterprise.admin.forms), 34
[traverse_get_content_metadata\(\)](#) (enter-prise.api_client.enterprise_catalog.EnterpriseCatalogApiClient method), 106
[traverse_pagination\(\)](#) (enter-prise.api_client.discovery.CourseCatalogApiClient method), 102
[traverse_pagination\(\)](#) (in module enterprise.utils), 264
[truncate_string\(\)](#) (in module enterprise.utils), 264
U
[UNAVAILABLE](#) (enterprise.heartbeat.utils.Status attribute), 71
[unenroll\(\)](#) (enterprise.models.EnterpriseCustomerUser method), 181
[UNENROLLED](#) (enter-prise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseSubsidy attribute), 76
[unenroll_user_from_course\(\)](#) (enter-prise.api_client.lms.EnrollmentApiClient method), 109
[UNENROLLED](#) (enterprise.api.v1.views.enterprise_subsidy_fulfillment.LicensedEnterpriseSubsidy attribute), 75

unenrolled (*enterprise.models.EnterpriseCourseEnrollment* attribute), 146

unenrolled (*enterprise.models.HistoricalEnterpriseCourseEnrollment* attribute), 195

unenrolled() (*enterprise.api.v1.views.enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet* method), 75

unenrolled_at (*enterprise.models.EnterpriseCourseEnrollment* attribute), 146

unenrolled_at (*enterprise.models.HistoricalEnterpriseCourseEnrollment* attribute), 195

ungettext_min_max() (in module *enterprise.utils*), 265

UNIQUE (*enterprise.api.v1.views.pending_enterprise_customer_pending_enterprise_subsidy_fulfillment.EnterpriseSubsidyFulfillmentViewSet* attribute), 78

unlink_user() (*enterprise.models.EnterpriseCustomerUserManager* method), 182

unlink_users() (*enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet* method), 62

UnlinkUserFromEnterpriseError, 131

UNPAID_EXECUTIVE_EDUCATION (*enterprise.constants.CourseModes* attribute), 129

unset_enterprise_learner_language() (in module *enterprise.utils*), 265

unset_language_of_all_enterprise_learners() (in module *enterprise.utils*), 265

update() (*enterprise.api.v1.serializers.EnterpriseCustomerUpdateSerializer* method), 83

update() (*enterprise.api.v1.serializers.EnterpriseCustomerUpdateSerializer* method), 88

update() (*enterprise.api.v1.serializers.ImmutableStateSerializer* method), 91

update() (*enterprise.api.v1.views.enterprise_customer_api_credentials.ApiCredentialsRegenerateViewSet* method), 63

update() (*enterprise.api.v1.views.enterprise_customer_api_credentials.ApiCredentialsViewSet* method), 64

update() (*enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingConfigurationViewSet* method), 70

update() (*enterprise.api.v1.views.enterprise_customer_sso_configuration.EnterpriseCustomerSsoConfigurationViewSet* method), 70

update() (*enterprise.api.v1.views.enterprise_group.EnterpriseGroupViewSet* method), 73

update_branding() (*enterprise.api.v1.views.enterprise_customer_branding_configuration.EnterpriseCustomerBrandingConfigurationViewSet* method), 65

update_config_last_errored_at() (*enterprise.management.commands.update_config_last_errored* method), 126

update_course_enrollment_mode_for_user() (*enterprise.api_client.lms.EnrollmentApiClient* method), 109

update_enterprise_catalog() (*enterprise.api_client.enterprise_catalog.EnterpriseCatalogApiClient* method), 106

update_enterprise_catalog_data() (in module *enterprise.signals*), 246

update_enterprise_catalog_query() (in module *enterprise.signals*), 246

update_from_metadata (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 176

update_from_metadata (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 212

update_from_metadata (*enterprise.models.PendingEnterpriseCustomerSsoConfiguration* attribute), 246

update_learner_language_preference() (in module *enterprise.signals*), 246

update_query_parameters() (in module *enterprise.utils*), 265

update_session() (*enterprise.models.EnterpriseCustomerUser* method), 181

update_throttle_scope() (*enterprise.api.throttles.BaseThrottle* method), 96

UpdateRoleAssignmentsWithCustomersConfig (class in *enterprise.config.models*), 113

