

SDK.finance 2.0 API Documentation

1. SDK.finance 2.0 API Documentation	11
1.1 Issuers	11
1.1.1 Create a new issuer description	11
1.1.1.1 Create a new issuer scheme	12
1.1.2 Obtain issuers description	13
1.1.2.1 Obtain issuers scheme	16
1.1.3 Update the issuer description	18
1.1.3.1 Update the issuer scheme	19
1.2 Currency	20
1.2.1 Create a new currency description	20
1.2.1.1 Create a new currency scheme	22
1.2.2 Obtain all currencies description	23
1.2.2.1 Obtain all currencies scheme	25
1.3 Organization membership management	26
1.3.1 Create an organization member description	26
1.3.1.1 Create an organization member scheme	29
1.3.2 Delete organization member description	30
1.3.2.1 Delete organization member scheme	32
1.3.3 Update organization member description	33
1.3.3.1 Update organization member scheme	35
1.4 Organization management	36
1.4.1 View organizations description	36
1.4.1.1 View organizations scheme	39
1.5 Contract management	40
1.5.1 Create a new commission profile description	40
1.5.1.1 Create a new commission profile scheme	43
1.5.2 Create a new limit profile description	44
1.5.2.1 Create a new limit profile scheme	46
1.5.3 Obtain all available operation flows description	47
1.5.3.1 Obtain all available operation flows scheme	49
1.5.4 Obtain commission profiles for specified contract description	50
1.5.4.1 Obtain commission profiles for specified contract scheme	52
1.5.5 Obtain contracts matching the specified criteria description	54
1.5.5.1 Obtain contracts matching the specified criteria scheme	56
1.5.6 Obtain limit profiles for specified commission profile description	58
1.5.6.1 Obtain limit profiles for specified commission profile scheme	59
1.5.7 Update an existing commission profile description	60
1.5.7.1 Update an existing commission profile scheme	63
1.5.8 Update an existing limit profile description	64
1.5.8.1 Update an existing limit profile scheme	66
1.5.9 View commission profile description	67
1.5.9.1 View commission profile scheme	69
1.5.10 View limit profile description	70
1.5.10.1 View limit profile scheme	71
1.6 Tariffs for changing contract\$	72
1.6.1 _Pay for transition description	72
1.6.1.1 _Pay for transition scheme	73
1.6.2 Get available contracts description	74
1.6.2.1 Get available contracts scheme	77
1.6.3 View dashboard description	79
1.6.3.1 View dashboard scheme	83
1.7 System roles management	85
1.7.1 View all available system roles description	85
1.7.1.1 View all available system roles scheme	86
1.8 User registration as supervisor	87
1.8.1 Direct registration description	87
1.8.1.1 Direct registration scheme	89
1.9 User registration	90
1.9.1 Registration description	90
1.9.1.1 Registration scheme	92
1.10 User management	94
1.10.1 Clean the ban for user description	94
1.10.1.1 Clean the ban for user scheme	95
1.10.2 Delete a user description	96
1.10.2.1 Delete a user scheme	97
1.10.3 Get profile history description	98
1.10.3.1 Get profile history scheme	100
1.10.4 Get user profile by id description	101
1.10.4.1 Get user profile by id scheme	103
1.10.5 Get users description	104
1.10.5.1 Get users scheme	107

1.10.6 Register a new user description	108
1.10.6.1 Register a new user scheme	110
1.10.7 Reset user's password and to send it to verified user contacts: phone or email description	111
1.10.7.1 Reset user's password and to send it to verified user contacts: phone or email scheme	112
1.10.8 Update user description	113
1.10.8.1 Update user scheme	119
1.10.9 Validate secret word description	121
1.10.9.1 Validate secret word scheme	122
1.11 Coin management	122
1.11.1 Create coin description	122
1.11.1.1 Create coin scheme	124
1.11.2 Delete coin description	126
1.11.2.1 Delete coin scheme	127
1.11.3 Get coins owned by current user description	128
1.11.3.1 Get coins owned by current user scheme	130
1.11.4 Set coin as main description	131
1.11.4.1 Set coin as main scheme	133
1.11.5 Toggle coin status description	134
1.11.5.1 Toggle coin status scheme	135
1.11.6 Update an existing coin description	136
1.11.6.1 Update an existing coin scheme	138
1.12 Authentication	140
1.12.1 Authentication description	140
1.12.1.1 Authentication scheme	142
1.12.2 Authentication with One Time Password (OTP) description	144
1.12.2.1 Authentication with OTP scheme	147
1.13 Password recovery	149
1.13.1 Send request to restore a password description	149
1.13.1.1 Send request to restore a password scheme	151
1.14 Profile - Master PIN	152
1.14.1 Confirmation of ban removal description	152
1.14.1.1 Confirmation of ban removal scheme	154
1.14.2 Get master PIN belonging to current user description	156
1.14.2.1 Get master PIN belonging to current user scheme	157
1.14.3 Restore an existing master PIN by generating a new one description	159
1.14.3.1 Restore an existing master PIN by generating a new one scheme	160
1.14.4 Set up a new or replace an existing master PIN description	161
1.14.4.1 Set up a new or replace an existing master PIN scheme	163
1.15 Bank accounts management	165
1.15.1 Approve bank account as compliance description	165
1.15.1.1 Approve bank account as compliance scheme	166
1.15.2 Reject bank account as compliance description	168
1.15.2.1 Reject bank account as compliance scheme	169
1.15.3 View bank accounts as compliance description	170
1.15.3.1 View bank accounts as compliance scheme	173
1.16 Validation for coin	174
1.16.1 Validate a coin description	174
1.16.1.1 Validate a coin scheme	176
1.17 Validation for user contact	178
1.17.1 Validate a contact description	178
1.17.1.1 Validate a contact scheme	179
1.18 User note API provider	181
1.18.1 Create note about user description	181
1.18.1.1 Create note about user scheme	182
1.18.2 Get all user notes description	182
1.18.2.1 Get all user notes scheme	183
1.19 Investment via cashdesk	184
1.19.1 Accept investment request 1 description	184
1.19.1.1 Accept investment request 1 scheme	187
1.19.2 Top up of authorized capital with cash description	188
1.19.2.1 Top up of authorized capital with cash scheme	191
1.19.3 View cash investment details 1 description	192
1.19.3.1 View cash investment details 1 scheme	195
1.19.4 View cash investment details2 description	196
1.19.4.1 View cash investment details2 scheme	199
1.20 Input to cash desk operations	200
1.20.1 Accept input request description	200
1.20.1.1 Accept input request scheme	203
1.20.2 Input collected cash to cash desk description	204
1.20.2.1 Input collected cash to cash desk scheme	207
1.20.3 View cash input details description	208
1.20.3.1 View cash input details scheme	210

1.21 Cashdesk - working day management	211
1.21.1 Accept closing of working day description	211
1.21.1.1 Accept closing of working day scheme	213
1.21.2 Get the cashdesk withdrawal requests which are requested for closing with Pagination description	214
1.21.2.1 Get the cashdesk withdrawal requests which are requested for closing with Pagination scheme	217
1.21.3 Send request to close a working day description	218
1.21.3.1 Send request to close a working day scheme	220
1.21.4 View the current working day description	221
1.21.4.1 View the current working day scheme	223
1.22 Cash desk operations	224
1.22.1 Get cash desks for specified coin description	224
1.22.1.1 Get cash desks for specified coin scheme	227
1.22.2 View cash desk turnover report description	228
1.22.2.1 View cash desk turnover report scheme	230
1.23 Cash desk collect	231
1.23.1 Accept collect request description	231
1.23.1.1 Accept collect request scheme	234
1.23.2 Cash collect request description	235
1.23.2.1 Cash collect request scheme	238
1.23.3 View transaction details description	239
1.23.3.1 View transaction details scheme	241
1.24 Exchange	242
1.24.1 Calculate commission for exchange operation description	242
1.24.1.1 Calculate commission for exchange operation scheme	244
1.24.2 Change currency exchange rate status to opposite description	246
1.24.2.1 Change currency exchange rate status to opposite scheme	247
1.24.3 Execute exchange operation description	249
1.24.3.1 Execute exchange operation scheme	250
1.24.4 Set currency exchange rate description	252
1.24.4.1 Set currency exchange rate scheme	253
1.24.5 View exchange rates description	255
1.24.5.1 View exchange rates scheme	256
1.25 Gate provider account management	258
1.25.1 Update settings of provider account description	258
1.25.1.1 Update settings of provider account scheme	259
1.25.2 View provider account settings description	260
1.25.2.1 View provider account settings scheme	262
1.25.3 View provider accounts description	263
1.25.3.1 View provider accounts scheme	265
1.26 Provider coin management	266
1.26.1 View provider coins of organization description	266
1.26.1.1 View provider coins of organization scheme	268
1.27 Gate investment operations	269
1.27.1 Accept investment request description	269
1.27.1.1 Accept investment request scheme	272
1.27.2 Top up of system provider coin description	273
1.27.2.1 Top up of system provider coin scheme	277
1.27.3 View cash investment details description	278
1.27.3.1 View cash investment details scheme	280
1.28 Gate - interaction with external payment systems	280
1.28.1 Filter transactions description	281
1.28.1.1 Filter transactions scheme	284
1.28.2 Step 1: retrieve list of payment providers for transaction with specified parameters description	285
1.28.2.1 Step 1: retrieve list of payment providers for transaction with specified parameters scheme	287
1.28.3 Step 2: calculate commission for execution of transaction with specified parameters description	288
1.28.3.1 Step 2: calculate commission for execution of transaction with specified parameters scheme	290
1.28.4 Step 3: create a transaction description	291
1.28.4.1 Step 3: create a transaction scheme	294
1.28.5 Step 4: retrieve list of payer fields request by payment provider description	295
1.28.5.1 Step 4: retrieve list of payer fields request by payment provider scheme	297
1.28.6 Step 5: submit payer data and send gate transaction to payment provider description	298
1.28.6.1 Step 5: submit payer data and send gate transaction to payment provider scheme	301
1.28.7 Step 6: get transaction state description	302
1.28.7.1 Step 6: get transaction state scheme	305
1.29 Gate contract management	306
1.29.1 Create a new exchange settings record description	306
1.29.1.1 Create a new exchange settings record scheme	308
1.29.2 Create a new gate commission profile description	309
1.29.2.1 Create a new gate commission profile scheme	311
1.29.3 Create a new gate limit profile description	312
1.29.3.1 Create a new gate limit profile scheme	315
1.29.4 Obtain gate limit profiles for specified gate commission profile description	316

1.29.4.1 Obtain gate limit profiles for specified gate commission profile scheme	318
1.29.5 Set up a commission settings description	319
1.29.5.1 Set up a commission settings scheme	322
1.29.6 Set up a product commission settings description	323
1.29.6.1 Set up a product commission settings scheme	326
1.29.7 Update an existing limit profile 1 description	327
1.29.7.1 Update an existing limit profile 1 scheme	330
1.29.8 View commission settings of gate commission profile description	330
1.29.8.1 View commission settings of gate commission profile scheme	332
1.29.9 View exchange settings of gate commission profile description	333
1.29.9.1 View exchange settings of gate commission profile scheme	335
1.29.10 View gate commission profile description	336
1.29.10.1 View gate commission profile scheme	338
1.29.11 View gate commission profiles for specified contract description	339
1.29.11.1 View gate commission profiles for specified contract scheme	341
1.29.12 View gate limit profile description	342
1.29.12.1 View gate limit profile scheme	344
1.29.13 View product commission settings of gate commission profile description	345
1.29.13.1 View product commission settings of gate commission profile scheme	347
1.30 Gate purchases - pay for external services with your wallet	348
1.30.1 Filter products available for purchase description	348
1.30.1.1 Filter products available for purchase scheme	351
1.30.2 Get all products for gate provider account description	352
1.30.2.1 Get all products for gate provider account scheme	354
1.30.3 Get payer fields for specific product description	355
1.30.3.1 Get payer fields for specific product scheme	357
1.30.4 Step 1: get list of all available products description	358
1.30.4.1 Step 1: get list of all available products scheme	360
1.30.5 Step 2: calculate commission amount description	361
1.30.5.1 Step 2: calculate commission amount scheme	363
1.30.6 Step 3: create purchase for selected product description	365
1.30.6.1 Step 3: create purchase for selected product scheme	367
1.31 Prepaid	368
1.31.1 Activate prepaid coin description	368
1.31.1.1 Activate prepaid coin scheme	372
1.31.2 Calculate commission for prepaid creation description	374
1.31.2.1 Calculate commission for prepaid creation scheme	375
1.31.3 Calculate commission for redeem of the prepaid description	377
1.31.3.1 Calculate commission for redeem of the prepaid scheme	378
1.31.4 Change pin for prepaid description	380
1.31.4.1 Change pin for prepaid scheme	381
1.31.5 Create a prepaid description	381
1.31.5.1 Create a prepaid scheme	383
1.31.6 View prepaid vouchers description	385
1.31.6.1 View prepaid vouchers scheme	386
1.32 Profile - bank accounts	388
1.32.1 Create a new bank account description	388
1.32.1.1 Create a new bank account scheme	389
1.32.2 Delete a bank account description	391
1.32.2.1 Delete a bank account scheme	391
1.32.3 Update a bank account description	393
1.32.3.1 Update a bank account scheme	394
1.32.4 View profile's bank accounts description	396
1.32.4.1 View profile's bank accounts scheme	397
1.33 Profile - documents approving	399
1.33.1 Approve the profile document description	399
1.33.1.1 Approve the profile document scheme	400
1.33.2 Decline the profile document description	402
1.33.2.1 Decline the profile document scheme	403
1.33.3 View all profile documents description	405
1.33.3.1 View all profile documents scheme	406
1.34 Profile - documents uploading	408
1.34.1 Get all document types required for approval description	408
1.34.1.1 Get all document types required for approval scheme	409
1.34.2 Get profile documents, last of each type description	411
1.34.2.1 Get profile documents, last of each type scheme	412
1.34.3 Submit file for approval as profile document description	414
1.34.3.1 Submit file for approval as profile document scheme	415
1.35 Profile - documents verification	417
1.35.1 Verify a profile document description	417
1.35.1.1 Verify a profile document scheme	418
1.35.2 View forgery tests of verified document description	420

1.35.2.1 View forgery tests of verified document scheme	421
1.35.3 View last verified documents description	423
1.35.3.1 View last verified documents scheme	424
1.35.4 View not verified documents description	426
1.35.4.1 View not verified documents scheme	427
1.36 Media files management	429
1.36.1 Delete an unused file description	429
1.36.1.1 Delete an unused file scheme	429
1.36.2 Get media files by specified filter description	430
1.36.2.1 Get media files by specified filter scheme	433
1.36.3 Update a file description	434
1.36.3.1 Update a file scheme	436
1.36.4 Upload a file description	437
1.36.4.1 Upload a file scheme	439
1.37 Profile - identification	440
1.37.1 Approve organization's identification description	440
1.37.1.1 Approve organization's identification scheme	444
1.37.2 Decline a reset identification request of user description	446
1.37.2.1 Decline a reset identification request of user scheme	447
1.37.3 Decline organization's identification description	449
1.37.3.1 Decline organization's identification scheme	452
1.37.4 Reset an identification of user and send mandatory fields of profile to update for reidentification description	454
1.37.4.1 Reset an identification of user and send mandatory fields of profile to update for reidentification scheme	457
1.37.5 Reset an identification of user description	459
1.37.5.1 Reset an identification of user scheme	460
1.37.6 View pending reset identification requests description	462
1.37.6.1 View pending reset identification requests scheme	463
1.38 Profile - information	465
1.38.1 Confirm contact verification request using one time password description	465
1.38.1.1 Confirm contact verification request using one time password scheme	468
1.38.2 Create a reset identification request description	469
1.38.2.1 Create a reset identification request scheme	471
1.38.3 Create contact verification request description	473
1.38.3.1 Create contact verification request scheme	474
1.38.4 Get user profile description	475
1.38.4.1 Get user profile scheme	479
1.38.5 Multiple profile updates description	481
1.38.5.1 Multiple profile updates scheme	488
1.38.6 Resend one time password to confirm contact description	490
1.38.6.1 Resend one time password to confirm contact scheme	491
1.39 Profile - transactions confirmation	492
1.39.1 Confirm update of transaction confirmation settings record description	493
1.39.1.1 Confirm update of transaction confirmation settings record scheme	494
1.39.2 Create transaction confirmation settings record description	495
1.39.2.1 Create transaction confirmation settings record scheme	497
1.39.3 Get current transactions confirmation settings description	499
1.39.3.1 Get current transactions confirmation settings scheme	500
1.39.4 Request update of transaction confirmation settings record description	502
1.39.4.1 Request update of transaction confirmation settings record scheme	503
1.39.5 Retrieve list of business process types eligible for confirmations set up description	504
1.39.5.1 Retrieve list of business process types eligible for confirmations set up scheme	505
1.40 Top up via bank	507
1.40.1 Create a request to top up via bank description	507
1.40.1.1 Create a request to top up via bank scheme	509
1.40.2 Accept to create request to top up via bank description	510
1.40.2.1 Accept to create request to top up via bank scheme	516
1.40.3 Decline to create request to top up via bank description	517
1.40.3.1 Decline to create request to top up via bank scheme	521
1.40.4 Calculate commission for top up via bank description	522
1.40.4.1 Calculate commission for top up via bank scheme	524
1.40.5 Perform top up via bank description	525
1.40.5.1 Perform top up via bank scheme	529
1.41 Top up via cash desk	531
1.41.1 Calculate commission for top-up via cash desk description	531
1.41.1.1 Calculate commission for top-up via cash desk scheme	532
1.41.2 Top up via cash desk description	533
1.41.2.1 Top up via cash desk scheme	538
1.42 Withdrawal via bank	540
1.42.1 Calculate commission for withdrawal via bank description	540
1.42.1.1 Calculate commission for withdrawal via bank scheme	541
1.42.2 Create a request to withdraw via bank description	543
1.42.2.1 Create a request to withdraw via bank scheme	548

1.42.3 Lift limit for withdrawal via bank description	550
1.42.3.1 Lift limit for withdrawal via bank scheme	554
1.43 Withdraw via cash desk	555
1.43.1 Calculate commission for withdrawal via specified cash desk description	555
1.43.1.1 Calculate commission for withdrawal via specified cash desk scheme	558
1.43.2 Execute withdrawal via specified cash desk description	560
1.43.2.1 Execute withdrawal via specified cash desk scheme	565
1.43.3 Withdraw via cash desk description	567
1.43.3.1 Withdraw via cash desk scheme	574
1.44 Transfer	576
1.44.1 Calculate commission fee for transfer description	576
1.44.1.1 Calculate commission fee for transfer scheme	578
1.44.2 Execute transfer transaction description	579
1.44.2.1 Execute transfer transaction scheme	583
1.45 Transactions	585
1.45.1 Get user transactions description	585
1.45.1.1 Get user transactions scheme	589
1.46 Card management	590
1.46.1 Delete card from list description	590
1.46.1.1 Delete card from list scheme	591
1.47 Card operations	592
1.47.1 Return funds for a completed card payment description	592
1.47.1.1 Return funds for a completed card payment scheme	594
1.48 Hold funds on a payment card	595
1.48.1 _Release card hold description	595
1.48.1.1 _Release card hold scheme	598
1.48.2 Add invoice hold payment settings description	599
1.48.2.1 Add invoice hold payment settings scheme	601
1.48.3 Capture card hold description	602
1.48.3.1 Capture card hold scheme	605
1.48.4 Create card hold for an invoice description	606
1.48.4.1 Create card hold for an invoice scheme	609
1.48.5 Create card hold for an invoice with new card description	610
1.48.5.1 Create card hold for an invoice with new card scheme	613
1.48.6 View invoice hold payment settings description	614
1.48.6.1 View invoice hold payment settings scheme	617
1.48.7 View list of card hold items description	618
1.48.7.1 View list of card hold items scheme	622
1.49 Smart cards management - as supervisor	623
1.49.1 Create a new smart card description	623
1.49.1.1 Create a new smart card scheme	625
1.49.2 Delete an existing smart card description	626
1.49.2.1 Delete an existing smart card scheme	628
1.49.3 Get smart cards by specified filter description	629
1.49.3.1 Get smart cards by specified filter scheme	632
1.49.4 UPDATE AN EXISTING SMART CARD description	633
1.49.4.1 UPDATE AN EXISTING SMART CARD scheme	635
1.50 Smart cards viewing - as client	636
1.50.1 GET SMART CARDS OWNED BY CURRENT USER description	636
1.50.1.1 GET SMART CARDS OWNED BY CURRENT USER scheme	638
1.51 Validation for smart card	639
1.51.1 Validate a smart card description	639
1.51.1.1 Validate a smart card scheme	641
1.52 Paysafe cards	642
1.52.1 Retrieve all cards available for user description	642
1.52.1.1 Retrieve all cards available for user scheme	644
1.53 Subscriptions	646
1.53.1 Activate subscription description	646
1.53.1.1 Activate subscription scheme	647
1.53.2 Create subscription description	648
1.53.2.1 Create subscription scheme	651
1.53.3 DELETE SUBSCRIPTION description	652
1.53.3.1 DELETE SUBSCRIPTION scheme	654
1.53.4 Get scheduled payments for the period description	655
1.53.4.1 Get scheduled payments for the period scheme	657
1.53.5 Get subscriptions for current user with Filter And Pagination description	658
1.53.5.1 Get subscriptions for current user with Filter And Pagination scheme	661
1.53.6 GET SUBSCRIPTIONS OWNED BY CURRENT USER description	662
1.53.6.1 GET SUBSCRIPTIONS OWNED BY CURRENT USER scheme	664
1.53.7 Stop subscription description	665
1.53.7.1 Stop subscription scheme	667
1.54 Mass payment management description	668

1.54.1 Mass payment management scheme	670
1.55 Point of sale	672
1.55.1 Create point of sale description	672
1.55.1.1 Create point of sale scheme	673
1.55.2 Delete point of sale description	674
1.55.2.1 Delete point of sale scheme	675
1.55.3 Generate new secret code for point of sale description	676
1.55.3.1 Generate new secret code for point of sale scheme	678
1.55.4 Get information about single point of sale description	679
1.55.4.1 Get information about single point of sale scheme	681
1.55.5 Get merchant payment transactions for specified POS description	682
1.55.5.1 Get merchant payment transactions for specified POS scheme	688
1.55.6 Get points of sale description	689
1.55.6.1 Get points of sale scheme	691
1.55.7 Update point of sale description	692
1.55.7.1 Update point of sale scheme	694
1.56 Invoice	695
1.56.1 Calculate commission for invoice as merchant description	695
1.56.1.1 Calculate commission for invoice as merchant scheme	697
1.56.2 Calculate commission for specified invoice as payer description	699
1.56.2.1 Calculate commission for specified invoice as payer scheme	700
1.56.3 Create an invoice description	702
1.56.3.1 Create an invoice scheme	704
1.56.4 Delete invoice description	705
1.56.4.1 Delete invoice scheme	706
1.56.5 Get file attached to the invoice description	707
1.56.5.1 Get file attached to the invoice scheme	709
1.56.6 Get information about invoice by identifier description	710
1.56.6.1 Get information about invoice by identifier scheme	712
1.56.7 Pay for the invoice description	713
1.56.7.1 Pay for the invoice scheme	715
1.56.8 View invoices description	716
1.56.8.1 View invoices scheme	719
1.57 Merchant Payment	720
1.57.1 Calculate commission fee description	720
1.57.1.1 Calculate commission fee scheme	722
1.57.2 Create a payment description	723
1.57.2.1 Create a payment scheme	727
1.57.3 Delete a pending payment description	728
1.57.3.1 Delete a pending payment scheme	729
1.57.4 Get payments description	730
1.57.4.1 Get payments scheme	734
1.57.5 Get purchases completed within payment description	735
1.57.5.1 Get purchases completed within payment scheme	736
1.57.6 Pay for payment description	737
1.57.6.1 Pay for payment scheme	740
1.57.7 Resend one time password description	741
1.57.7.1 Resend one time password scheme	742
1.58 Merchant payment via external payment system	743
1.58.1 Create gate transaction for merchant payment description	743
1.58.1.1 Create gate transaction for merchant payment scheme	746
1.59 Templates	747
1.59.1 Add new purchase template to a cart template description	748
1.59.1.1 Add new purchase template to a cart template scheme	752
1.59.2 Create a cart template description	753
1.59.2.1 Create a cart template scheme	758
1.59.3 Create a transfer template description	759
1.59.3.1 Create a transfer template scheme	761
1.59.4 Delete a transfer template description	762
1.59.4.1 Delete a transfer template scheme	763
1.59.5 Delete cart template description	764
1.59.5.1 Delete cart template scheme	765
1.59.6 Delete purchase template from a cart template description	766
1.59.6.1 Delete purchase template from a cart template scheme	767
1.59.7 Get cart templates for current user description	768
1.59.7.1 Get cart templates for current user scheme	770
1.59.8 Get transfer templates for current user description	771
1.59.8.1 Get transfer templates for current user scheme	773
1.59.9 Get transfer templates for current user with Filter And Pagination description	774
1.59.9.1 Get transfer templates for current user with Filter And Pagination scheme	777
1.59.10 Payment by cart template description	778
1.59.10.1 Payment by cart template scheme	781