UpdateRoleAssignmentsWithCustomersConfig.DoesNotExist, 113

UpdateRoleAssignmentsWithCustomersConfig.MultipleObjectsReturned, 113

UpdateRoleAssignmentsWithCustomersConfig.UrlNames (class in *enterprise.admin.utils*), 35

usage_count (*enterprise.models.EnterpriseCustomerInviteKey* attribute), 35

usage_limit (*enterprise.models.EnterpriseCustomerInviteKey* attribute), 35

usage_limit (*enterprise.models.HistoricalEnterpriseCustomerInviteKey* attribute), 308

use_for_related_fields (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 169

use_group_virus_scan (*enterprise.api.v1.serializers.EnterpriseCustomerUserWriteSerializer* attribute), 90

user (*enterprise.models.EnterpriseCustomerUser* property), 185

user (*enterprise.models.EnterpriseCustomerBrandingConfigurationViewSet* attribute), 185

user (*enterprise.models.EnterpriseFeatureUserRoleAssignment* attribute), 227

user (*enterprise.models.HistoricalPendingEnrollment* attribute), 233

user (*enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment* attribute), 233

user (*enterprise.models.PendingEnrollment* attribute), 233

238

`user` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` attribute), 243

`USER_ALREADY_REGISTERED` (`enterprise.utils.ValidationMessages` attribute), 249

`USER_DOES_NOT_EXIST` (`enterprise.api.v1.serializers.EnterpriseCustomerUserWriteSerializer` attribute), 90

`user_email` (`enterprise.models.EnterpriseCustomerUser` property), 181

`user_email` (`enterprise.models.HistoricalPendingEnterpriseCustomerUser` attribute), 229

`user_email` (`enterprise.models.HistoricalPendingEnterpriseCustomerUser` attribute), 231

`user_email` (`enterprise.models.PendingEnterpriseCustomerUser` attribute), 239

`user_email` (`enterprise.models.PendingEnterpriseCustomerUser` attribute), 241

`USER_EXISTS_ERROR` (`enterprise.api.v1.views.pending_enterprise_customer_user.PendingEnterpriseCustomerUserViewSet` attribute), 78

`user_id` (`enterprise.models.EnterpriseCustomerUser` attribute), 181

`user_id` (`enterprise.models.EnterpriseFeatureUserRoleAssignment` attribute), 185

`user_id` (`enterprise.models.HistoricalEnterpriseCustomerUser` attribute), 215

`user_id` (`enterprise.models.HistoricalPendingEnrollment` attribute), 227

`user_id` (`enterprise.models.HistoricalSystemWideEnterpriseUserRoleAssignment` attribute), 233

`user_id` (`enterprise.models.PendingEnrollment` attribute), 238

`user_id` (`enterprise.models.SystemWideEnterpriseUserRoleAssignment` attribute), 244

`user_id_attribute` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176

`user_id_attribute` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212

`USER_ID_FILTER` (`enterprise.api.v1.views.base_views.EnterpriseModelViewSet` attribute), 55

`USER_ID_FILTER` (`enterprise.api.v1.views.enterprise_course_enrollment.EnterpriseCourseEnrollmentViewSet` attribute), 58

`USER_ID_FILTER` (`enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerViewSet` attribute), 60

`USER_ID_FILTER` (`enterprise.api.v1.views.enterprise_customer_branding.EnterpriseCustomerBrandingViewSet` attribute), 64

`USER_ID_FILTER` (`enterprise.api.v1.views.enterprise_customer_catalog.EnterpriseCustomerCatalogViewSet` attribute), 65

`USER_ID_FILTER` (`enterprise.api.v1.views.enterprise_customer_reporting.EnterpriseCustomerReportingViewSet` attribute), 69

`USER_NOT_EXIST` (`enterprise.utils.ValidationMessages` attribute), 249

`USER_NOT_LINKED` (`enterprise.utils.ValidationMessages` attribute), 249

`UserAPIClient` (class in `enterprise.api_client.client`), 94

`UserFilterBackend` (class in `enterprise.api.filters`), 95

`username` (`enterprise.models.EnterpriseCustomerUser` property), 181

`username_attribute` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176

`username_attribute` (`enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration` attribute), 212

`UserSerializer` (class in `enterprise.api.v1.serializers`), 93

`UserSerializer.Meta` (class in `enterprise.api.v1.serializers`), 93

`uuid` (`enterprise.models.ChatGPTResponse` attribute), 140

`uuid` (`enterprise.models.EnterpriseCatalogQuery` attribute), 143

`uuid` (`enterprise.models.EnterpriseCourseEntitlement` attribute), 148

`uuid` (`enterprise.models.EnterpriseCustomer` attribute), 161

`uuid` (`enterprise.models.EnterpriseCustomerCatalog` attribute), 166

`uuid` (`enterprise.models.EnterpriseCustomerInviteKey` attribute), 169

`uuid` (`enterprise.models.EnterpriseCustomerReportingConfiguration` attribute), 173