1.59.11 Update a cart template description	782
1.59.11.1 Update a cart template scheme	783
1.59.12 Update a transfer template description	784
1.59.12.1 Update a transfer template scheme	786
1.60 Invoice Template	787
1.60.1 Create an invoice template description	787
1.60.1.1 Create an invoice template scheme	790
1.60.2 Delete an invoice template description	791
1.60.2.1 Delete an invoice template scheme	793
1.60.3 Update invoice template description	794
1.60.3.1 Update invoice template scheme	797
1.60.4 View templates description	798
1.60.4.1 View templates scheme	801
1.61 Merchant product categories	802
1.61.1 Add product into category description	802
1.61.1.1 Add product into category scheme	804
1.61.2 Create a product category description	806
1.61.2.1 Create a product category scheme	808
1.61.3 Delete a product category description	809
1.61.3.1 Delete a product category scheme	811
1.61.4 Remove product from category description	812
1.61.4.1 Remove product from category scheme	814
1.61.5 Update a product category scheme	815
1.61.5.1 Update a product category description	817
1.61.6 View product categories belonging to this merchant description	818
1.61.6.1 View product categories belonging to this merchant scheme	820
1.61.7 View products belonging to category description	821
1.61.7.1 View products belonging to category scheme	823
1.62 Merchant product	824
1.62.1 Create one product description	825
1.62.1.1 Create one product scheme	827
1.62.2 Create products from batch description	828
1.62.2.1 Create products from batch scheme	831
1.62.3 Delete by external code description	832
1.62.3.1 Delete by external code scheme	834
1.62.4 Delete by ID description	835
1.62.4.1 Delete by ID scheme	837
1.62.5 Delete list of products description	838
1.62.5.1 Delete list of products scheme	840
1.62.6 Update product by it's external code description	841
1.62.6.1 Update product by it's external code scheme	844
1.62.7 Update product by it's identifier description	845
1.62.7.1 Update product by it's identifier scheme	848
1.62.8 View categories specified product belongs to description	849
1.62.8.1 View categories specified product belongs to scheme	851
1.62.9 View product by it's ID description	852
1.62.9.1 View product by it's ID scheme	854
1.62.10 View product list description	856
1.62.10.1 View product list scheme	858
1.63 Merchant product measure units	859
1.63.1 Create a measure unit description	860
1.63.1.1 Create a measure unit scheme	862
1.63.2 Create a measure unit from batch description	862
1.63.2.1 Create a measure unit from batch scheme	865
1.63.3 Delete a measure unit by it's external code description	866
1.63.3.1 Delete a measure unit by it's external code scheme	868
1.63.4 Delete a measure unit by it's ID description	869
1.63.4.1 Delete a measure unit by it's ID scheme	871
1.63.5 Get list of all measure units belonging to this merchant description	872
1.63.5.1 Get list of all measure units belonging to this merchant scheme	874
1.63.6 Update measure unit data by it's external code description	875
1.63.6.1 Update measure unit data by it's external code scheme	877
1.63.7 Update measure unit data by it's ID description	878
1.63.7.1 Update measure unit data by it's ID scheme	880
1.64 Merchant product price	881
1.64.1 Create prices from batch description	881
1.64.1.1 Create prices from batch scheme	884
1.64.2 Delete a price by it's ID for the given product ID description	885
1.64.2.1 Delete a price by it's ID for the given product ID scheme	887
1.64.3 Delete a price by product filter description	888
1.64.3.1 Delete a price by product filter scheme	890
1.64.4 Delete all prices for product and point of sale description	891

1.64.4.1 Delete all prices for product and point of sale scheme	893
1.64.5 Get prices by product filter description	894
1.64.5.1 Get prices by product filter scheme	897
1.64.6 Get prices by product ID description	898
1.64.6.1 Get prices by product ID scheme	900
1.64.7 Add price for a product with the given ID description	902
1.64.7.1 Add price for a product with the given ID scheme	904
1.65 Loyalty groups management	905
1.65.1 Create a new loyalty group description	905
1.65.1.1 Create a new loyalty group scheme	907
1.65.2 Delete loyalty group description	908
1.65.2.1 Delete loyalty group scheme	910
1.65.3 Get loyalty groups description	911
1.65.3.1 Get loyalty groups scheme	913
1.65.4 Get profile fields eligible to use during loyalty group creation description	914
1.65.4.1 Get profile fields eligible to use during loyalty group creation scheme	916
1.66 Loyalty rule management	917
1.66.1 Add issuer to the loyalty rule description	917
1.66.1.1 Add issuer to the loyalty rule scheme	919
1.66.2 Add loyalty group to the loyalty rule description	920
1.66.2.1 Add loyalty group to the loyalty rule scheme	922
1.66.3 Add merchant product category to the loyalty rule description	923
1.66.3.1 Add merchant product category to the loyalty rule scheme	925
1.66.4 Add merchant product to the loyalty rule description	927
1.66.4.1 Add merchant product to the loyalty rule scheme	928
1.66.5 Add point of sale to the loyalty rule description	929
1.66.5.1 Add point of sale to the loyalty rule scheme	931
1.66.6 Create a new loyalty rule description	932
1.66.6.1 Create a new loyalty rule scheme	934
1.66.7 Find loyalty rules eligible for current user description	935
1.66.7.1 Find loyalty rules eligible for current user scheme	937
1.66.8 Remove issuer from loyalty rule description	938
1.66.8.1 Remove issuer from loyalty rule scheme	940
1.66.9 Remove loyalty group from loyalty rule description	941
1.66.9.1 Remove loyalty group from loyalty rule scheme	943
1.66.10 Remove merchant product category from loyalty rule description	944
1.66.10.1 Remove merchant product category from loyalty rule scheme	946
1.66.11 Remove merchant product from loyalty rule description	947
1.66.11.1 Remove merchant product from loyalty rule scheme	949
1.66.12 Remove point of sale from loyalty rule description	950
1.66.12.1 Remove point of sale from loyalty rule scheme	952
1.66.13 Retrieve media files attached to the loyalty rule description	953
1.66.13.1 Retrieve media files attached to the loyalty rule scheme	955
1.66.14 Retrieve merchant's loyalty rules description	956
1.66.14.1 Retrieve merchant's loyalty rules scheme	958
1.66.15 Update an existing loyalty rule description	959
1.66.15.1 Update an existing loyalty rule scheme	961
1.66.16 View issuers used by loyalty rule description	962
1.66.16.1 View issuers used by loyalty rule scheme	964
1.66.17 View loyalty groups used by loyalty rule description	965
1.66.17.1 View loyalty groups used by loyalty rule scheme	967
1.66.18 View merchant products categories used by loyalty rule description	968
1.66.18.1 View merchant products categories used by loyalty rule scheme	970
1.66.19 View merchant products used by loyalty rule description	971
1.66.19.1 View merchant products used by loyalty rule scheme	973
1.66.20 View merchant's loyalty rules description	974
1.66.20.1 View merchant's loyalty rules scheme	977
1.66.21 View points of sale used by loyalty rule description	977
1.66.21.1 View points of sale used by loyalty rule scheme	979
1.67 Loyalty rule activations	980
1.67.1 Get activations of the loyalty rules occurred during merchant payment description	980
1.67.1.1 Get activations of the loyalty rules occurred during merchant payment scheme	982
1.68 I18n - management	983
1.68.1 Create a new I18n record description	984
1.68.1.1 Create a new I18n record scheme	985
1.68.2 Delete a I18n record description	987
1.68.2.1 Delete a I18n record scheme	987
1.68.3 Download all properties in excel file description	989
1.68.3.1 Download all properties in excel file scheme	989
1.68.4 Update a I18n record description	991
1.68.4.1 Update a I18n record scheme	991
1.68.5 Upload all properties from excel file description	993

1.68.5.1 Upload all properties from excel file scheme	993
1.68.6 View all I18n records by locale description	995
1.68.6.1 View all I18n records by locale scheme	996
1.68.7 View all I18n records description	998
1.68.7.1 View all I18n records scheme	999
1.68.8 View all locales description	1001
1.68.8.1 View all locales description scheme	1002
1.68.9 View I18n records description	1004
1.68.9.1 View I18n records description scheme	1005
1.69 Support	1007
1.69.1 Create a ticket to customer support description	1007
1.69.1.1 Create a ticket to customer support scheme	1008
1.69.2 Retrieve list of tickets as customer support manager description	1010
1.69.2.1 Retrieve list of tickets as customer support manager scheme	1012
1.70 User message API provider	1014
1.70.1 Create and send message to user description	1014
1.70.1.1 Create and send message to user scheme	1015
1.70.2 Get all messages to user description	1016
1.70.2.1 Get all messages to user scheme	1018
1.71 Callback settings for non-merchant operations	1019
1.71.1 Create or update callback settings description	1019
1.71.1.1 Create or update callback settings scheme	1021
1.71.2 Generate secret code for callback settings description	1022
1.71.2.1 Generate secret code for callback settings scheme	1023
1.71.3 View callback settings description	1024
1.71.3.1 View callback settings scheme	1025
1.72 In-system communication	1026
1.72.1 Create system broadcasts description	1027
1.72.1.1 Create system broadcasts scheme	1029
1.72.2 Delete a conversation description	1029
1.72.2.1 Delete a conversation scheme	1030
1.72.3 Get files attached to the specified message description	1031
1.72.3.1 Get files attached to the specified message scheme	1033
1.72.4 Mark message as read description	1034
1.72.4.1 Mark message as read scheme	1036
1.72.5 Retrieve list of conversations by specified filter description	1037
1.72.5.1 Retrieve list of conversations by specified filter scheme	1040
1.72.6 Retrieve list of messages for specified conversation description	1041
1.72.6.1 Retrieve list of messages for specified conversation scheme	1044
1.72.7 Write a message as participant of the specified conversation description	1044
1.72.7.1 Write a message as participant of the specified conversation scheme	1046
1.73 Environment variables management	1047
1.73.1 Get list of environment variables description	1047
1.73.1.1 Get list of environment variables scheme	1049
1.73.2 Update multiple variables description	1051
1.73.2.1 Update multiple variables scheme	1052
1.74 Error codes	1054
1.74.1 Get list of error codes description	1054
1.74.1.1 Get list of error codes scheme	1055
1.75 Postman Collections	1057

SDK.finance 2.0 API Documentation

In this documentation all API consist from:

- Basic Description
- Direct API using flow scheme
- Web implementation of API calls scheme
- Request and Return parameters
- Postman collection for <https://www.getpostman.com/>

Issuers

Create a new issuer description

Use Case Name

Create a new issuer

Brief Description

A User or External Entity on behalf of a User with role permission ISSUER_MANAGER will go through all steps of “Obtain issuers” Use Case, and then send a request to Endpoint “Create a new issuer”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ISSUER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain issuers”.
2. External Entity sends a request to Endpoint “Create a new issuer”.

Endpoint URL: POST /issuers

Parameters:

```
{
  "snPrefix": "string",
  "currencyCode": "string",
  "name": "string",
  "description": "string",
  "active": false,
  "orderNumber": 0,
  "orderQuote": 0
}
```

3. System Operator returns new Issuer information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.
2. A user sends a request to Endpoint “Create a new issuer”.

Endpoint URL: POST /issuers

Parameters:

```
{
  "snPrefix": "string",
  "currencyCode": "string",
  "name": "string",
  "description": "string",
  "active": false,
  "orderNumber": 0,
  "orderQuote": 0
}
```

3. System Operator returns new Issuer information to User (See Result example below).

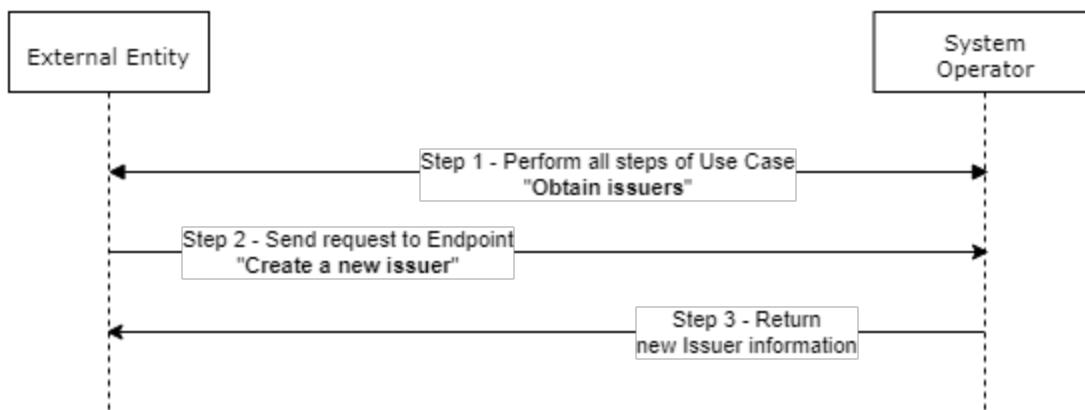
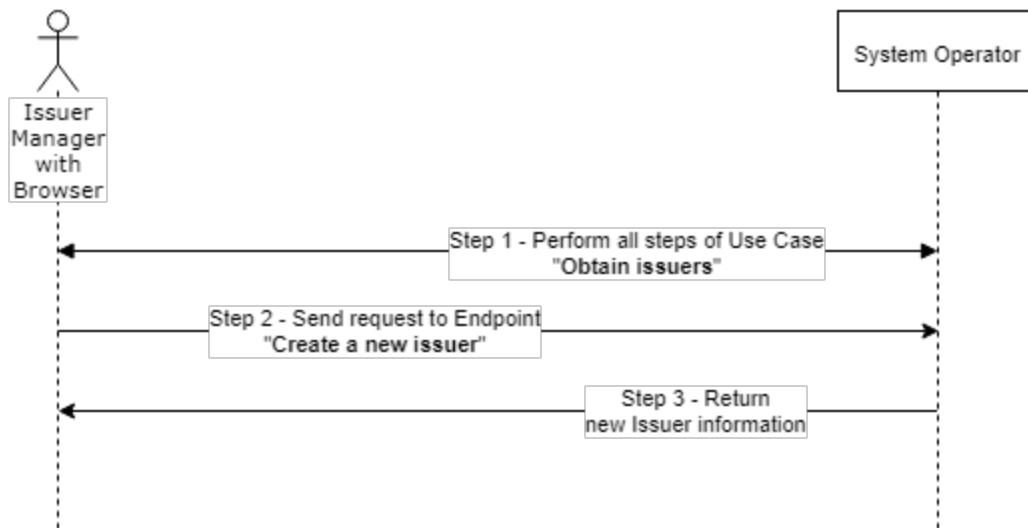
Post Conditions

new Issuer is available.

Result example

```
{
  "issuer": {
    "id": "string",
    "sn": "string",
    "name": "string",
    "description": "string",
    "orderNumber": 0,
    "orderQuote": 0,
    "active": false,
    "currency": {
      "code": "string",
      "digitalCode": "string",
      "symbol": "string",
      "name": "string",
      "description": "string"
    }
  },
  "status": "ok",
  "message": "string"
}
```

Create a new issuer scheme

Use case: Create a new issuer**Basic FFlow****Optional Web UI Flow**

Obtain issuers description

Use Case Name

Obtain issuers

Brief Description

Use Case allows an External Entity or an Individual to request a list of available Issuers.

The issuer is the payment unit of the system - the types of units/currencies that are available in the system.

It can be a separate electronic currency, with reference to real currencies or without it, also bonuses units, etc. Currently, we have Issuers which are connected to real types of currencies: USD, EUR, GBP etc. (obtain, create, update issuers)

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with permissions: ISSUER_VIEWER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to get a list all Issuers.

Endpoint URL: GET /issuers

Parameter: Security TOKEN.

3. System Operator returns a List of Issuers to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. The user sends a request to get a list of all Issuers.

Endpoint URL: GET /issuers

Parameter: Security TOKEN.

3. System Operator returns a web page with a List of all Issuers to User.

Post Conditions

External Entity or User can see Issuers and corresponding Currencies.

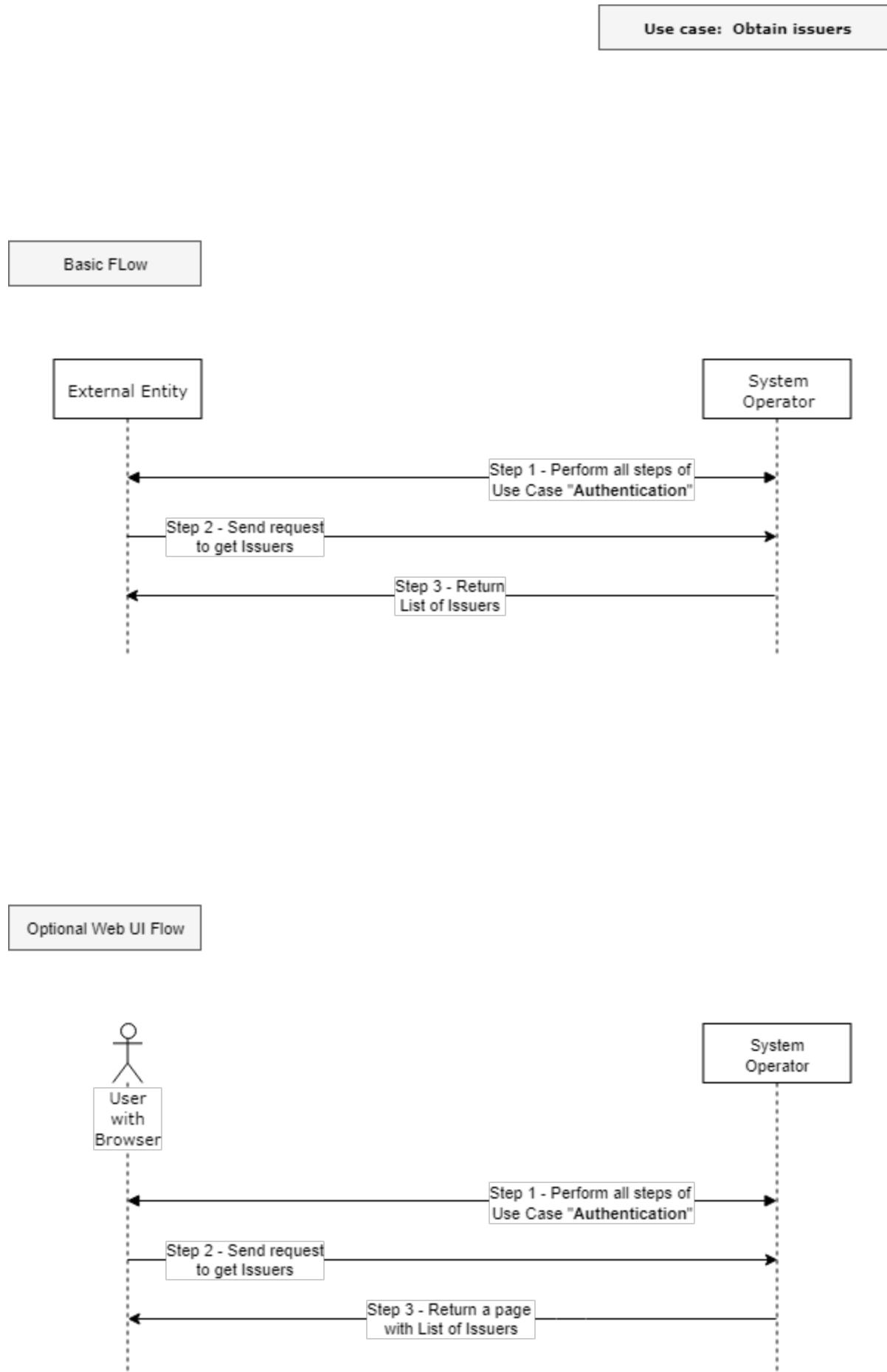
Result example

```
{
  "records": [
    {
      "id": "add864c2-0d26-4bc2-a459-7fe51f501021",
      "sn": "USD",
      "name": "USD issuer",
      "description": "desc",
      "orderNumber": 100,
      "orderQuote": 100,
      "active": true,
      "currency": {
        "code": "USD",
        "symbol": "\u20ac"
      }
    }
  ]
}
```

```
        "digitalCode": "840",
        "symbol": "$",
        "name": "US Dollar",
        "description": null
    }
},
{
    "id": "ae8fa896-750a-4459-a7a6-e6a3ff2852ee",
    "sn": "EUR",
    "name": "EUR issuer",
    "description": "desc",
    "orderNumber": 200,
    "orderQuote": 200,
    "active": true,
    "currency": {
        "code": "EUR",
        "digitalCode": "978",
        "symbol": "€",
        "name": "Euro",
        "description": null
    }
},
{
    "id": "2b51bbc0-e112-4723-86c6-7bcb966d6846",
    "sn": "UAH",
    "name": "UAH issuer",
    "description": "desc",
    "orderNumber": 300,
    "orderQuote": 100,
    "active": true,
    "currency": {
        "code": "UAH",
        "digitalCode": "980",
        "symbol": " ",
        "name": "Ukrainian Hryvnia",
        "description": null
    }
}
```

```
        }  
    ]  
}
```

Obtain issuers scheme



Update the issuer description

Use Case Name

Update the issuer

Brief Description

A User or External Entity on behalf of a User with role permission ISSUER_MANAGER will go through all steps of “Obtain issuers” Use Case, and then send a request to Endpoint “Update the issuer”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ISSUER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain issuers”.
2. External Entity sends a request to Endpoint “Update the issuer”.

Endpoint URL: PATCH /issuers/{id}

Parameters:

```
{
  "name": "string",
  "description": "string",
  "active": false,
  "orderNumber": 0,
  "orderQuote": 0
}
```

3. System Operator returns updated Issuer information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.
2. A user sends a request to Endpoint “Update the issuer”.

Endpoint URL: PATCH /issuers/{id}

Parameters:

```
{  
    "name": "string",  
    "description": "string",  
    "active": false,  
    "orderNumber": 0,  
    "orderQuote": 0  
}
```

3. System Operator returns updated Issuer information to User (See Result example below).

Post Conditions

Updated Issuer information is available.

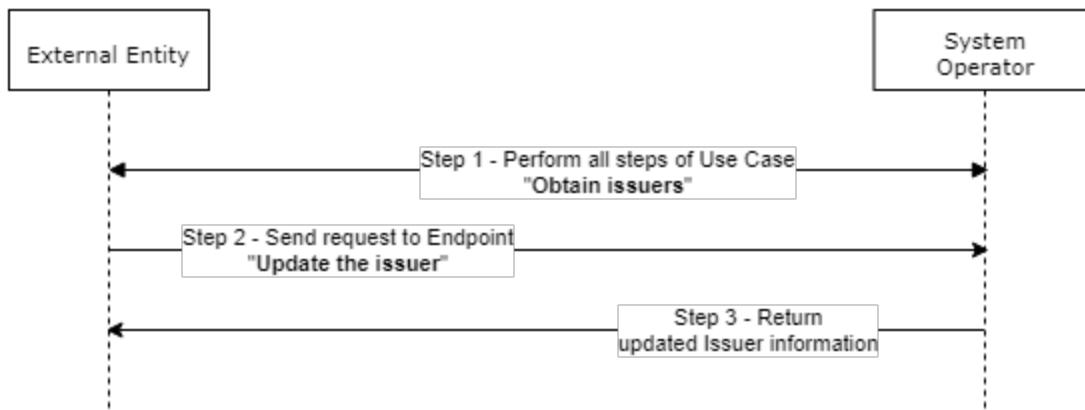
Result example

```
{  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "name": "string",  
        "description": "string",  
        "orderNumber": 0,  
        "orderQuote": 0,  
        "active": false,  
        "currency": {  
            "code": "string",  
            "digitalCode": "string",  
            "symbol": "string",  
            "name": "string",  
            "description": "string"  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

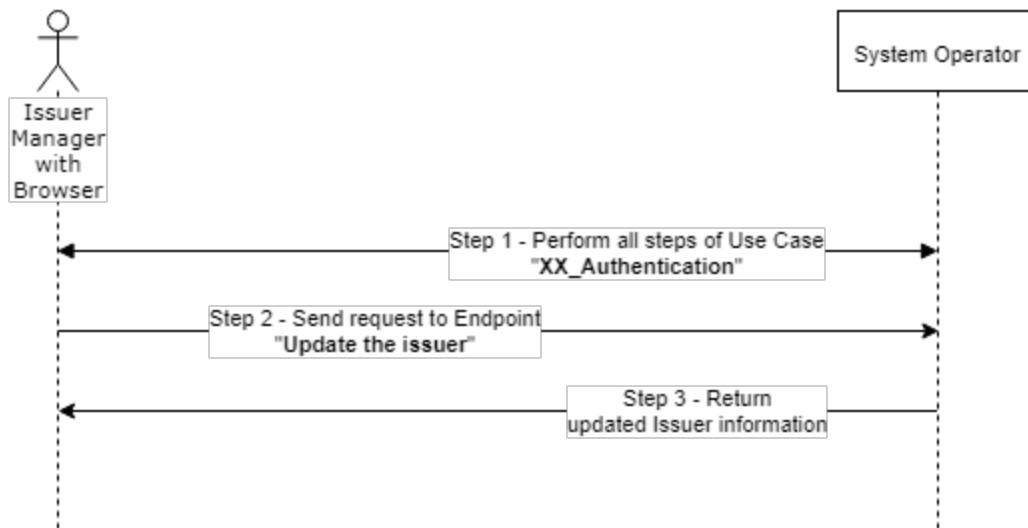
Update the issuer scheme

Use case: Update the issuer

Basic FFlow



Optional Web UI Flow



Currency

Create a new currency description

Use Case Name

Create a new currency

Brief Description

Authenticated System User or External Entity acting as a User with corresponding access privileges CURRENCY_MANAGER connects to the System, executes all steps of Authentication and “Obtain all currencies” Use Cases”. Then the request could be sent to add to the System a currency that is not currently managed by the System.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with corresponding access privileges CURRENCY_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with enough privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain all currencies”.
2. External Entity sends a request to Endpoint “Create a new currency”.

Endpoint URL: POST /currencies

Parameter:

```
{
  "code": "string",
  "digitalCode": "string",
  "symbol": "string",
  "name": "string",
  "description": "string"
}
```

3. System Operator returns a result confirmation. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain all currencies”.
2. A user sends a request to Endpoint “Create a new currency”.

Endpoint URL: POST /currencies

Parameter:

```
{
  "code": "string",
  "digitalCode": "string",
  "symbol": "string",
  "name": "string",
  "description": "string"
}
```

3. System Operator returns a result confirmation. (See Result example below)

Post Conditions

Currency added to the System is available for transactions.

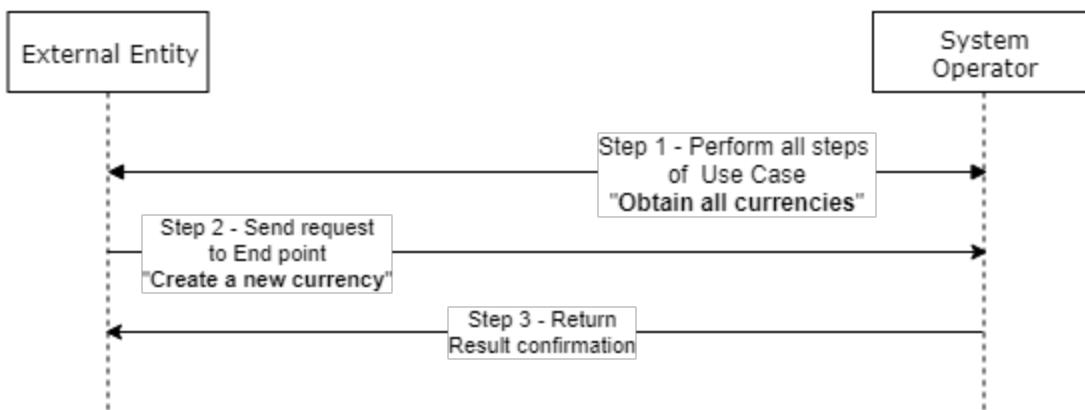
Result example

```
{  
  "currency": {  
    "code": "string",  
    "digitalCode": "string",  
    "symbol": "string",  
    "name": "string",  
    "description": "string"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

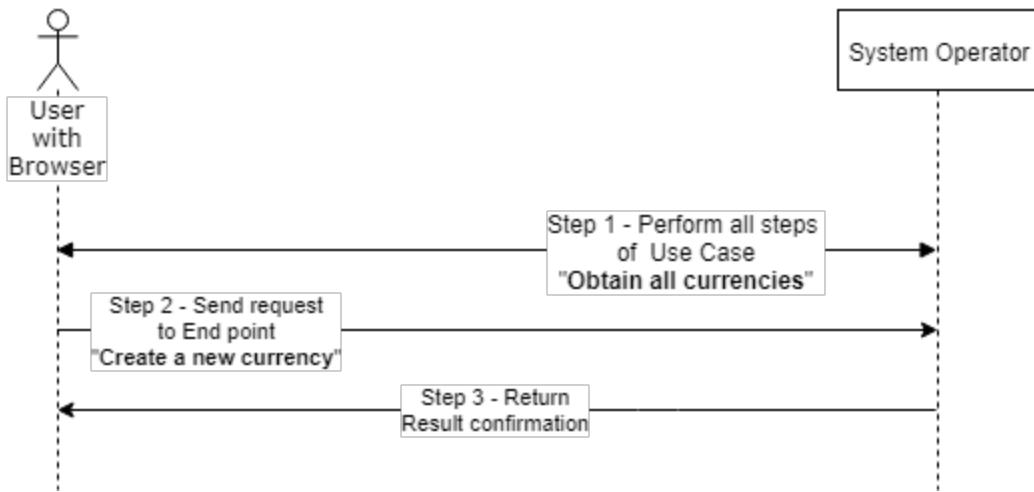
Create a new currency scheme

Use case: Create a new currency

Basic FLow



Optional Web UI Flow



Obtain all currencies description

Use Case Name

Obtain all currencies

Brief Description

After executing all steps of “Authentication” Use Case a User with privileges CURRENCY_VIEWER, CURRENCY_MANAGER requests a List of currencies in the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with privileges CURRENCY_VIEWER, CURRENCY_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.
2. The user must have enough access privileges to see currencies.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Obtain all currencies”.

Endpoint URL: GET /currencies

Parameter: Security TOKEN.

3. System Operator returns a List of available currencies. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Obtain all currencies”.

Endpoint URL: GET /currencies

Parameter: Security TOKEN.

3. System Operator returns a List of available currencies. (See Result example below)

Post Conditions

List of currencies is available.

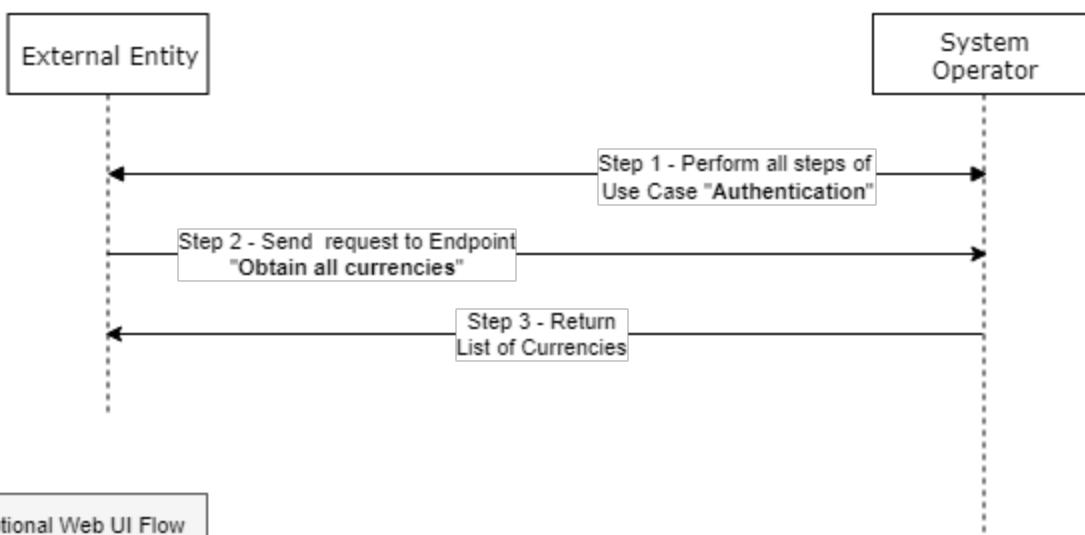
Result example

```
{  
  "currencies": [  
    {  
      "code": "USD",  
      "digitalCode": "840",  
      "symbol": "$",  
      "name": "US Dollar",  
      "description": null  
    },  
    {  
      "code": "EUR",  
      "digitalCode": "978",  
      "symbol": "€",  
      "name": "Euro",  
      "description": null  
    },  
    {  
      "code": "UAH",  
      "digitalCode": "980",  
      "symbol": "",  
      "name": "Ukrainian Hryvnia",  
      "description": null  
    }  
  ],  
  "status": "ok",  
  "message": "processed successfully"  
}
```

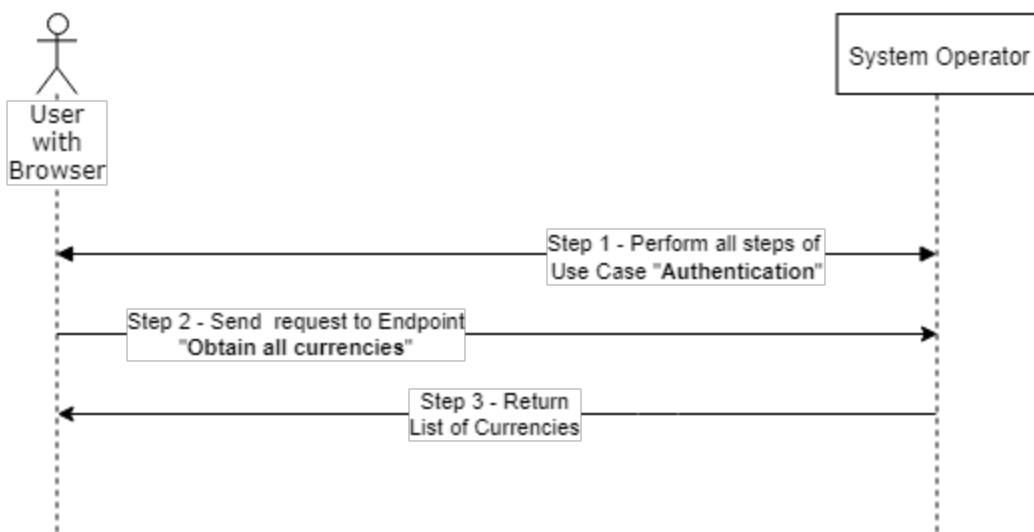
Obtain all currencies scheme

Use case: Obtain all currencies

Basic FLow



Optional Web UI Flow



Organization membership management

Create an organization member description

Use Case Name

Create an organization member

Brief Description

A User, most likely CFO, or External Entity on behalf of a User with role permissions ORGANIZATION_VIEWER, USER_VIEWER, USER_MANAGER will go through all steps of “View organizations” and “Get users” Use Cases. These Use Cases will provide all parameters needed to send a request to Endpoint “Create an organization member”. Any user of a System must belong to at least one organization. When this Use Case is successfully executed, a User becomes a member of one more organization.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ORGANIZATION_VIEWER, USER_VIEWER, USER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.
2. The target organization must be compatible with the user role.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Cases “View organizations” and “Get users”.
2. External Entity sends a request to Endpoint “Create an organization member”.

Endpoint URL: POST /members

Parameters:

```
{
  "userId": "string",
  "organizationId": "string",
  "role": "string",
  "active": false
}
```

“userId”: “string”, - retrieved in Use Case “Get users”

“organizationId”: “string”, - retrieved in Use Case “View organizations”

3. System Operator returns organization information that a User just joined to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Cases “View organizations” and “Get users”.
2. A user sends a request to Endpoint “Create an organization member”.

Endpoint URL: POST /members

Parameters:

```
{
  "userId": "string",
  "organizationId": "string",
  "role": "string",
  "active": false
}
```

"userId": "string", - retrieved in Use Case “Get users”

"organizationId": "string", - retrieved in Use Case “View organizations”

3. System Operator returns organization information that a User just joined to User (See Result example below).

Post Conditions

A user becomes a member of an additional organization.

Result example

```
{
  "member": {
    "id": "string",
    "role": "string",
    "user": {
      "id": "string",
      "name": "string"
    },
    "organization": {
      "organizationId": "string",
      "type": "string",
      "name": "string"
    }
  },
  "status": "ok",
  "message": "string"
}
```

"organizationId": "string", - a User with userId became a member of one more organization.

Example:

After executing of this operation a user with id “b733ab3b-564e-4bef-bfc9-2ab7b490d90e” becomes a member of two organizations:

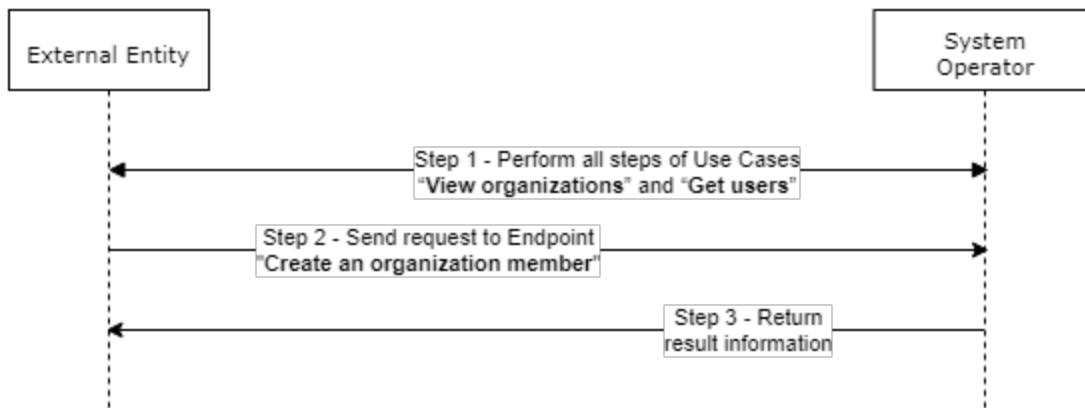
```
{
  "userId": "b733ab3b-564e-4bef-bfc9-2ab7b490d90e",
  "name": "",
  "createdAt": "2018-08-22T14:16:00.414Z",
  "active": true,
  "banned": false,
  "banExpiryDate": null,
  "contact": {
    "phoneNumber": null,
    "phoneVerified": false,
    "email": "teststtst@mailinator.com",
    "emailVerified": true,
    "countryCode": null
  }
}
```

```
},
"members": [
{
  "id": "79851ccb-9f2e-4372-bfb2-aa10514d8a10",
  "role": "cashier",
  "organization": {
    "id": "9fbf407e-a91d-461c-87a8-53699b2be71e",
    "type": "cash_desk",
    "name": ""
  }
},
{
  "id": "588b7670-86aa-4f39-bde2-8ce003c44545",
  "role": "cashier",
  "organization": {
    "id": "2b1b5263-539b-4901-aff0-e69b9750d5da",
    "type": "cash_desk",
    "name": "cash_desk"
  }
}
]
```

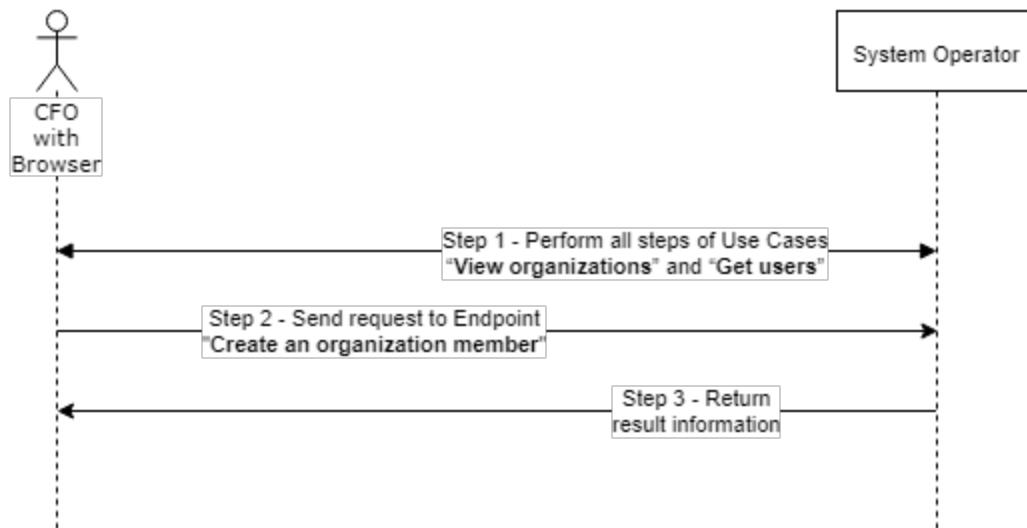
Create an organization member scheme

Use case: Create an organization member

Basic FFlow



Optional Web UI Flow



Delete organization member description

Use Case Name

Delete organization member

Brief Description

A User or External Entity on behalf of a User with role permission USER_MANAGER will go through all steps of “Get users” Use Case, and then send a request to Endpoint “Delete organization member”.

The Use Case “Get users” will provide the following pieces of information:

- a User ID in the system
- for each User a list of Organization with which User is associated.
- a member ID for that User in every Organization.

The system allows a User to belong to one or more Organizations. If this Use Case executed on the User who was a member of only one organization, the User will be deleted.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.
2. The user must be a member of any organization.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Delete organization member”.

Endpoint URL: DELETE /members/{memberId} - not user Id.

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Delete organization member”.

Endpoint URL: DELETE /members/{memberId} - not user Id.

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

A User is removed from the organization.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Data example

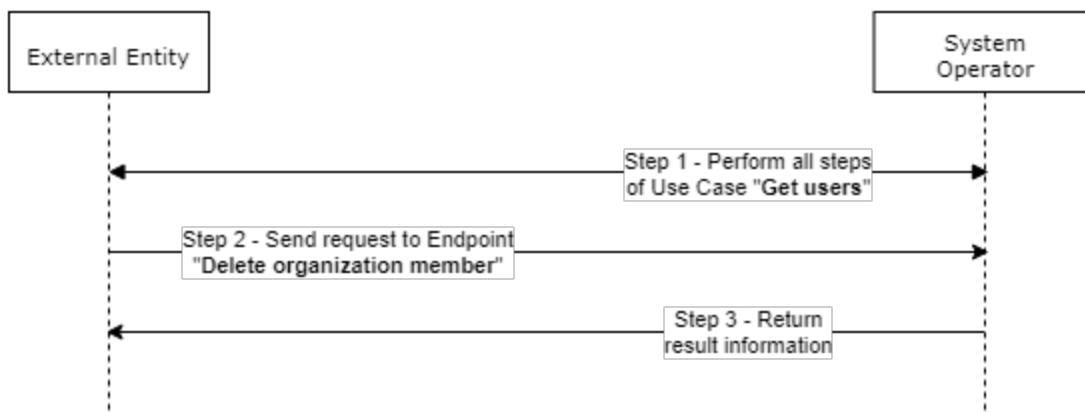
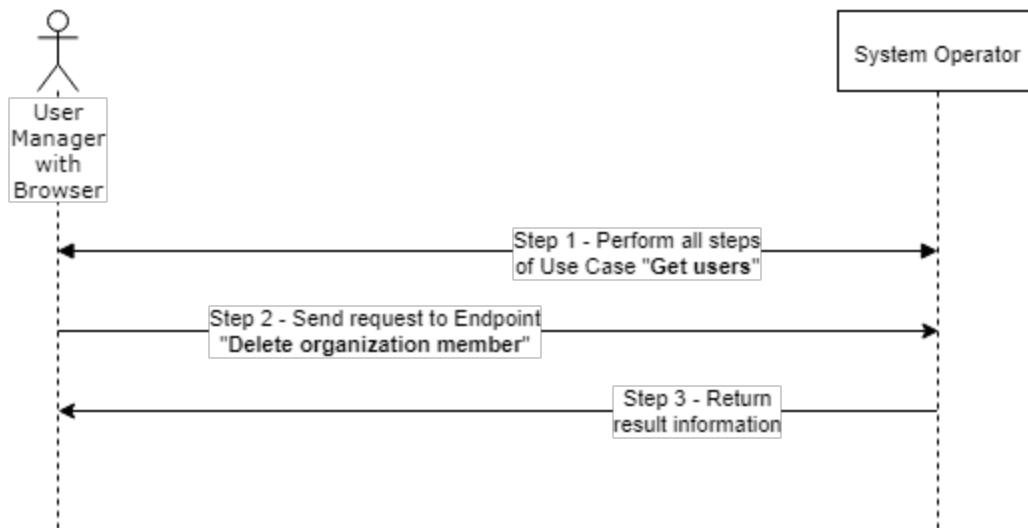
```
{  
  "userID": "1",  
  "name": "User1",  
  "members": [  
    {  
      "memberID": "2",  
      "organization": {"id": "3", "name": "Org3"}  
    },  
    {  
      "memberID": "4",  
      "organization": {"id": "5", "name": "Org5"}  
    }  
  ]  
}
```

In the structure above a User1 is a member of two organizations:

-- Org3 with member ID 2
-- Org5 with member ID 4

If you delete both members, you delete a User.

Delete organization member scheme

Use case: Delete organization member**Basic FLow****Optional Web UI Flow**

Update organization member description

Use Case Name

Update organization member

Brief Description

A User or External Entity on behalf of a User with role permission USER_MANAGER will go through all steps of “Get users” Use Case obtain an organization member ID for the user in question, and then send a request to Endpoint “Update organization member”. The only property that gets updated is the ACTIVE flag. Possible values are “true” and “false”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Update organization member”.

Endpoint URL: PATCH /members/{id} - organization member Id, not a user Id.