`uuid` (`enterprise.models.EnterpriseCustomerSsoConfiguration` attribute), 176

`uuid` (`enterprise.models.EnterpriseFulfillmentSource` attribute), 187

`uuid` (`enterprise.models.EnterpriseGroup` attribute), 189

`uuid` (`enterprise.models.EnterpriseGroupMembership` attribute), 190

`uuid` (`enterprise.models.EnterpriseGroupEnrollment` attribute), 197

`uuid` (`enterprise.models.HistoricalEnterpriseCustomer` attribute), 203

`uuid` (`enterprise.models.HistoricalEnterpriseCustomerCatalog` attribute), 206

`uuid` (`enterprise.models.HistoricalEnterpriseCustomerBranding` attribute), 208

uuid(*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 212
 uuid(*enterprise.models.HistoricalEnterpriseGroup* attribute), 217
 uuid(*enterprise.models.HistoricalEnterpriseGroupMembership* attribute), 219
 uuid(*enterprise.models.HistoricalLearnerCreditEnterpriseCourseEnrollment* attribute), 222
 uuid(*enterprise.models.HistoricalLicensedEnterpriseCourseEnrollment* attribute), 224
 uuid(*enterprise.models.LearnerCreditEnterpriseCourseEnrollment* attribute), 235
 uuid(*enterprise.models.LicensedEnterpriseCourseEnrollment* attribute), 236
 uuid_nowrap() (*enterprise.admin.EnterpriseCustomerCatalogAdmin* method), 48
 V
 valid_image_extensions (*enterprise.apps.EnterpriseConfig* attribute), 128
 valid_max_image_size (*enterprise.apps.EnterpriseConfig* attribute), 128
 validate() (*enterprise.api.v1.serializers.EnrollmentsInfoSerializer* method), 82
 validate() (*enterprise.api.v1.serializers.EnterpriseCustomerBulkEnrollmentsSerializer* method), 84
 validate() (*enterprise.api.v1.serializers.EnterpriseCustomerBulkSubscriptionEnrollmentsSerializer* method), 84
 validate() (*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* method), 86
 validate() (*enterprise.api.v1.serializers.EnterpriseCustomerReportingConfigurationSerializer* method), 88
 validate_compression() (*enterprise.models.EnterpriseCustomerReportingConfiguration* class method), 173
 validate_content_filter_fields() (in module *enterprise.validators*), 266
 validate_course_exists_for_enterprise() (in module *enterprise.utils*), 266
 validate_course_run_id() (*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* method), 86
 validate_csv() (in module *enterprise.admin.utils*), 36
 validate_delivery_method() (*enterprise.models.EnterpriseCustomerReportingConfiguration* method), 173
 validate_email_to_link() (in module *enterprise.utils*), 266
 validate_enterprise_customer() (*enterprise.api.v1.serializers.LinkLearnersSerializer* method), 92
 validate_enterprise_customer_uuid() (*enterprise.api.v1.serializers.EnterpriseCustomerInviteKeyWriteSerializer* method), 87
 validate_hex_color() (in module *enterprise.validators*), 266
 validate_image_extension() (in module *enterprise.validators*), 266
 validate_image_size() (in module *enterprise.validators*), 266
 validate_lms_user_id() (*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* method), 86
 validate_pgp_encryption_key() (*enterprise.api.v1.serializers.EnterpriseCustomerReportingConfigurationSerializer* method), 88
 validate_pgp_key() (in module *enterprise.validators*), 267
 validate_provider_config() (in module *enterprise.tpa_pipeline*), 248
 validate_tpa_user_id() (*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* method), 86
 validate_user_email() (*enterprise.api.v1.serializers.EnterpriseCustomerCourseEnrollmentsSerializer* method), 86
 validate_username() (*enterprise.api.v1.serializers.EnterpriseCourseEnrollmentWriteSerializer* method), 83
 validate_username() (*enterprise.api.v1.serializers.EnterpriseCustomerUserWriteSerializer* method), 90
 validated_at (*enterprise.models.EnterpriseCustomerSsoConfiguration* attribute), 177
 validated_at (*enterprise.models.HistoricalEnterpriseCustomerSsoConfiguration* attribute), 212
 ValidationMessages (class in *enterprise.utils*), 248
 VERIFIED (*enterprise.constants.CourseModes* attribute), 129
 VerifiedModeUnavailableException, 272
 verify_edx_resources() (in module *enterprise.views*), 272
 view_type_contexts (*enterprise.admin.views.TemplatePreviewView* attribute), 41
 VIEWS (*enterprise.views.RouterView* attribute), 271
 W
 widgets (*enterprise.admin.forms.AdminNotificationForm.Meta* attribute), 26
 widgets (*enterprise.admin.forms.EnterpriseCustomerReportingConfigurationAdminForm* attribute), 30
 with_access_to() (*enterprise.api.v1.views.enterprise_customer.EnterpriseCustomerView* method), 62
 with_additional_fields (*enterprise.models.EnterpriseCourseEnrollment* method), 87

attribute), [146](#)

X

xapilrsconfiguration (enter-
prise.models.EnterpriseCustomer *attribute*),
[161](#)