Parameters:

```
{
  "active":false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Update organization member”.

Endpoint URL: PATCH /members/{id} - organization member Id, not a user Id.

Parameters:

```
{
  "active":false
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

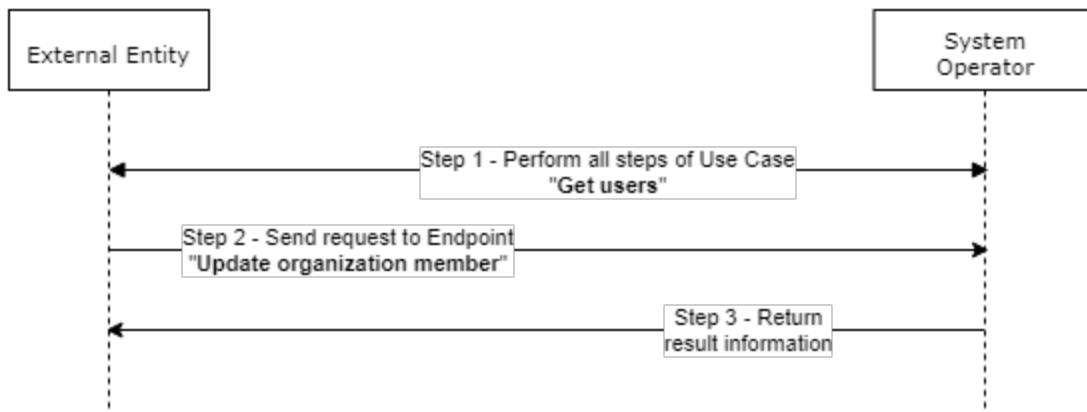
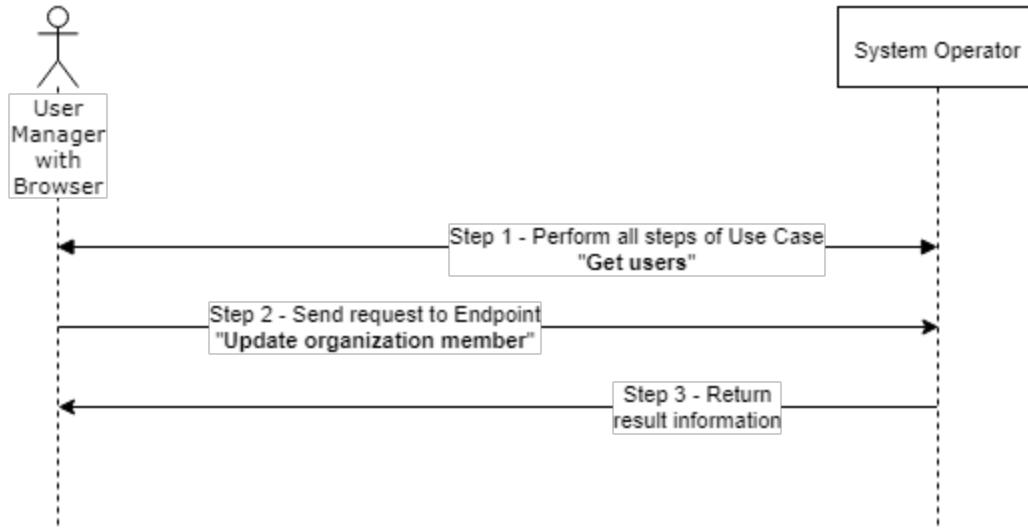
Member information is updated per user request.

Result example

```
{  
  "member": {  
    "id": "string",  
    "role": "string",  
    "user": {  
      "id": "string",  
      "name": "string"  
    },  
    "organization": {  
      "id": "string",  
      "type": "string",  
      "name": "string"  
    }  
  },  
  "status": "ok",  
  "message": "string"  
}
```

"id": "string", - Member's identifier

Update organization member scheme

Use case: Update organization member**Basic FLow****Optional Web UI Flow**

Organization management

View organizations description

Use Case Name

View organizations

Brief Description

A User or External Entity on behalf of a User with role permission ORGANIZATION_VIEWER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View organizations”. When there are no filtering parameter available, the filter could be omitted to list all organizations in the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ORGANIZATION_VIEWER, e.g. organization, accountant or CFO.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View organizations”.

Endpoint URL: POST /organizations

Parameters: {

“filter”: { - filter is optional

“types”: [

“string”

],

“emitterId”: “string”,

“organizationTypes”: [

{

“business”: false,

“coinType”: “regular_commission”,

“registerable”: false,

“issuableCoinTypes”: [

“regular_commission”

],

“autoDeletable”: false,

```

    "bundleKey": "string",
    "code": "string"
  }
],
},
"sort": {
  "creationDate": "asc"
},
"pageNumber": 0,
"pageSize": 0
}

```

1. System Operator returns list of organizations to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View organizations”.

Endpoint URL: POST /organizations

```

Parameters: {
  "filter": { - filter is optional
  "types": [
    "string"
  ],
  "emitterId": "string",
  "organizationTypes": [
    {
      "business": false,
      "coinType": "regular_commission",
      "registerable": false,
      "issuableCoinTypes": [
        "regular_commission"
      ],
      "autoDeletable": false,
      "bundleKey": "string",
      "code": "string"
    }
  ]
},
"sort": {
  "creationDate": "asc"
},

```

```
"pageNumber": 0,  
"pageSize": 0  
}
```

1. System Operator returns list of organizations to User (See Result example below).

Post Conditions

List of organizations is available. Token is available.

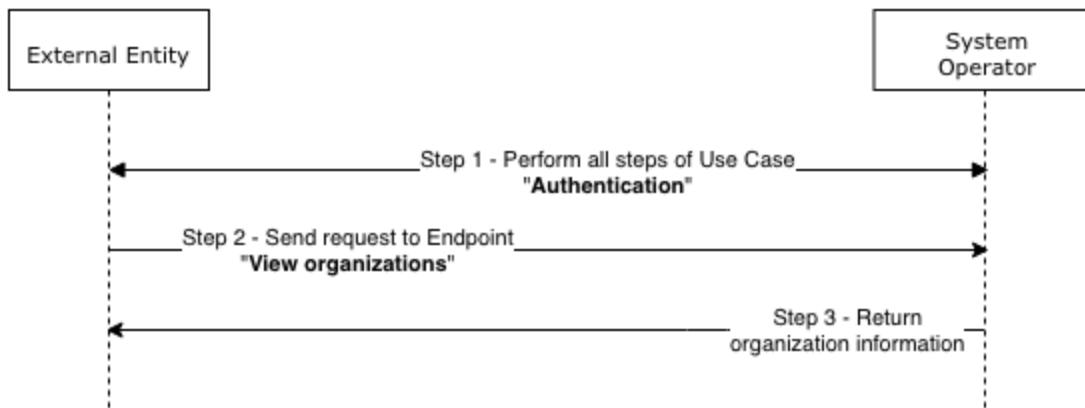
Result example

```
{  
"pageNumber": 0,  
"pageSize": 0,  
"totalRecords": 0,  
"totalPages": 0,  
"status": "ok",  
"message": "string",  
"records": [  
{}  
]  
}
```

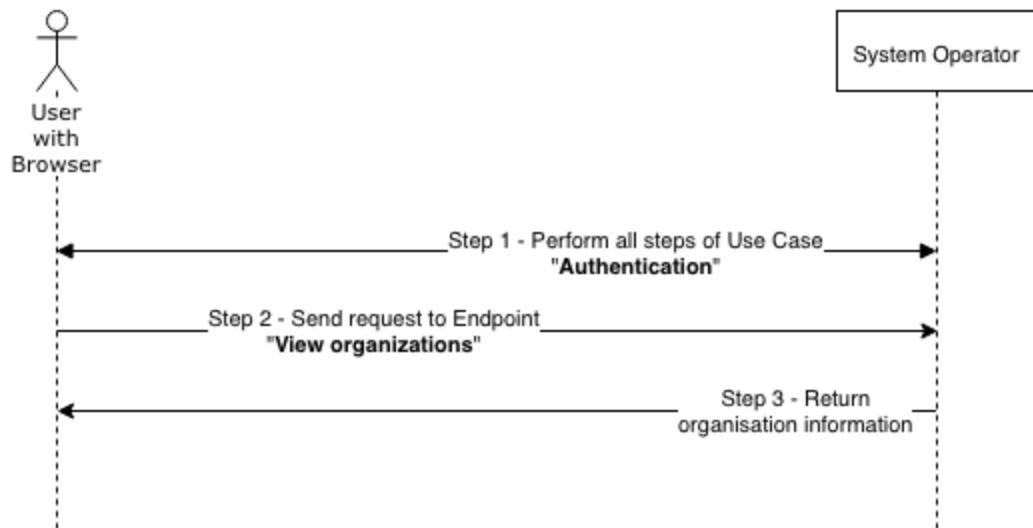
View organizations scheme

Use case: Organization management

Basic FFlow



Optional Web UI Flow



Contract management

Create a new commission profile description

Use Case Name

Create a new commission profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Obtain commission profiles for specified contract” Use Case, and then send a request to Endpoint “Create a new commission profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain commission profiles for the specified contract”.
2. External Entity sends a request to Endpoint “Create a new commission profile”.

Endpoint URL: POST /contracts/{contractId}/commission-profiles

Parameters:

```
{
  "issuerId": "string",
  "operationFlowId": "string",
  "srcParticipantSpecification": {
    "type": "no",
    "value": "string"
  },
  "destParticipantSpecification": {
    "type": "no",
    "value": "string"
  },
  "direction": "IN",
  "value": {
    "type": "zero",
    "valuePercent": 0,
    "valueFixed": 0
  }
}
```

3. System Operator returns new profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain commission profiles for the specified contract”.
2. A user sends a request to Endpoint “Create a new commission profile”.

Endpoint URL: POST /contracts/{contractId}/commission-profiles

Parameters:

```
{  
    "issuerId": "string",  
    "operationFlowId": "string",  
    "srcParticipantSpecification": {  
        "type": "no",  
        "value": "string"  
    },  
    "destParticipantSpecification": {  
        "type": "no",  
        "value": "string"  
    },  
    "direction": "IN",  
    "value": {  
        "type": "zero",  
        "valuePercent": 0,  
        "valueFixed": 0  
    }  
}
```

3. System Operator returns new profile information information to User (See Result example below).

Post Conditions

New Commission profile is available.

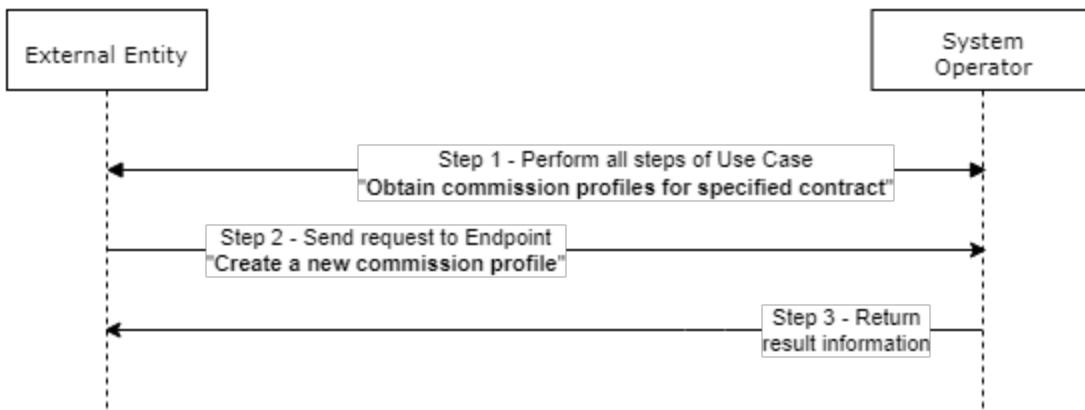
Result example

```
{  
    "profile": {  
        "id": "string",  
        "createdAt": "2018-08-10T14:07:38.636Z",  
        "updatedAt": "2018-08-10T14:07:38.636Z",  
        "flow": {  
            "id": "string",  
            "code": "string",  
            "transactionType": "transfer",  
            "processType": "string",  
            "srcCoinType": "regular_commission",  
            "destCoinType": "regular_commission"  
        },  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "srcParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "destParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "active": false,  
        "direction": "IN",  
        "value": {  
            "type": "zero",  
            "valuePercent": 0,  
            "valueFixed": 0  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

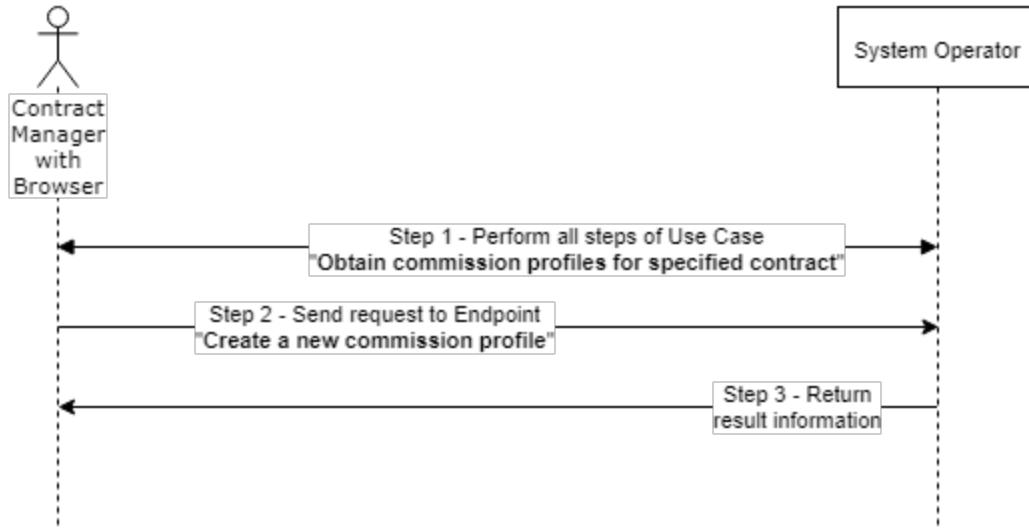
Create a new commission profile scheme

Use case: Create a new commission profile

Basic FFlow



Optional Web UI Flow



Create a new limit profile description

Use Case Name

Create a new limit profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Obtain limit profiles for specified commission profile” Use Case and then send a request to Endpoint “Create a new limit profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain limit profiles for specified commission profile”.
2. External Entity sends a request to Endpoint “Create a new limit profile”.

Endpoint URL: POST /contracts/{contractId}/commission-profiles/{profileId}/limit-profiles

Parameters:

```
{
  "qualifier": "amount",
  "timeUnit": "per_transaction",
  "value": 0,
  "active": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain limit profiles for specified commission profile”.
2. A user sends a request to Endpoint “Create a new limit profile”.

Endpoint URL: POST /contracts/{contractId}/commission-profiles/{profileId}/limit-profiles

Parameters:

```
{
  "qualifier": "amount",
  "timeUnit": "per_transaction",
  "value": 0,
  "active": false
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

New Limit Profile is available.

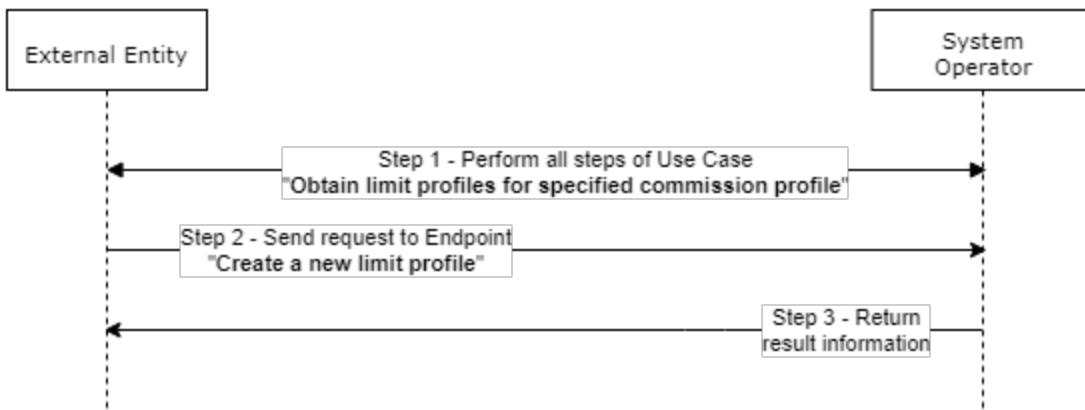
Result example

```
{  
  "profile": {  
    "id": "string",  
    "commissionProfileId": "string",  
    "qualifier": "amount",  
    "timeUnit": "per_transaction",  
    "value": 0,  
    "active": false  
  },  
  "status": "ok",  
  "message": "string"  
}
```

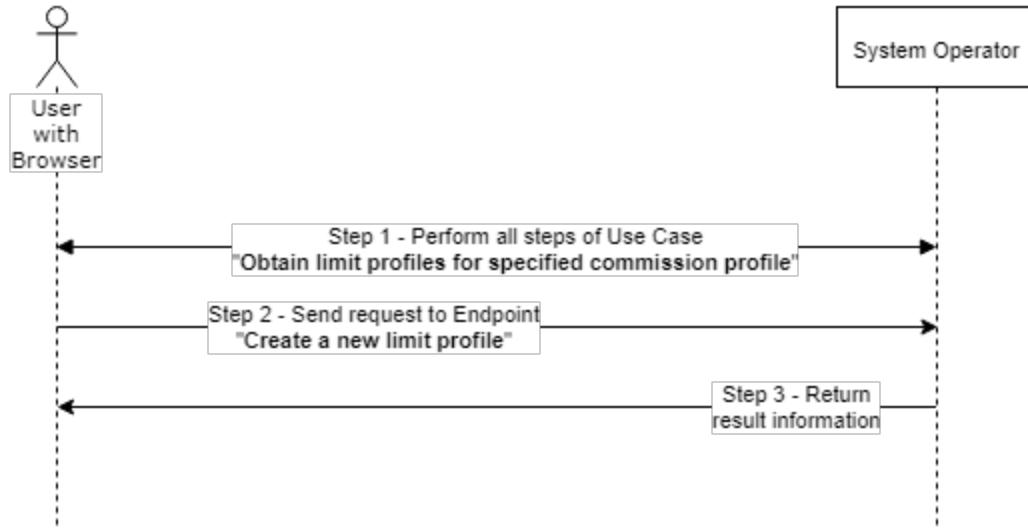
Create a new limit profile scheme

Use case: Create a new limit profile

Basic FFlow



Optional Web UI Flow



Obtain all available operation flows description

Use Case Name

Obtain all available operation flows

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Obtain all available operation flows”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Obtain all available operation flows”.

Endpoint URL: GET /operation-flows

Parameters: TOKEN

3. System Operator returns List of Flows to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Obtain all available operation flows”.

Endpoint URL: GET /operation-flows

Parameters: TOKEN

3. System Operator returns List of Flows to User (See Result example below).

Post Conditions

List of Flows is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "code": "string",  
            "transactionType": "transfer",  
            "processType": "string",  
            "srcCoinType": "regular_commission",  
            "destCoinType": "regular_commission"  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Obtain all available operation flows scheme

Use case: Obtain all available operation flows



External Entity

System Operator

Step 1 - Perform all steps of Use Case
"Authentication"Step 2 - Send request to Endpoint
"Obtain all available operation flows"Step 3 - Return
List of flows

Contract Manager with Browser

System Operator

Step 1 - Perform all steps of Use Case
"Authentication"Step 2 - Send request to Endpoint
"Obtain all available operation flows"Step 3 - Return
List of flows

Obtain commission profiles for specified contract description

Use Case Name

Obtain commission profiles for the specified contract

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Obtain commission profiles for the specified contract”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain contracts matching the specified criteria”.
2. External Entity sends a request to Endpoint “Obtain commission profiles for the specified contract”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles

Parameters: TOKEN

3. System Operator returns List of profiles to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain contracts matching the specified criteria”.
2. A user sends a request to Endpoint “Obtain commission profiles for the specified contract”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles

Parameters: TOKEN

3. System Operator returns List of profiles to User (See Result example below).

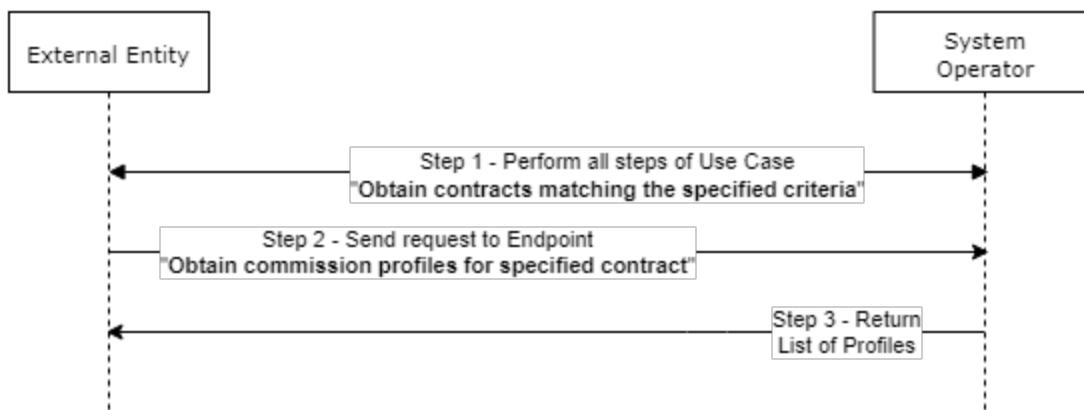
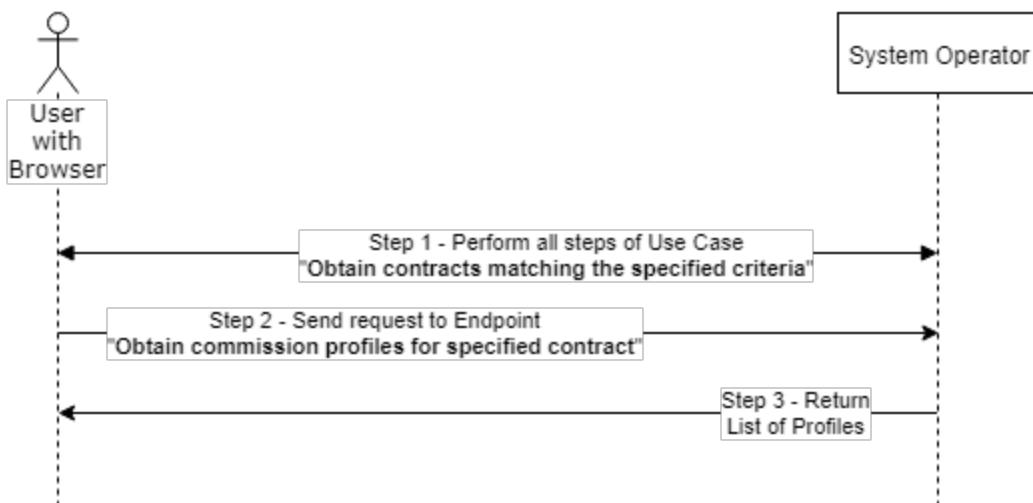
Post Conditions

List of profiles is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "createdAt": "2018-08-10T14:07:38.628Z",  
            "updatedAt": "2018-08-10T14:07:38.629Z",  
            "flow": {  
                "id": "string",  
                "code": "string",  
                "transactionType": "transfer",  
                "processType": "string",  
                "srcCoinType": "regular_commission",  
                "destCoinType": "regular_commission"  
            },  
            "issuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            },  
            "srcParticipantSpecification": {  
                "type": "no",  
                "value": "string"  
            },  
            "destParticipantSpecification": {  
                "type": "no",  
                "value": "string"  
            },  
            "active": false,  
            "direction": "IN",  
            "value": {  
                "type": "zero",  
                "valuePercent": 0,  
                "valueFixed": 0  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Obtain commission profiles for specified contract scheme

Use case: Obtain commission profiles for specified contract**Basic FFlow****Optional Web UI Flow**

Obtain contracts matching the specified criteria description

Use Case Name

Obtain contracts matching the specified criteria

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Obtain contracts matching the specified criteria”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Obtain contracts matching the specified criteria”.

Endpoint URL: POST /contracts/view

Parameters:

```
{
  "filter": {
    "id": "string",
    "types": [
      "string"
    ],
    "global": false,
    "personTypes": [
      "base"
    ],
    "dateFrom": "2018-08-10T14:07:37.801Z",
    "dateTo": "2018-08-10T14:07:37.801Z"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 20
}
```

3. System Operator returns List of Contracts to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “Obtain contracts matching the specified criteria”.

Endpoint URL: POST /contracts/view

Parameters:

```
{  
    "filter": {  
        "id": "string",  
        "types": [  
            "string"  
        ],  
        "global": false,  
        "personTypes": [  
            "base"  
        ],  
        "dateFrom": "2018-08-10T14:07:37.801Z",  
        "dateTo": "2018-08-10T14:07:37.801Z"  
    },  
    "sort": {  
        "date": "asc"  
    },  
    "pageNumber": 0,  
    "pageSize": 20  
}
```

3. System Operator returns List of Contracts to User (See Result example below).

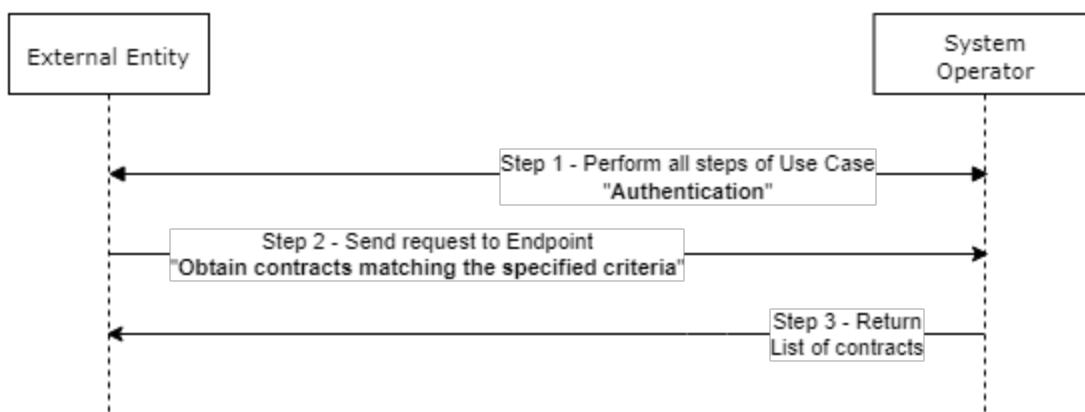
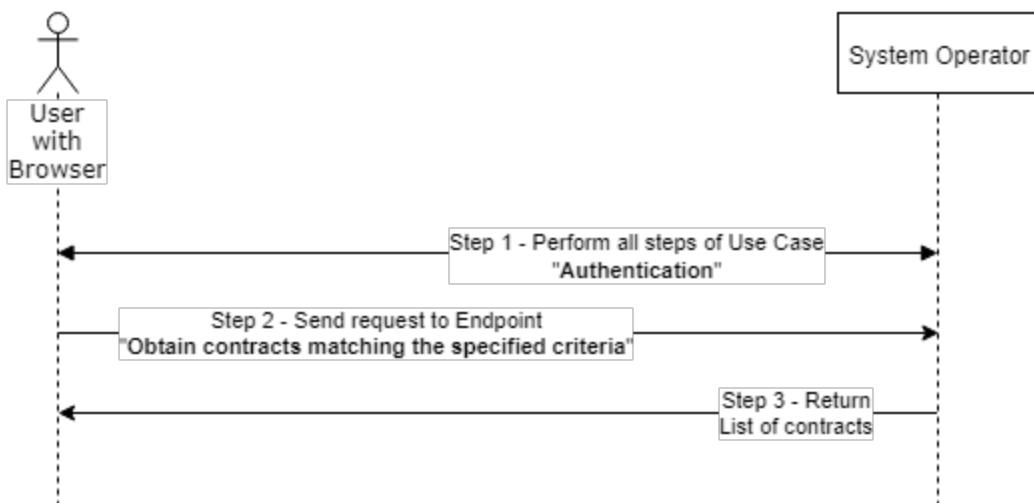
Post Conditions

Contract list is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "organizationType": "string",  
            "personType": "base",  
            "name": "string",  
            "description": "string",  
            "active": false,  
            "global": false  
        }  
    ],  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0  
}
```

Obtain contracts matching the specified criteria scheme

Use case: Obtain contracts matching the specified criteria**Basic FFlow****Optional Web UI Flow**

Obtain limit profiles for specified commission profile description

Use Case Name

Obtain limit profiles for specified commission profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “View commission profile” Use Case, and then send a request to Endpoint “Obtain limit profiles for specified commission profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View commission profile”.
2. External Entity sends a request to Endpoint “Obtain limit profiles for a specified commission profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{profileId}/limit-profiles

Parameters: TOKEN

3. System Operator returns List of Limit Profiles to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View commission profile”.
2. A user sends a request to Endpoint “Obtain limit profiles for specified commission profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{profileId}/limit-profiles

Parameters: TOKEN

3. System Operator returns List of Limit Profiles to User (See Result example below).

Post Conditions

List of Limit Profiles is available.

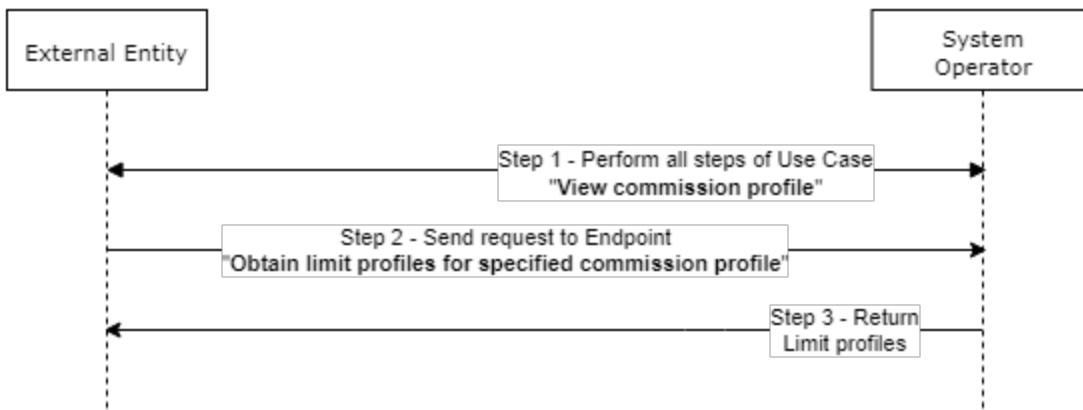
Result example

```
{  
    "limit_profiles": [  
        {  
            "id": "string",  
            "commissionProfileId": "string",  
            "qualifier": "amount",  
            "timeUnit": "per_transaction",  
            "value": 0,  
            "active": false  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

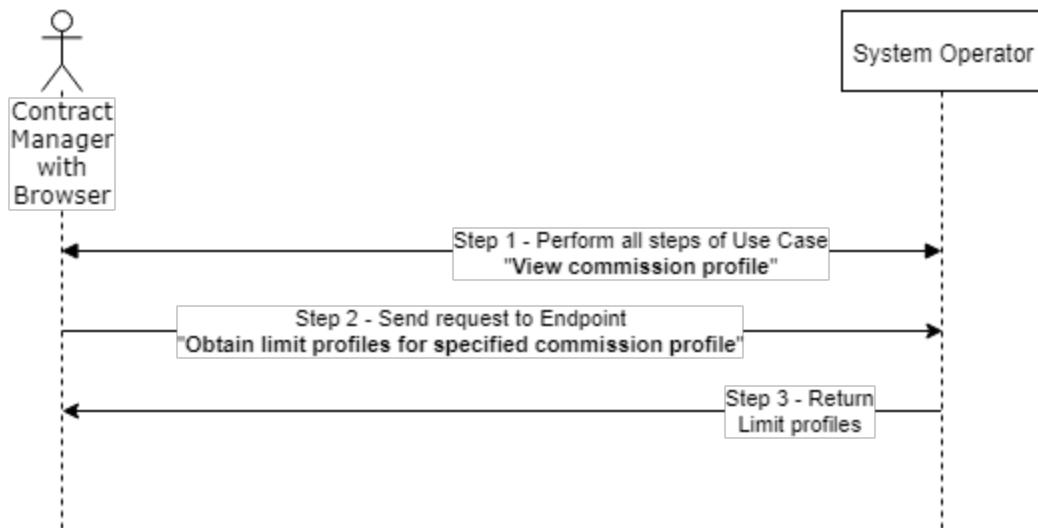
Obtain limit profiles for specified commission profile scheme

Use case: Obtain limit profiles for specified commission profile

Basic FLow



Optional Web UI Flow



Update an existing commission profile description

Use Case Name

Update an existing commission profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “View commission profile” Use Case, and then send a request to Endpoint “Update an existing commission profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View commission profile”.
2. External Entity sends a request to Endpoint “Update an existing commission profile”.

Endpoint URL: PATCH /contracts/{contractId}/commission-profiles/{profileId}

Parameters:

```
{
  "direction": "IN",
  "value": {
    "type": "zero",
    "valuePercent": 0,
    "valueFixed": 0
  },
  "active": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View commission profile”.
2. A user sends a request to Endpoint “Update an existing commission profile”.

Endpoint URL: PATCH /contracts/{contractId}/commission-profiles/{profileId}

Parameters:

```
{  
    "direction": "IN",  
    "value": {  
        "type": "zero",  
        "valuePercent": 0,  
        "valueFixed": 0  
    },  
    "active": false  
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

The updated profile is available.

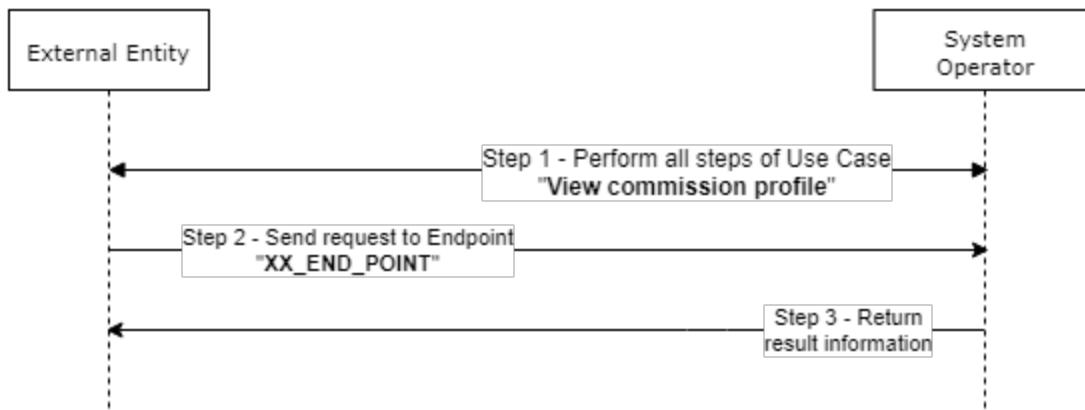
Result example

```
{  
    "profile": {  
        "id": "string",  
        "createdAt": "2018-08-10T14:07:38.604Z",  
        "updatedAt": "2018-08-10T14:07:38.604Z",  
        "flow": {  
            "id": "string",  
            "code": "string",  
            "transactionType": "transfer",  
            "processType": "string",  
            "srcCoinType": "regular_commission",  
            "destCoinType": "regular_commission"  
        },  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "srcParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "destParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "active": false,  
        "direction": "IN",  
        "value": {  
            "type": "zero",  
            "valuePercent": 0,  
            "valueFixed": 0  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

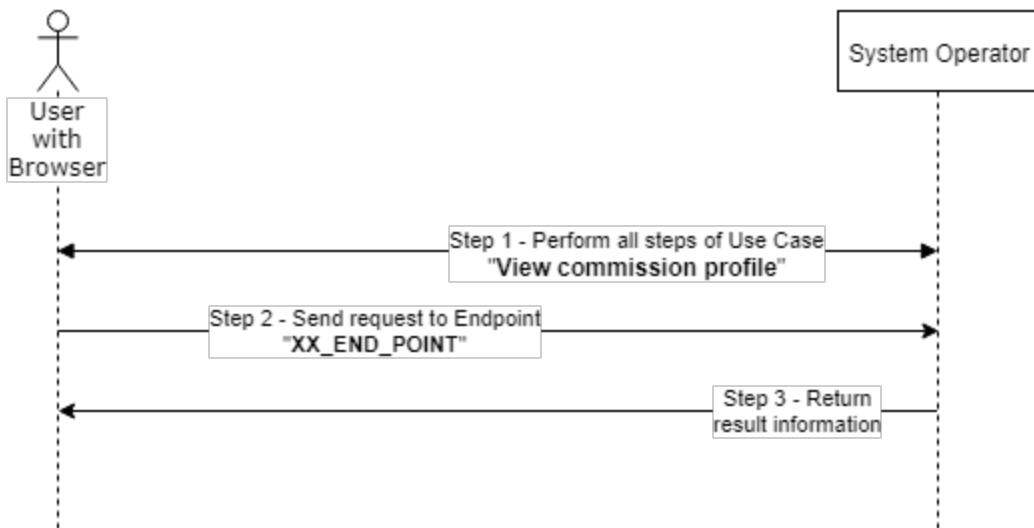
Update an existing commission profile scheme

Use case: XX_TEMPLATE

Basic FFlow



Optional Web UI Flow



Update an existing limit profile description

Use Case Name

Update an existing limit profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “View limit profile” Use Case, and then send a request to Endpoint “Update an existing limit profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View limit profile”.
2. External Entity sends a request to Endpoint “Update an existing limit profile”.

Endpoint URL:

PATCH /contracts/{contractId}/commission-profiles/{commissionProfileId}/limit-profiles/{profileId}

Parameters:

```
{
  "value": 0,
  "active": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View limit profile”.
2. A user sends a request to Endpoint “Update an existing limit profile”.

Endpoint URL:

PATCH /contracts/{contractId}/commission-profiles/{commissionProfileId}/limit-profiles/{profileId}

Parameters:

```
{
  "value": 0,
  "active": false
}
```

3. System Operator returns result information to User (See Result example below).

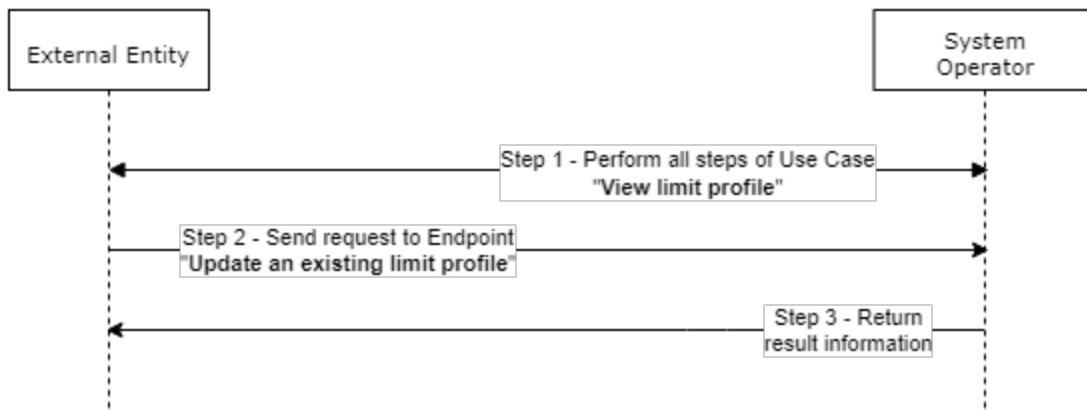
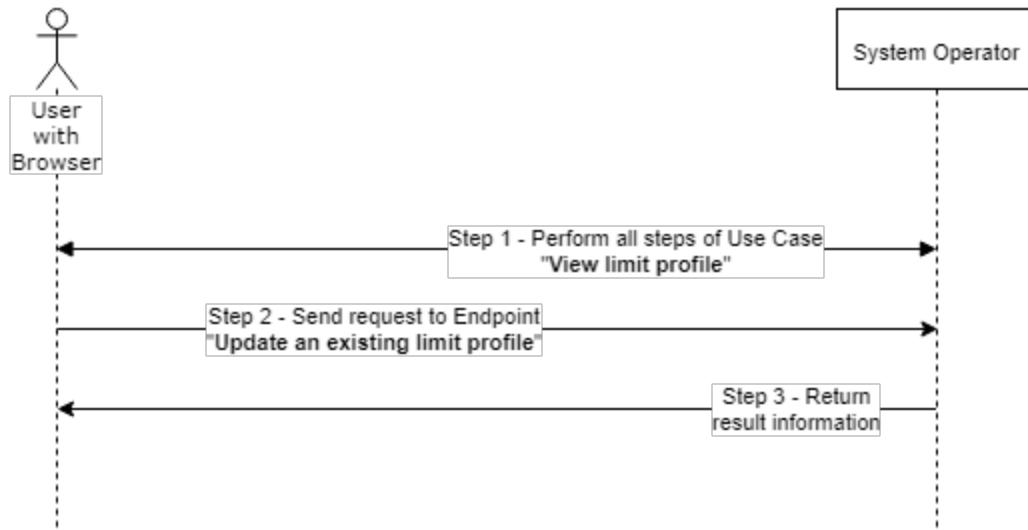
Post Conditions

The updated limit is available if active.

Result example

```
{  
    "profile": {  
        "id": "string",  
        "commissionProfileId": "string",  
        "qualifier": "amount",  
        "timeUnit": "per_transaction",  
        "value": 0,  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Update an existing limit profile scheme

Use case: Update an existing limit profile**Basic FFlow****Optional Web UI Flow**

View commission profile description

Use Case Name

View commission profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Obtain commission profiles for the specified contract” Use Case and then send a request to Endpoint “View commission profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain commission profiles for the specified contract”.
2. External Entity sends a request to Endpoint “View commission profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{profileId}

Parameters: TOKEN

3. System Operator returns profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain commission profiles for the specified contract”.
2. A user sends a request to Endpoint “View commission profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{profileId}

Parameters: TOKEN

3. System Operator returns profile information to User (See Result example below).

Post Conditions

Profile information is available.

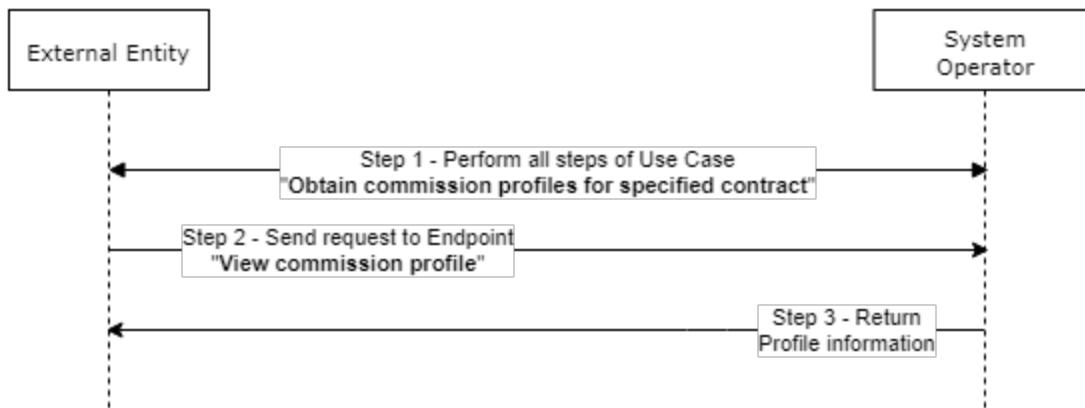
Result example

```
{  
    "profile": {  
        "id": "string",  
        "createdAt": "2018-08-10T14:07:38.595Z",  
        "updatedAt": "2018-08-10T14:07:38.595Z",  
        "flow": {  
            "id": "string",  
            "code": "string",  
            "transactionType": "transfer",  
            "processType": "string",  
            "srcCoinType": "regular_commission",  
            "destCoinType": "regular_commission"  
        },  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "srcParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "destParticipantSpecification": {  
            "type": "no",  
            "value": "string"  
        },  
        "active": false,  
        "direction": "IN",  
        "value": {  
            "type": "zero",  
            "valuePercent": 0,  
            "valueFixed": 0  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

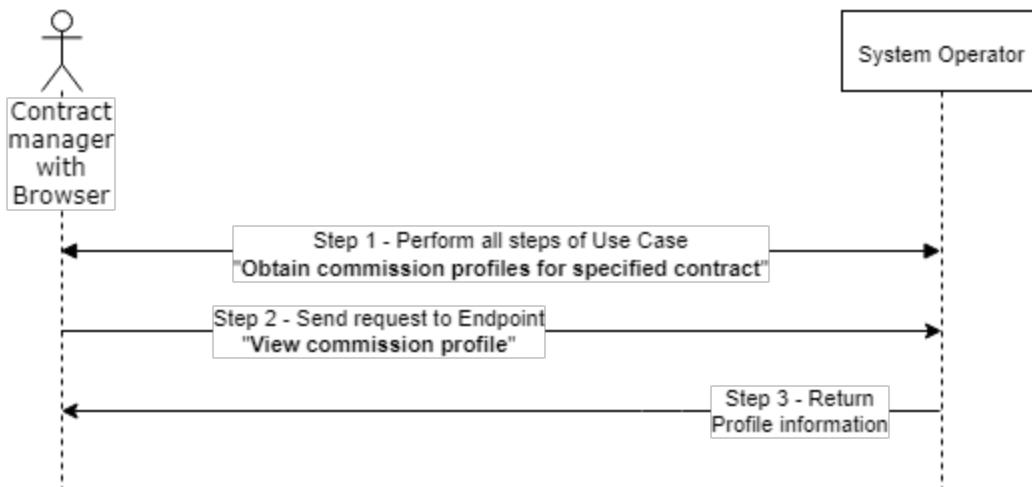
View commission profile scheme

Use case: View commission profile

Basic FFlow



Optional Web UI Flow



View limit profile description

Use Case Name

View limit profile

Brief Description

A User or External Entity on behalf of a User with role permission CONTRACT_MANAGER will go through all steps of “Obtain limit profiles for specified commission profile” Use Case and then send a request to Endpoint “View limit profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONTRACT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain limit profiles for specified commission profile”.
2. External Entity sends a request to Endpoint “View limit profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{commissionProfileId}/limit-profiles/{profileId}

Parameters: TOKEN

3. System Operator returns Limit profile details to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain limit profiles for specified commission profile”.
2. A user sends a request to Endpoint “View limit profile”.

Endpoint URL: GET /contracts/{contractId}/commission-profiles/{commissionProfileId}/limit-profiles/{profileId}

Parameters: TOKEN

3. System Operator returns Limit profile details to User (See Result example below).

Post Conditions

Limit profile details are available.

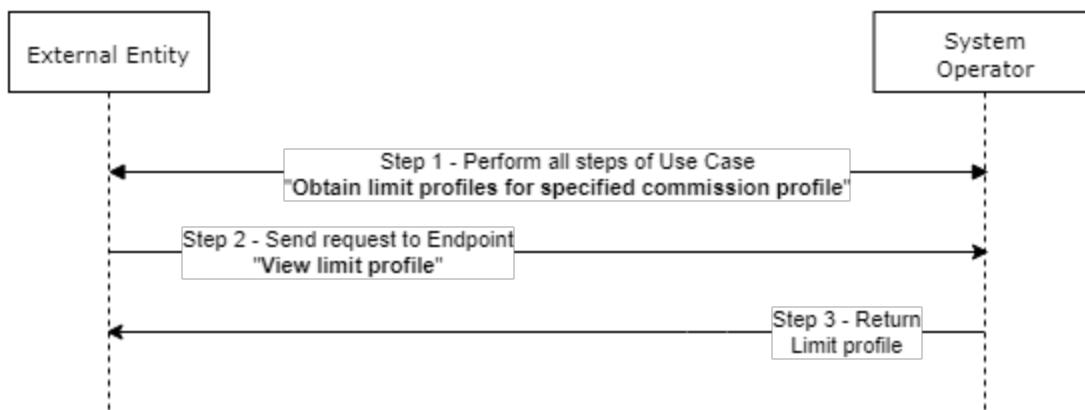
Result example

```
{
  "profile": {
    "id": "string",
    "commissionProfileId": "string",
    "qualifier": "amount",
    "timeUnit": "per_transaction",
    "value": 0,
    "active": false
  },
  "status": "ok",
  "message": "string"
}
```

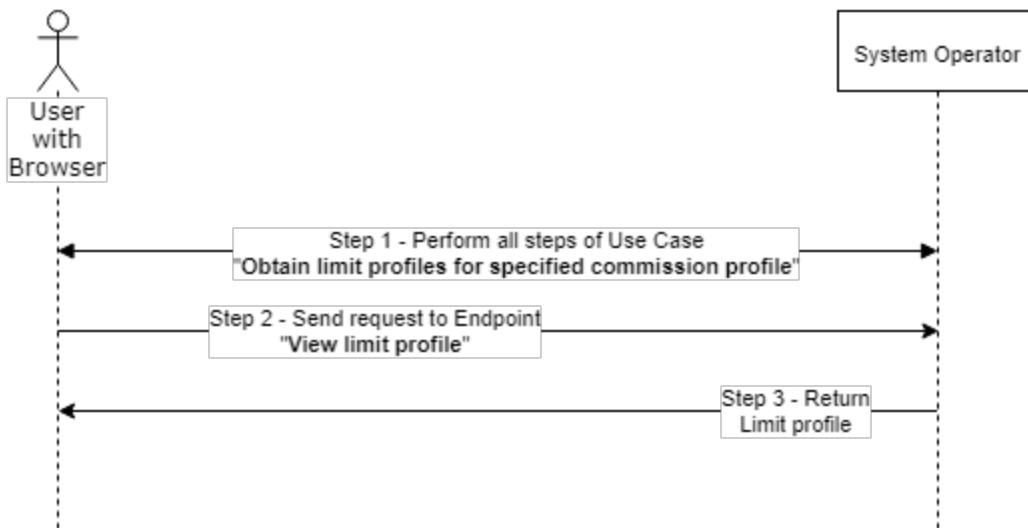
View limit profile scheme

Use case: View limit profile

Basic FFlow



Optional Web UI Flow



Tariffs for changing contract\$

_Pay for transition description

Use Case Name

xx

Brief Description

xx

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to get a list of all user coins.

Endpoint URL: GET /issuers

Parameter: Security TOKEN.

3. System Operator return a List of Issuers to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to get a list of all Issuers.

Endpoint URL: GET /issuers

Parameter: Security TOKEN.

3. System Operator return a web page with a List of Issuers to User.

Post Conditions

xx

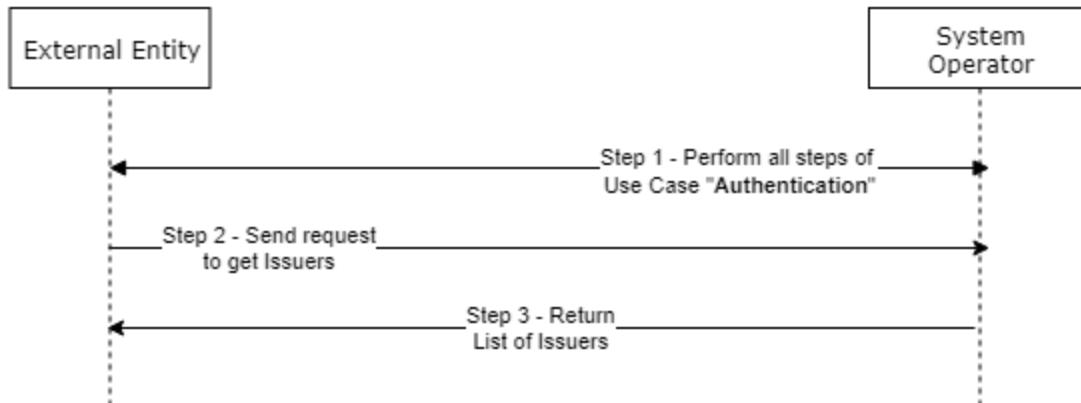
Result example

xx

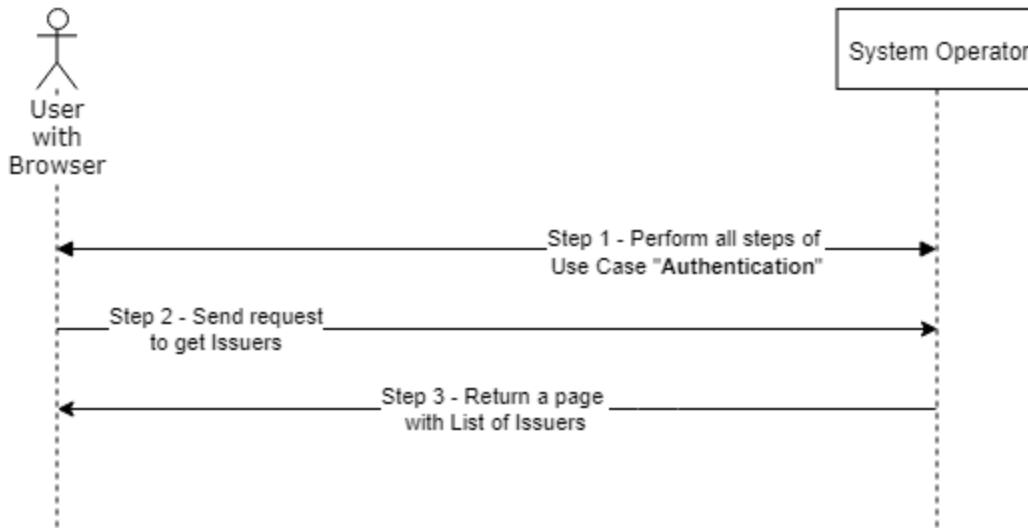
_Pay for transition scheme

Basic Flow

Use case: TEMPLATE



Optional Web UI Flow



Get available contracts description

Use Case Name

Get available contracts

Brief Description

xx

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
 2. External Entity sends a request to get a list of all user coins.
- Endpoint URL: GET /contracts/tariffs
- Parameter: Security TOKEN.
3. System Operator returns a List of Contracts.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
 2. A user sends a request to get a List of Contracts.
- Endpoint URL: GET /contracts/tariffs
- Parameter: Security TOKEN.
3. System Operator returns a List of Contracts.

Post Conditions

Contract list is available.

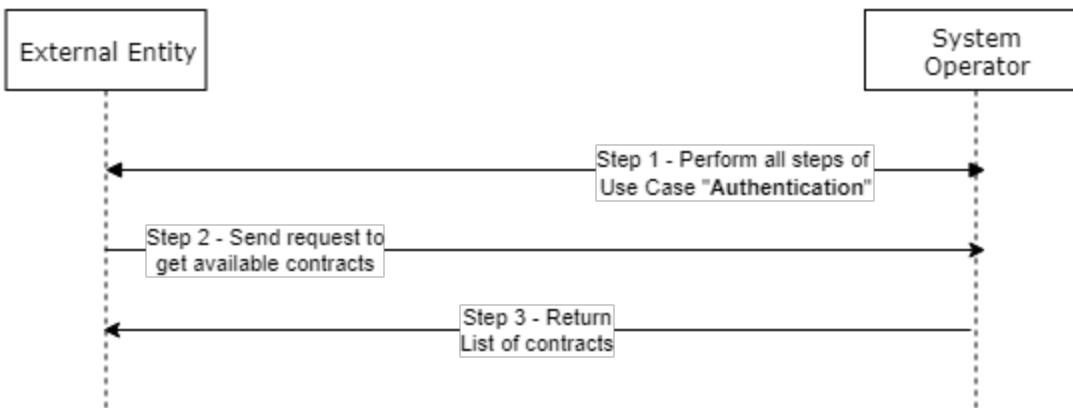
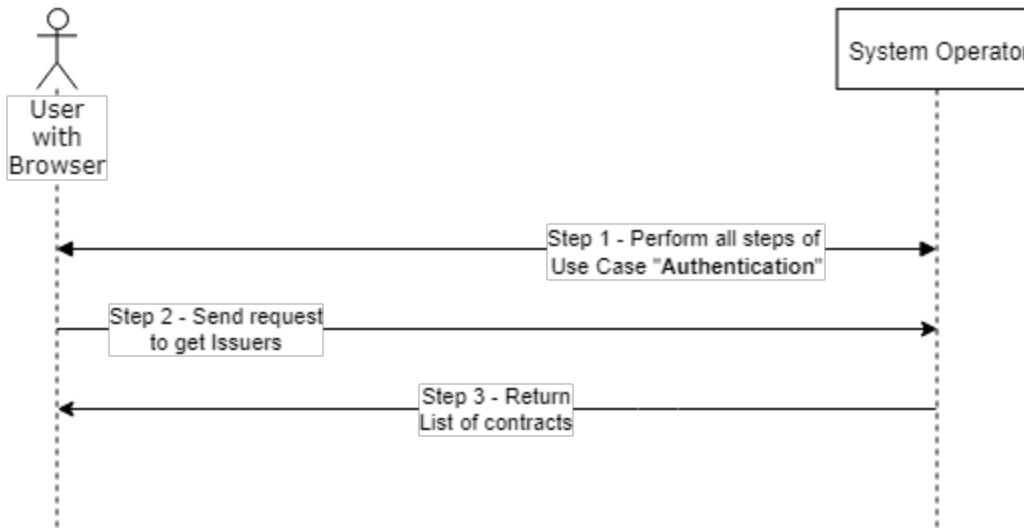
Result example

```
{
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "records": [
    {
      "id": "string",
      "createdAt": "2018-07-23T09:40:15.198Z",
      "updatedAt": "2018-07-23T09:40:15.198Z",
      "type": "string",
      "status": "limited",
      "requestIdentifier": 0,
      "requestStatus": "limited",
      "transactions": [
        {
          "id": "string",
          "parentId": "string",
          "type": "transfer",
          "from": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "to": {
              "serial": "string",
              "organizationId": "string",
              "organizationName": "string"
            }
          }
        }
      ]
    }
  ]
}
```

```
        "technical":false,
        "type":"regular_commission",
        "issuer":{
            "id":"string",
            "sn":"string",
            "currency":"string"
        }
    },
    "to":{
        "serial":"string",
        "organizationId":"string",
        "organizationName":"string",
        "technical":false,
        "type":"regular_commission",
        "issuer":{
            "id":"string",
            "sn":"string",
            "currency":"string"
        }
    },
    "amount":0,
    "performedAt":"2018-07-23T09:40:15.198Z",
    "issuer:{
        "id":"string",
        "sn":"string",
        "currency":"string"
    }
}
],
"children":[
{
}
],
"errorMessage":"string"
```

```
        }  
    ]  
}
```

Get available contracts scheme

Use case: Get available contracts**Basic FLow****Optional Web UI Flow**

[View dashboard description](#)

Use Case Name

[View dashboard](#)

Brief Description

An authenticated user obtains a list of Issuers available at System Operator.

Then a request is sent to get a Dashboard for selected Issuers.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain issuers”.

2. System Operator returns a list of all Issuers.

3. External Entity selects Issuer and sends a request to get a Dash.

Endpoint URL: GET /contracts/tariffs/dashboard/{issuerId}

Parameter: Security TOKEN.

4. System Operator return a List of Issuers to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.

2. A user sends a request to get a list of all Issuers.

Endpoint URL: GET /contracts/tariffs/dashboard/{issuerId}

Parameter: Security TOKEN.

3. System Operator return a web page with a List of Issuers to User.

Post Conditions

The dashboard is available.

Result example

```
{
  "status": "ok",
  "message": "string",
  "contracts": [
    {
      "id": "string",
      "organizationType": "string",
      "personType": "base",
      "name": "string",
      "tariff": "string"
    }
  ]
}
```

```
        "description": "string",
        "active": false,
        "global": false
    }
],
"selectedContract": {
    "id": "string",
    "organizationType": "string",
    "personType": "base",
    "name": "string",
    "description": "string",
    "active": false,
    "global": false
},
"emitents": [
    {
        "id": "string",
        "sn": "string",
        "name": "string",
        "description": "string",
        "orderNumber": 0,
        "orderQuote": 0,
        "active": false,
        "currency": {
            "code": "string",
            "digitalCode": "string",
            "symbol": "string",
            "name": "string",
            "description": "string"
        }
    }
],
"selectedEmitent": {
    "id": "string",
    "sn": "string",
    "name": "string",
    "description": "string",
    "orderNumber": 0,
    "orderQuote": 0,
    "active": false,
    "currency": {
        "code": "string",
        "digitalCode": "string",
        "symbol": "string",
        "name": "string",
        "description": "string"
    }
},
"operationRows": [
    {

```

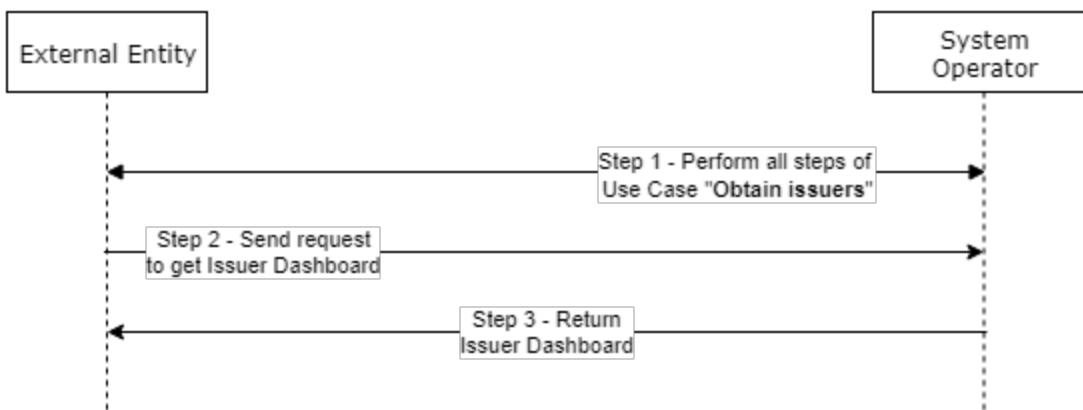
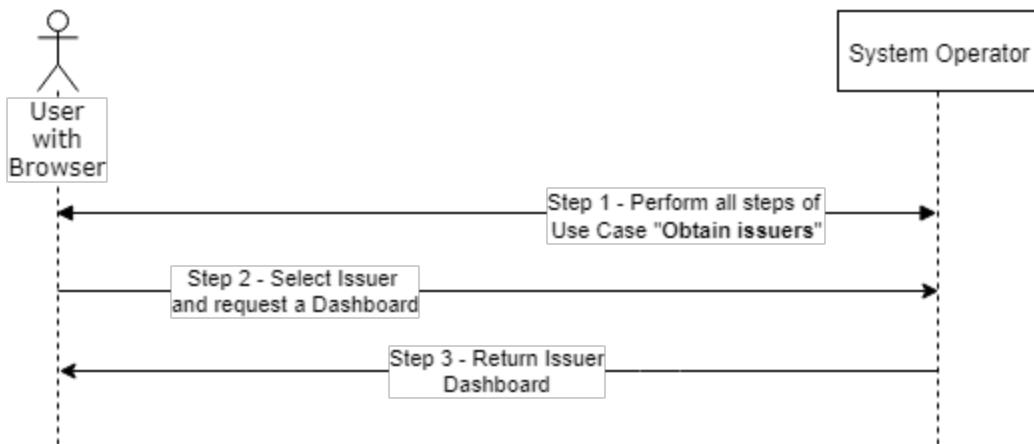
```
"flow": {
    "transactionType": "transfer",
    "processType": "string",
    "message": "string"
},
"columns": [
    {
        "commission": {
            "type": "zero",
            "valuePercent": 0,
            "valueFixed": 0,
            "message": "string"
        },
        "limits": [
            {
                "qualifier": "amount",
                "timeUnit": "per_transaction",
                "value": 0,
                "message": "string"
            }
        ]
    }
],
"gatewayRows": [
    {
        "providerAccount": {
            "providerName": "string"
        },
        "topupColumns": [
            {
                "commission": {
                    "type": "zero",
                    "valuePercent": 0,
                    "valueFixed": 0,
                    "message": "string"
                },
                "limits": [
                    {
                        "qualifier": "amount",
                        "timeUnit": "per_transaction",
                        "value": 0,
                        "message": "string"
                    }
                ]
            }
        ],
        "redeemColumns": [
            {

```

```
        "commission": {
            "type": "zero",
            "valuePercent": 0,
            "valueFixed": 0,
            "message": "string"
        },
        "limits": [
            {
                "qualifier": "amount",
                "timeUnit": "per_transaction",
                "value": 0,
                "message": "string"
            }
        ]
    }
],
"productRows": [
    {
        "providerAccount": {
            "providerName": "string"
        },
        "products": {
        }
    }
]
```

```
        }  
    ]  
}
```

[View dashboard scheme](#)

Use case: View dashboard**Basic FLow****Optional Web UI Flow**

System roles management

View all available system roles description

Use Case Name

View all available system roles

Brief Description

A User or External Entity on behalf of a User with role permission "USER_MANAGER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View all available system roles".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "USER_MANAGER", e.g. head of compliance, administrator, financial specialist, compliance manager, customer support manager, payroll manager or CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View all available system roles".

Endpoint URL: GET /role-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of roles to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "View all available system roles".

Endpoint URL: GET /role-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of roles to User (See Result example below).

Post Conditions

Roles are available.

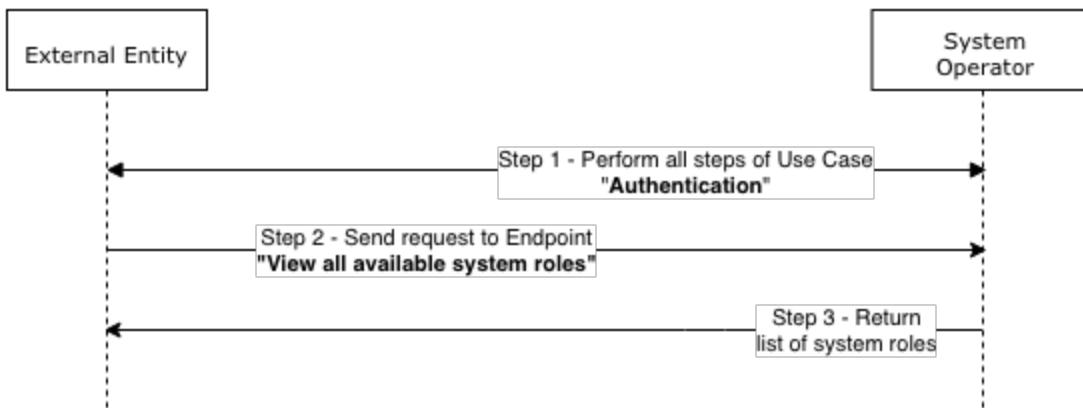
Result example

```
{  
  "records": [  
    {  
      "code": "string"  
    },  
    {"status": "ok",  
     "message": "string"  
   }  
]
```

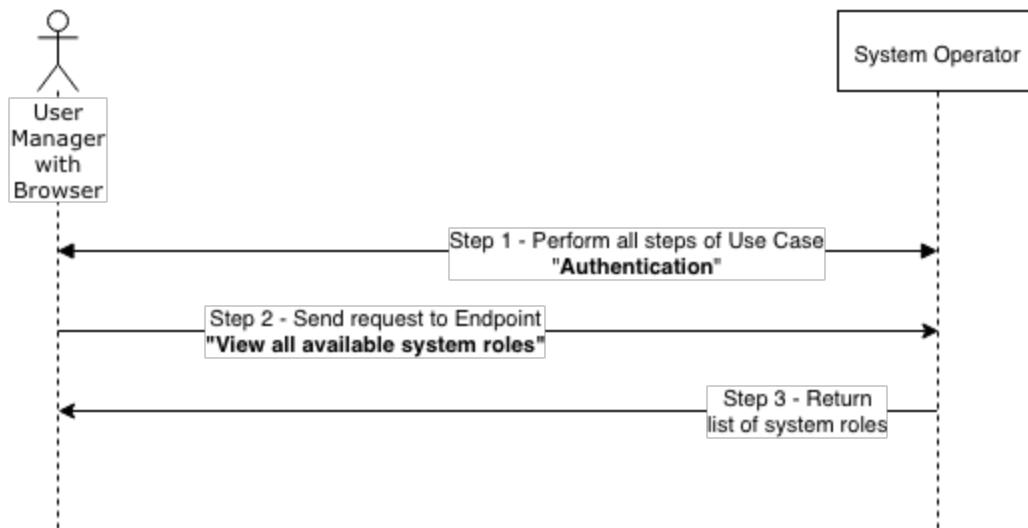
[View all available system roles scheme](#)

Use case: View all available system roles

Basic FFlow



Optional Web UI Flow



User registration as supervisor

Direct registration description

Use Case Name

Direct registration

Brief Description

A User or External Entity on behalf of a User with role permission USER_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Direct registration”.

Notes:

To be registered, the new person who wants to use the System Operator services must provide an email address, which will be used as User ID.

The response to the Endpoint is sent to the User (USER_MANAGER) who made the request.

At the same time an email is sent to the provided email address with all necessary information to add new person to the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MANAGER, e.g. head of compliance, administrator, financial specialist, compliance manager, customer support manager, payroll manager or CFO.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
3. Email service provider

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Direct registration”.

Endpoint URL: POST /direct-registration

Parameters: {

```
"login": "individual_01@abc.com",
"role": "individual"
}
```

1. System Operator returns created User information to External Entity. (See Result example below)
2. System Operator sends email message to the new User with all needed information to enter the System.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Direct registration”.

Endpoint URL: POST /direct-registration

Parameters: {

```
"login": "individual_01@abc.com",
"role": "individual"
}
```

1. System Operator returns created User information to User (See Result example below).
2. System Operator sends email message to the new User with all needed information to enter the System.

Post Conditions

User information, e.g. a valid email address, is available.

Result example

```
{
  "status": "ok",
  "message": "processed successfully",
  "action": "EMAIL_SENT",
  "login": "individual_01@abc.com",
  "password": "C!$KBAC:"
}
```

Email example:

You have successfully registered

SDK.Finance

Dear user!

Log in to your account, click on the link

<https://sandbox.sdk.finance>

Username : individual_01@abc.com

Password : C!\$KBAC:

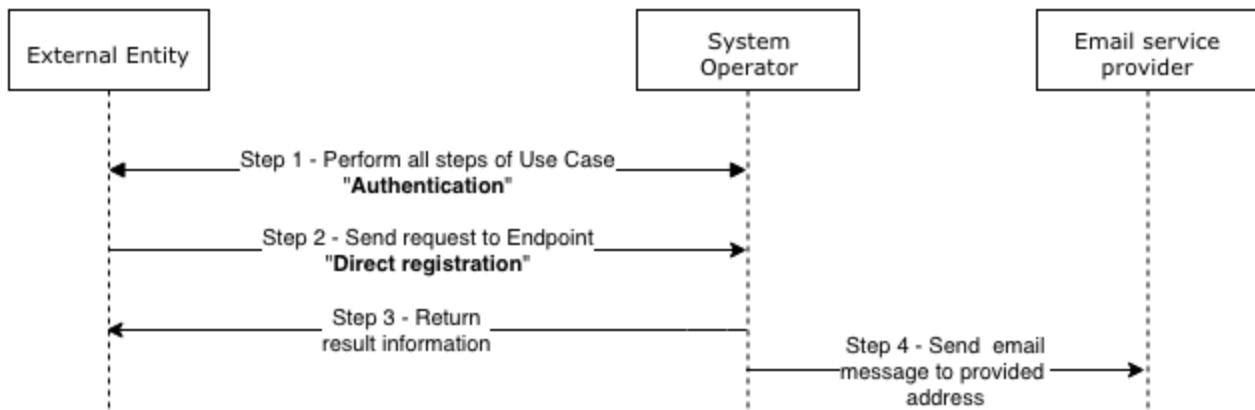
In the case of an erroneous request, please ignore this email

Sincerely, SDK.Finance customer support - support@cyberforcegroup.com.

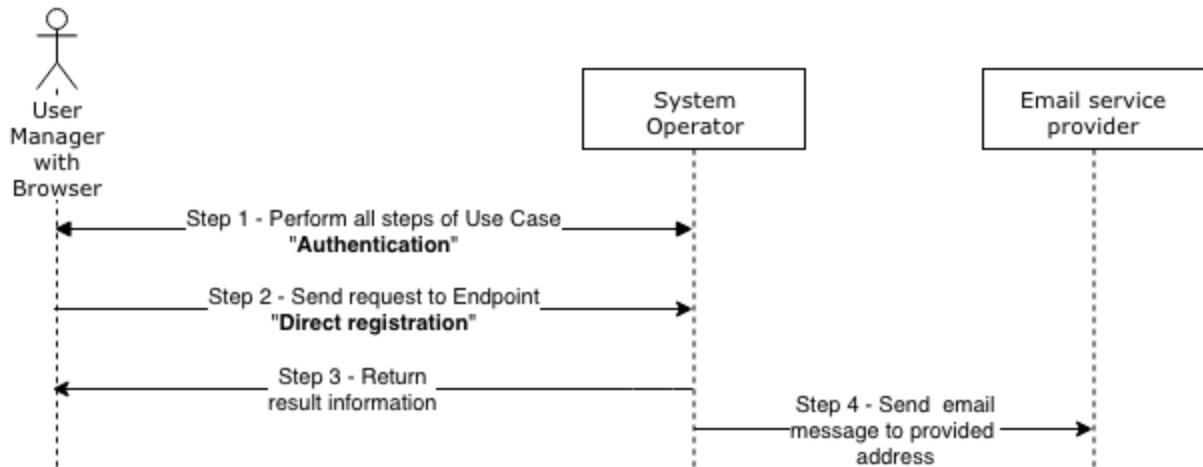
Direct registration scheme

Use case: Direct registration

Basic FFlow



Optional Web UI Flow



User registration

Registration description

Use Case Name

Validate a coin

Brief Description

External Entity or User gets a list of all available Coins, then sends a request to check the Coin status.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to get a list of all Coins.
Endpoint URL: GET /coins
Parameter: Security TOKEN.
3. System Operator returns a List of User Coins to External Entity.
4. External Entity selects a Coin and sends a request to check Coin’s status.
Endpoint URL: POST /coins/validate
Parameters:

```
{
  "serial": "string"
}
```

5. System Operator returns Coin status information (see below in Result Example section)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The user sends a request to get a list of all Coins.
Endpoint URL: GET /coins
Parameter: Security TOKEN.
3. System Operator returns a List of User Coins.
4. The user selects a Coin and sends a request to check Coin’s status.
Endpoint URL: POST /coins/validate
Parameters:

```
{
  "serial": "string"
}
```

5. System Operator returns Coin status information (see below in Result Example section)

Post Conditions

External Entity or User can verify Coin status.

Result example

```
{  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    },  
    "owner": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

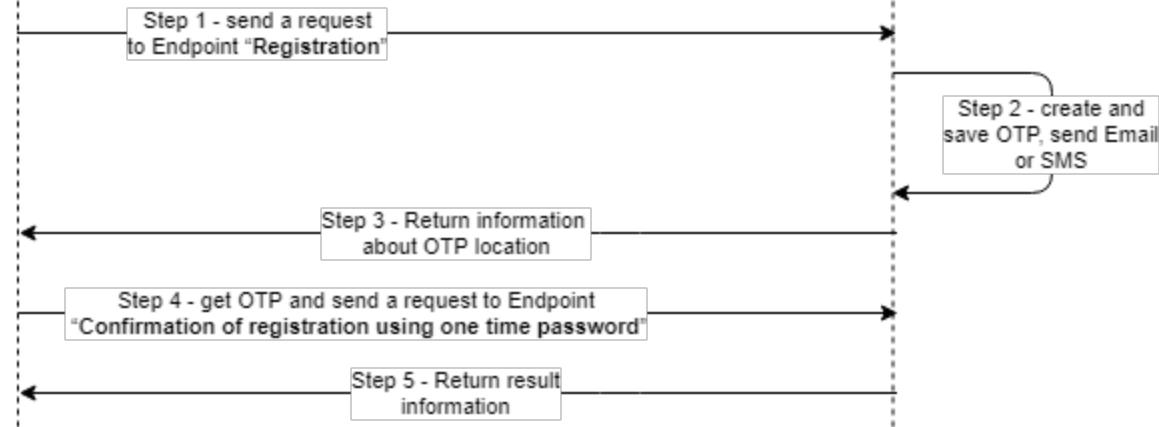
Registration scheme

Use case: Registration

Basic FFlow

External Entity

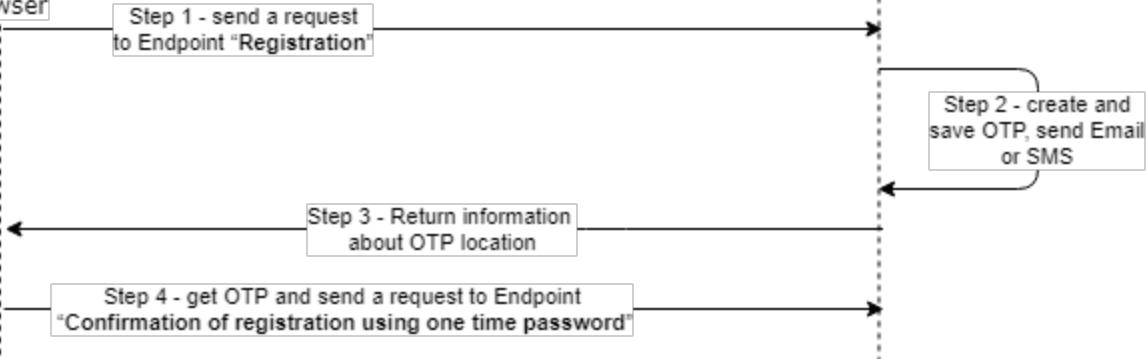
System Operator

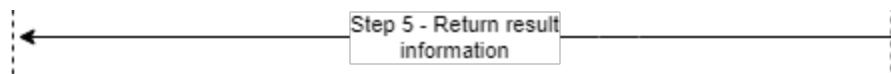


Optional Web UI Flow

User with Browser

System Operator





User management

Clean the ban for user description

Use Case Name

Clean the ban for the user

Brief Description

Authenticated users with roles of Admin and CFO execute steps of Use case Get users. Find a user whose status is “Banned”. Send a request to Endpoint “Clean the ban for user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with the roles of Admin and CFO.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.
2. There are Banned users in the System.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Clean the ban for user”.

Endpoint URL: POST /users/{userId}/unban

Parameter: Security TOKEN.

3. System Operator returns a result confirmation to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Clean the ban for user”.

Endpoint URL: POST /users/{userId}/unban

Parameter: Security TOKEN.

3. System Operator returns a result confirmation to External Entity. (See Result example below)

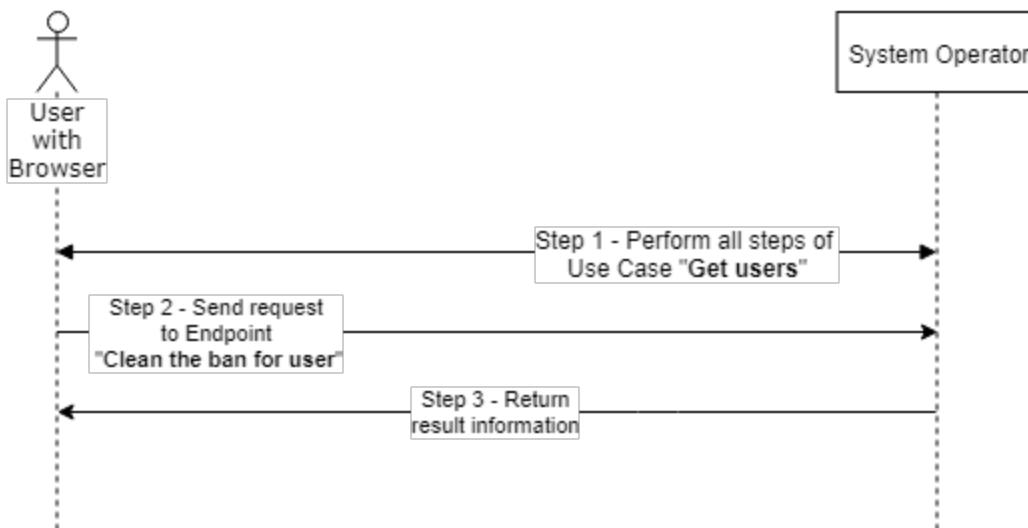
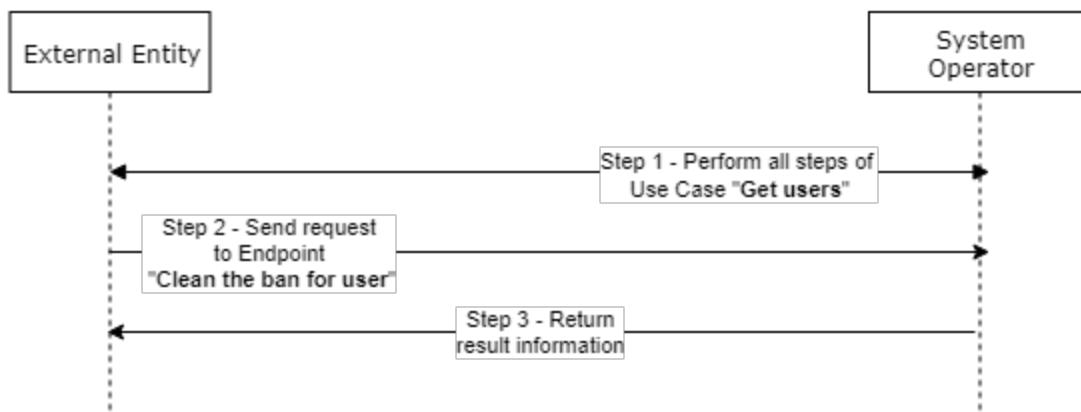
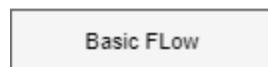
Post Conditions

Earlier banned user is now active.

Result example

```
{  
    "user": {  
        "id": "string",  
        "name": "string",  
        "createdAt": "2018-07-25T12:12:34.347Z",  
        "active": false,  
        "banned": false,  
        "banExpiryDate": "2018-07-25T12:12:34.347Z",  
        "contact": {  
            "phoneNumber": "string",  
            "phoneVerified": false,  
            "email": "string",  
            "emailVerified": false,  
            "countryCode": "AD"  
        },  
        "members": [  
            {  
                "id": "string",  
                "role": "string",  
                "organization": {  
                    "id": "string",  
                    "type": "string",  
                    "name": "string"  
                }  
            }  
        ]  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Clean the ban for user scheme

Use case: Clean the ban for user

Delete a user description

Use Case Name

Delete a user

Brief Description

Authenticated users with role of USER_MANAGER executes steps of Use case Get users. Finds a user whose status is “Banned”. Sends a request to Endpoint “Clean the ban for user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with the roles of USER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile.
2. There are Banned users in the System.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Delete a user”.

Endpoint URL: DELETE /users/{userId}

Parameter: Security TOKEN.

3. System Operator returns a result confirmation to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Delete a user”.

Endpoint URL: DELETE /users/{userId}

Parameter: Security TOKEN.

3. System Operator returns a result confirmation to External Entity. (See Result example below)

Post Conditions

The user is not in the System anymore.

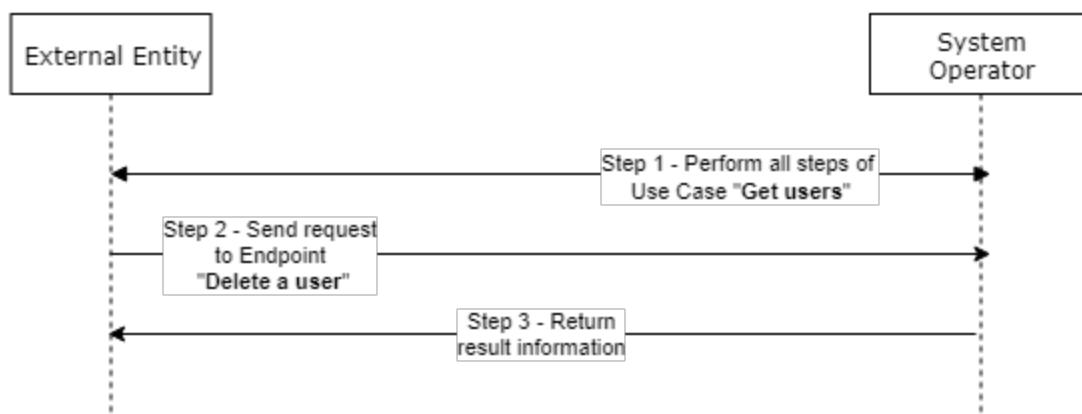
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

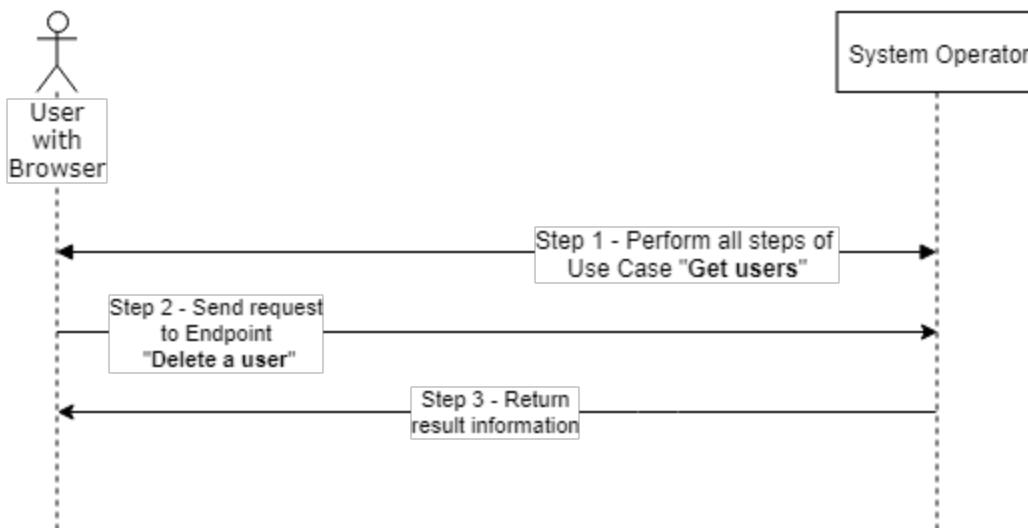
Delete a user scheme

Use case: Delete a user

Basic FLow



Optional Web UI Flow



Get profile history description

Use Case Name

Get profile history

Brief Description

The user or External entity acting as a User with Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER can access the list of System Users, pick up a particular User and request a profile history for some date range.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with the following Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile by id”.
2. External Entity sends a request to Endpoint “Get profile history”.

Endpoint URL: POST /profiles/{userId}/history

Parameter:

```
{
  "dateFrom": "2018-07-25T12:12:32.409Z",
  "dateTo": "2018-07-25T12:12:32.409Z"
}
```

3. System Operator returns requested information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile by id”.
2. A user sends a request to Endpoint “Get profile history”.

Endpoint URL: POST /profiles/{userId}/history

Parameter:

```
{
  "dateFrom": "2018-07-25T12:12:32.409Z",
  "dateTo": "2018-07-25T12:12:32.409Z"
}
```

3. System Operator returns a web page with a List of Issuers to User (See Result example below).

Post Conditions

History is available to the requestor.

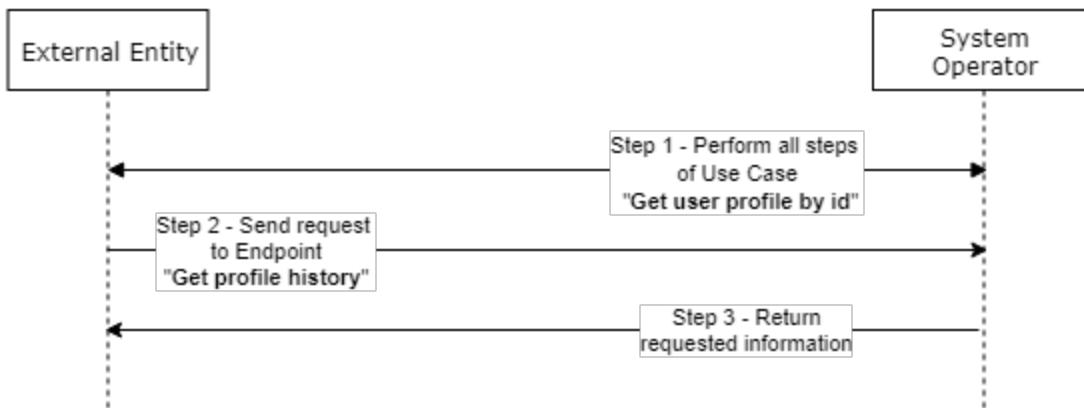
Result example

```
{  
    "profileHistory": {  
        "personHistory": {  
            },  
        "contactHistory": {  
            },  
        "typeHistory": {  
            },  
        "statusHistory": {  
            },  
        "businessHistory": {  
            },  
        "addressHistory": {  
            },  
        "additionalHistory": {  
            }  
    "status": "ok",  
    "message": "string"  
}
```

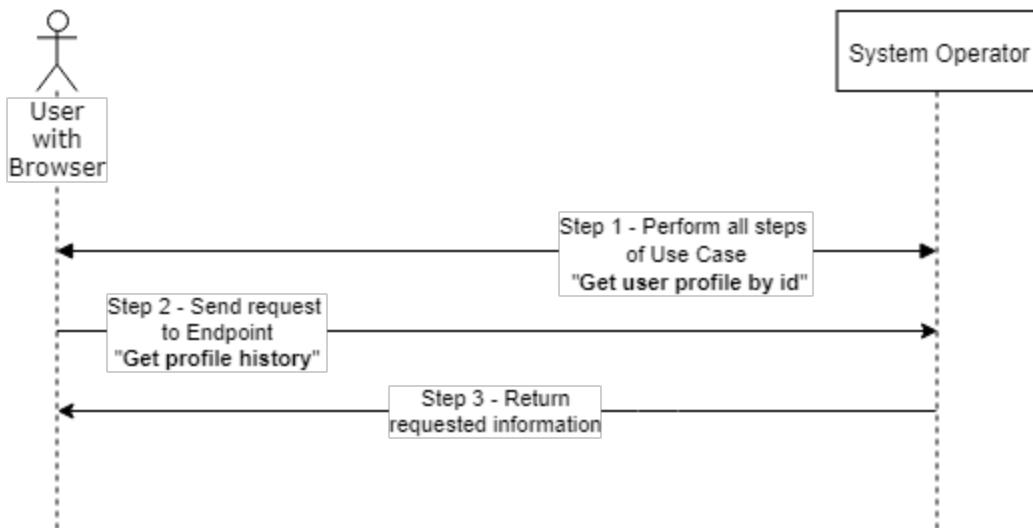
Get profile history scheme

Use case: Get profile history

Basic FLow



Optional Web UI Flow



Get user profile by id description

Use Case Name

Get profile history

Brief Description

The user or External entity acting as a User with Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER can access list of System Users, pick up a particular User and request a profile history for some date range.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with the following Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile by id”.
2. External Entity sends a request to Endpoint “Get profile history”.

Endpoint URL: POST /profiles/{userId}/history

Parameter:

```
{
  "dateFrom": "2018-07-25T12:12:32.409Z",
  "dateTo": "2018-07-25T12:12:32.409Z"
}
```

3. System Operator returns requested information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile by id”.
2. A user sends a request to Endpoint “Get profile history”.

Endpoint URL: POST /profiles/{userId}/history

Parameter:

```
{
  "dateFrom": "2018-07-25T12:12:32.409Z",
  "dateTo": "2018-07-25T12:12:32.409Z"
}
```

3. System Operator returns a web page with a List of Issuers to User (See Result example below).

Post Conditions

History is available to the requestor.

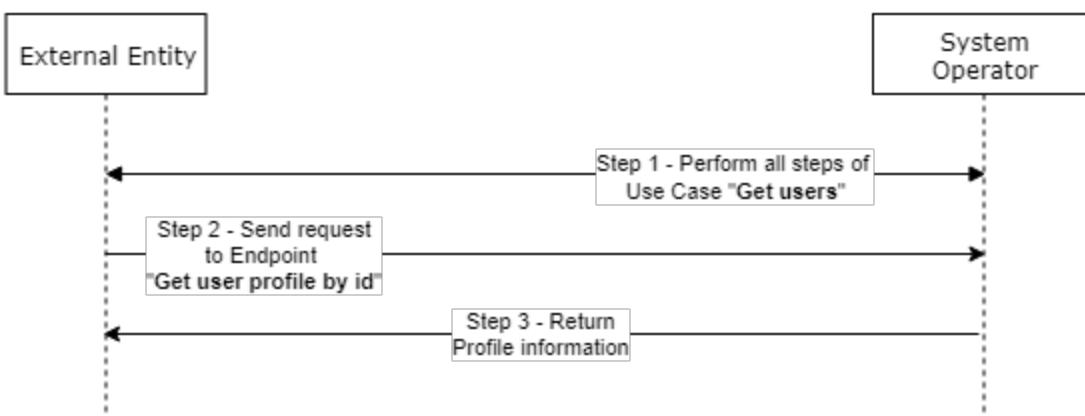
Result example

```
{  
    "profileHistory": {  
        "personHistory": {  
            },  
        "contactHistory": {  
            },  
        "typeHistory": {  
            },  
        "statusHistory": {  
            },  
        "businessHistory": {  
            },  
        "addressHistory": {  
            },  
        "additionalHistory": {  
            }  
    "status": "ok",  
    "message": "string"  
}
```

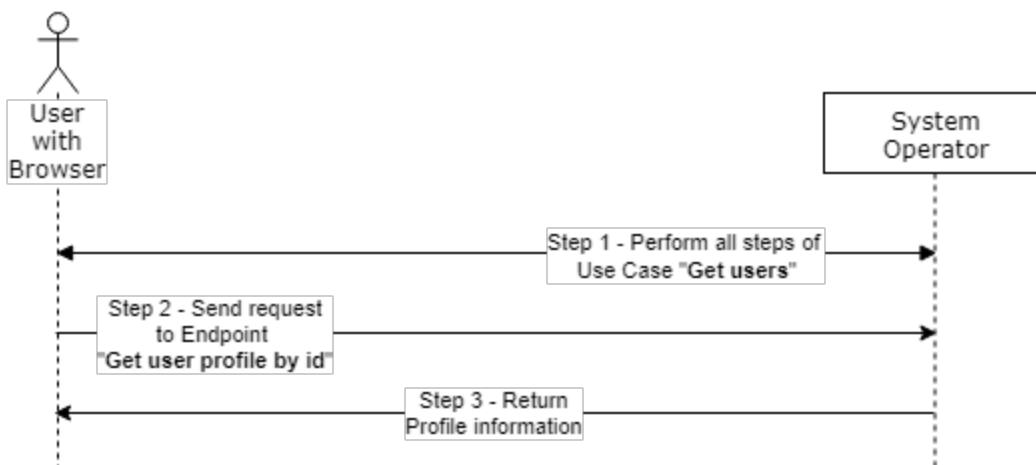
Get user profile by id scheme

Use case: Get user profile by id

Basic Flow



Optional Web UI Flow



Get users description

Use Case Name

Get users

Brief Description

Use Case allows an External Entity or an Individual to request a list of all Users registered with System Operator. The selection is based on the filtering parameters submitted with the request.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to System Operator with filtering information to get a list of Users.

Endpoint URL: POST /users/view

Parameters:

```
{
  "filter": {
    "ids": [
      "string"
    ],
    "email": "string",
    "emailVerified": false,
    "phone": "string",
    "phoneVerified": false,
    "text": "string",
    "banned": false,
    "active": false,
    "roles": [
      "string"
    ],
    "organizationIds": [
      "string"
    ]
  },
  "sort": {
    "date": "asc",
    "active": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns a List of Users to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. The user sends a request from the Browser to System Operator with filtering information to get a list of Users.

Endpoint URL: POST /users/view

Parameters:

```
{  
    "filter": {  
        "ids": [  
            "string"  
        ],  
        "email": "string",  
        "emailVerified": false,  
        "phone": "string",  
        "phoneVerified": false,  
        "text": "string",  
        "banned": false,  
        "active": false,  
        "roles": [  
            "string"  
        ],  
        "organizationIds": [  
            "string"  
        ]  
    },  
    "sort": {  
        "date": "asc",  
        "active": "asc"  
    },  
    "pageNumber": 0,  
    "pageSize": 0  
}
```

3. System Operator returns a web page with a List of Users.

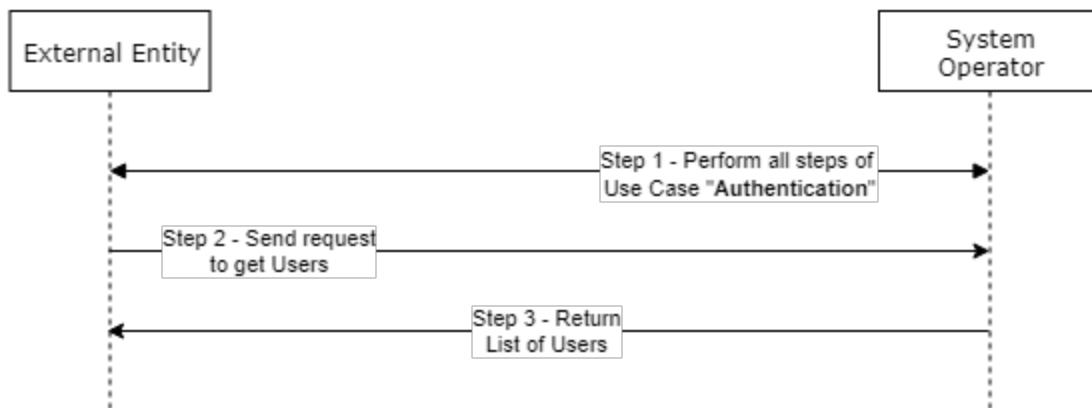
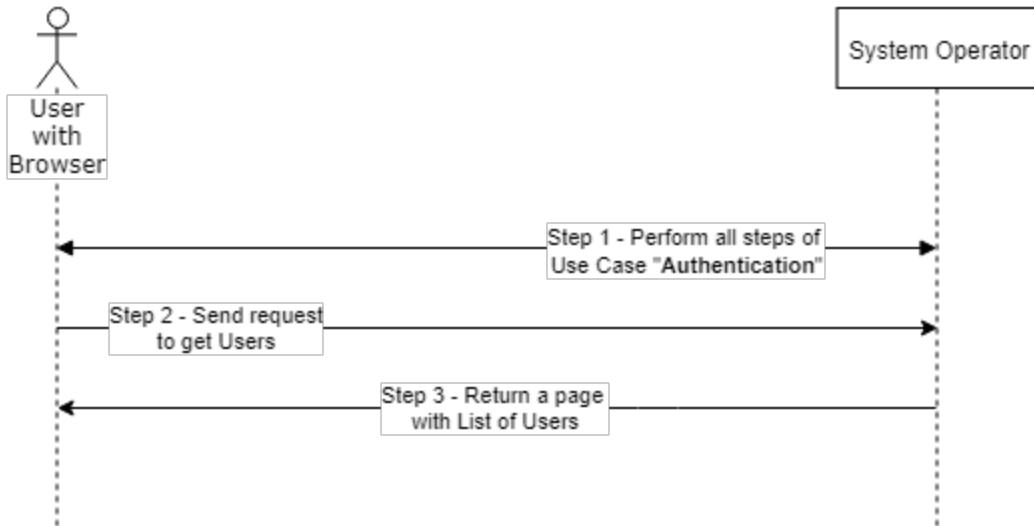
Post Conditions

External Entity or User can see requested Users.

Result

```
{  
  "users": [  
    {  
      "userId": "string",  
      "name": "string",  
      "createdAt": "2018-06-25T11:17:06.079Z",  
      "active": false,  
      "banned": false,  
      "banExpiryDate": "2018-06-25T11:17:06.079Z",  
      "contact": {  
        "phoneNumber": "string",  
        "phoneVerified": false,  
        "email": "string",  
        "emailVerified": false,  
        "countryCode": "AD"  
      },  
      "members": [  
        {  
          "id": "string",  
          "role": "string",  
          "organization": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
          }  
        }  
      ]  
    }  
  ]  
}
```

Get users scheme

Use case: Get users**Basic FLow****Optional Web UI Flow**

Register a new user description

Use Case Name

Register a new user

Brief Description

A User or External Entity on behalf of a User with role permission USER_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Register a new user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”
2. External Entity sends a registration request to System Operator.

Endpoint URL: POST /users

Parameters:

```
{
  "login": "string",
  "role": "string",
  "organizationId": "string",
  "legalType": "string"
}
```

3. System Operator creates a new User and send back to External Entity a result status.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”
2. External Entity sends a registration request to System Operator.

Endpoint URL: POST /users

Parameters:

```
{
  "login": "string",
  "role": "string",
  "organizationId": "string",
  "legalType": "string"
}
```

3. System Operator creates a new User and send back to External Entity a result status.

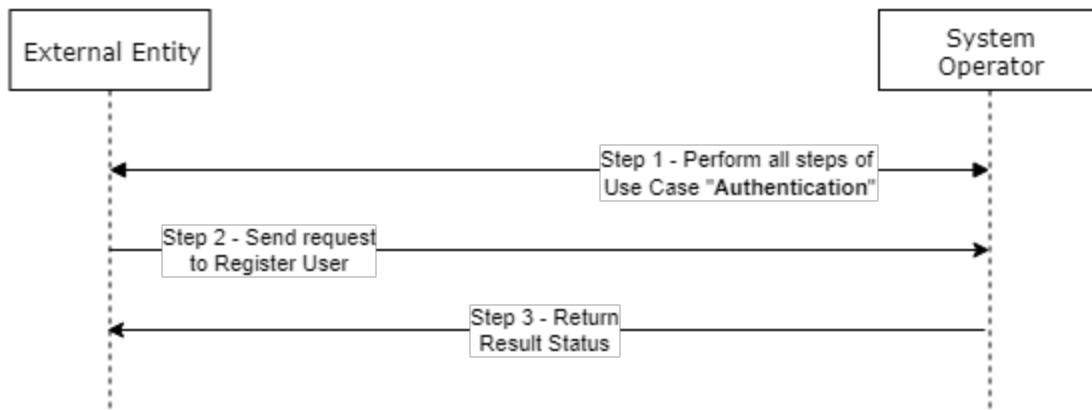
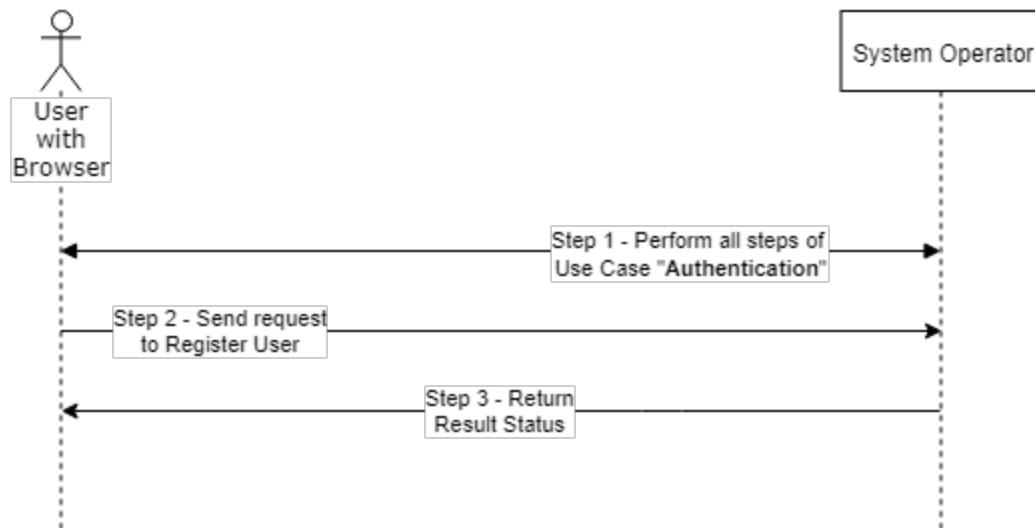
Post Conditions

The new user can connect to the system and see Coin information.

Result example

```
{
  "user": {
    "id": "4b597d17-73f5-4239-a74b-a98e3836179c",
    "name": "",
    "createdAt": "2018-08-23T13:14:48.527Z",
    "active": true,
    "banned": false,
    "banExpiryDate": null,
    "contact": {
      "phoneNumber": null,
      "phoneVerified": false,
      "email": "ind01@mailinator.com",
      "emailVerified": true,
      "countryCode": null
    },
    "members": [
      {
        "id": "391f56ec-748a-4819-8610-4a9cad9f5011",
        "role": "individual",
        "organization": {
          "id": "d1e313da-cfbd-4709-bf85-8755549f2f1f",
          "type": "individual",
          "name": ""
        }
      }
    ]
  },
  "login": "ind01@mailinator.com",
  "password": "FDZX563:",
  "status": "ok",
  "message": "processed successfully"
}
```

Register a new user scheme

Use case: Register a new user**Basic FLow****Optional Web UI Flow**

Reset user's password and to send it to verified user contacts: phone or email description

Use Case Name

Reset user's password and to send it to verified user contacts: phone or email

Brief Description

The user or External entity acting as a User with Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER can access the list of System Users, pick up a particular User, reset User password and send it via email or SMS.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with the following Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile by id”.
 2. External Entity sends a request to Endpoint “Reset user's password and to send it to verified user contacts: phone or email”.
- Endpoint URL: POST /profiles/{userId}/reset-password
3. System Operator sends a new password to User via email or SMS.
 4. System Operator returns requested information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile by id”.
 2. A user sends a request to Endpoint “Reset user's password and to send it to verified user contacts: phone or email”.
- Endpoint URL: POST /profiles/{userId}/reset-password
3. System Operator sends a new password to User via email or SMS.
 4. System Operator returns a web page with a List of Issuers to User (See Result example below).

Post Conditions

History is available to the requestor.

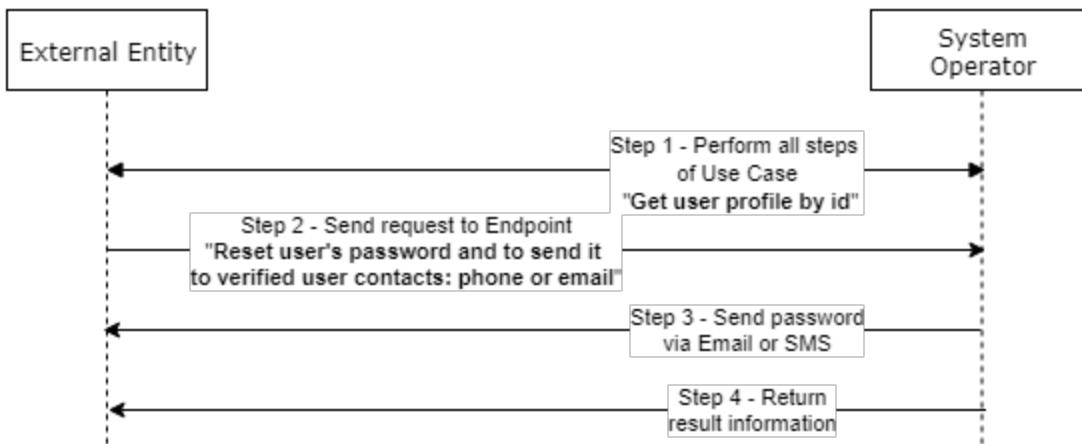
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

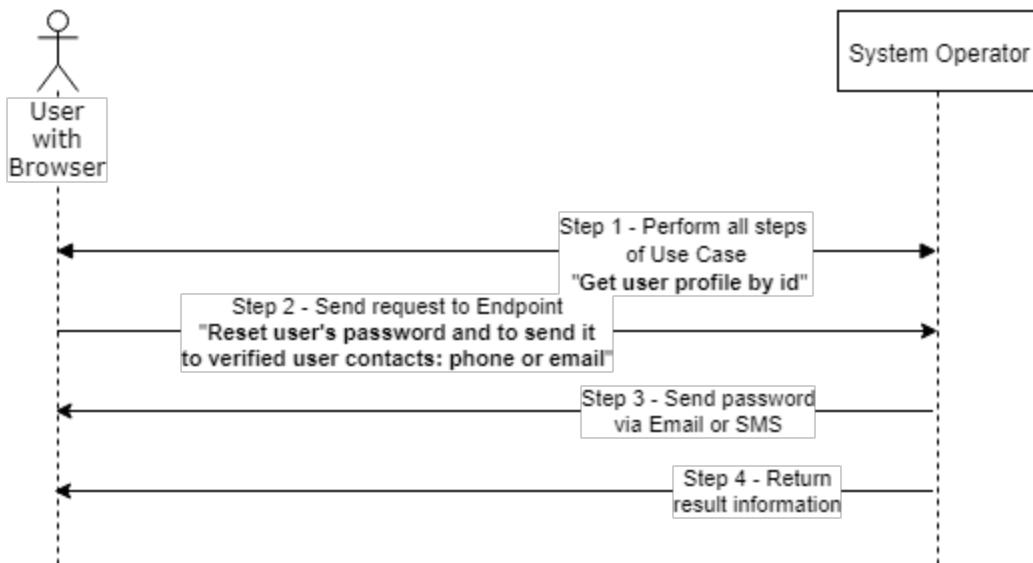
Reset user's password and to send it to verified user contacts: phone or email scheme

Use case: Reset user's password and to send it to verified user contacts: phone or email

Basic FLow



Optional Web UI Flow



Update user description

Use Case Name

Update user

Brief Description

A User or External Entity acting as System User goes through all steps of “Authenticate” Use Case.

Then several update operations performed, including:

Update additional information

Update an existing user

Update and verify contact without confirmation

Update business information

Update person information

Update security settings

Update user address

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with role PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with role PROFILE_OWNER.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile by id”.
2. External Entity sends a request to Endpoint “Update additional information”.

Endpoint URL: PATCH /profiles/{userId}/additional

Parameter:

```
{
  "additional": {
    }
}
```

3. External Entity sends a request to Endpoint “Update an existing user”.

Endpoint URL: PATCH /users/{userId}

Parameter:

```
{
  "active": false
}
```

4. External Entity sends a request to Endpoint “Update and verifies contact without confirmation”.

Endpoint URL: PATCH /users/{userId}

Parameter:

```
{
  "active":false
}
```

5. External Entity sends a request to Endpoint “Update business information”.

Endpoint URL: PATCH /profiles/my/business

Parameter:

```
{
  "companyName": "string",
  "type": "eshop",
  "vat": "string",
  "administrator": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string"
  }
}
```

6. External Entity sends a request to Endpoint “Update person information”.

Endpoint URL: PATCH /profiles/my/person

Parameter:

```
{
  "person": {
    "namePlain": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "nameIntl": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "description": "string"
  }
}
```

7. External Entity sends a request to Endpoint “Update security settings”.

Endpoint URL: PATCH /profiles/my/security-settings

Parameter:

```
{
  "security": {
    "twoFactorsAuthEnabled": false,
    "secretWord": "string",
    "transactionNotification": {
      "phone": false,
      "email": false
    },
    "authorizationNotification": {
      "phone": false,
      "email": false
    }
  }
}
```

8. External Entity sends a request to Endpoint “Update user address”.

Endpoint URL: PATCH /profiles/{userId}/address

Parameter:

```
{
  "address": {
    "country": "AD",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "houseNumber": "string"
  }
}
```

9. System Operator returns a result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile by id”.

2. A user sends a request to get a list of all Issuers.

Endpoint URL: GET /issuers

Parameter: Security TOKEN.

3. External Entity sends a request to Endpoint “Update an existing user”.

Endpoint URL: PATCH /users/{userId}

Parameter:

```
{  
    "active":false  
}
```

4. External Entity sends a request to Endpoint “Update and verifies contact without confirmation”.

Endpoint URL: PATCH /users/{userId}

Parameter:

```
{  
    "active":false  
}
```

5. External Entity sends a request to Endpoint “Update business information”.

Endpoint URL: PATCH /profiles/my/business

Parameter:

```
{  
    "companyName": "string",  
    "type": "eshop",  
    "vat": "string",  
    "administrator": {  
        "firstName": "string",  
        "lastName": "string",  
        "email": "string",  
        "phone": "string"  
    }  
}
```

6. External Entity sends a request to Endpoint “Update person information”.

Endpoint URL: PATCH /profiles/my/person

Parameter:

```
{
  "person": {
    "namePlain": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "nameIntl": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "description": "string"
  }
}
```

7. External Entity sends a request to Endpoint “Update security settings”.

Endpoint URL: PATCH /profiles/my/security-settings

Parameter:

```
{
  "security": {
    "twoFactorsAuthEnabled": false,
    "secretWord": "string",
    "transactionNotification": {
      "phone": false,
      "email": false
    },
    "authorizationNotification": {
      "phone": false,
      "email": false
    }
  }
}
```

8. External Entity sends a request to Endpoint “Update user address”.

Endpoint URL: PATCH /profiles/{userId}/address

Parameter:

```
{  
    "address": {  
        "country": "AD",  
        "zipCode": "string",  
        "city": "string",  
        "street": "string",  
        "houseNumber": "string"  
    }  
}
```

9. System Operator returns a result information to External Entity. (See Result example below)

Post Conditions

All updated information is reflected on the User profile.

Result example

All updated information is returned back to request the sender with the message and status code:

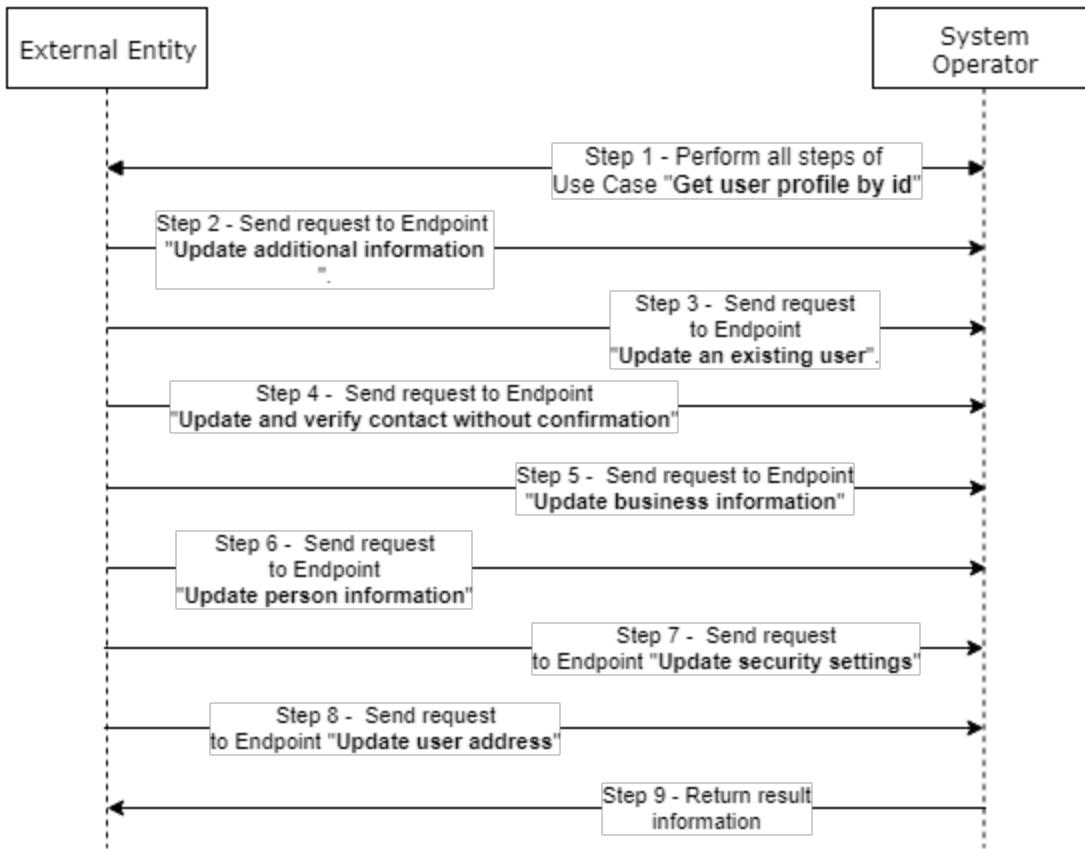
"status": "ok", (or fail)

"message": "string"

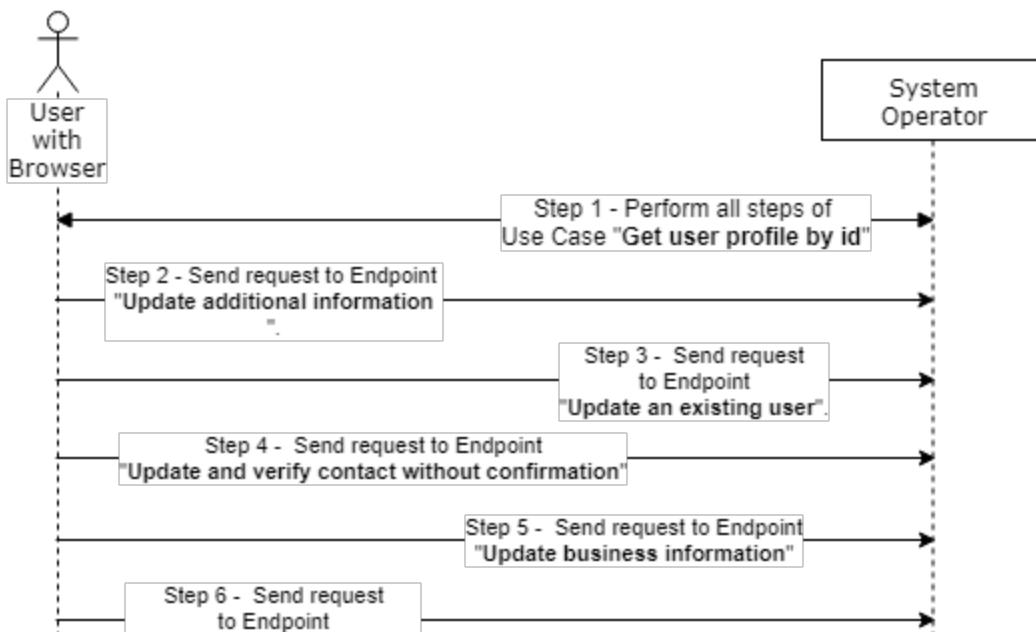
Update user scheme

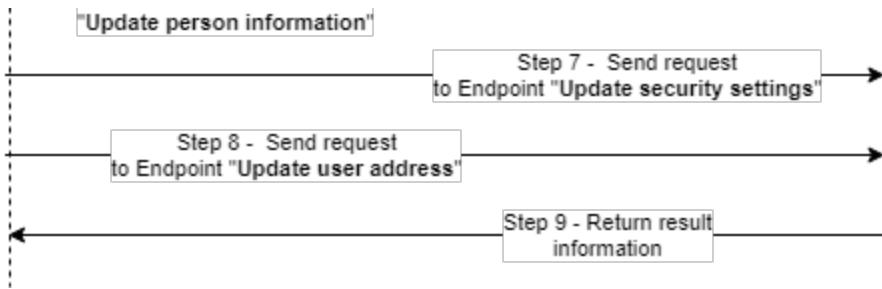
Use case: Update user

Basic FLow



Optional Web UI Flow





Validate secret word description

Use Case Name

Validate secret word

Brief Description

The user or External entity acting as a User with Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER can access the list of System Users, pick up a particular User and send the request to validate a secret word.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with the following Role Permissions: USER_VIEWER, USER_MANAGER, PROFILE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile by id”.
2. External Entity sends a request to Endpoint “Validate secret word”.

Endpoint URL: POST /profiles/{userId}/secret-word

Parameter:

```
{
  "secretWord": "string"
}
```

3. System Operator validates a secret word and returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile by id”.
2. A user sends a request to Endpoint “Validate secret word”.

Endpoint URL: POST /profiles/{userId}/secret-word

Parameter:

```
{
    "secretWord": "string"
}
```

3. System Operator validates a secret word and returns result information to User (See Result example below).

Post Conditions

If the secret word is valid the result is “ok”.

Result example

```
{
    "status": "ok",
    "message": "string"
}
```

Validate secret word scheme

Coin management

Create coin description

Use Case Name

Create coin

Brief Description

After going through all registration steps and approval process a System Operator Authorized User with role permission COIN_OWNER can connect to the Cabinet a create Coins. A User can create multiple coins in the Cabinet. Each Coin is associated with one and only one Currency. On the other hand, there could be multiple Coins associate with the same currency.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with role permission COIN_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. A user must have a System Operator profile.
2. There must be an Issuer for desired Currency

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Obtain issuers”.
2. External Entity selects an Issuer, inputs Coin name, and requests Coin creation.

Endpoint URL: POST /coins

Parameters:

```
{  
    "name": "coin-name",  
    "issuerId": "string",  
    "type": "CLIENT"  
}
```

3. System Operator creates a Coin.
4. System Operator responds with operation confirmation

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.
2. The user selects an Issuer, inputs Coin name, and requests a Coin creation.

Endpoint URL: POST /coins

Parameters:

```
{  
    "name": "coin-name",  
    "issuerId": "string",  
    "type": "client"  
}
```

3. System Operator creates a Coin.
4. System Operator responds with operation result.

Post Conditions

A user can see and edit new Coin.

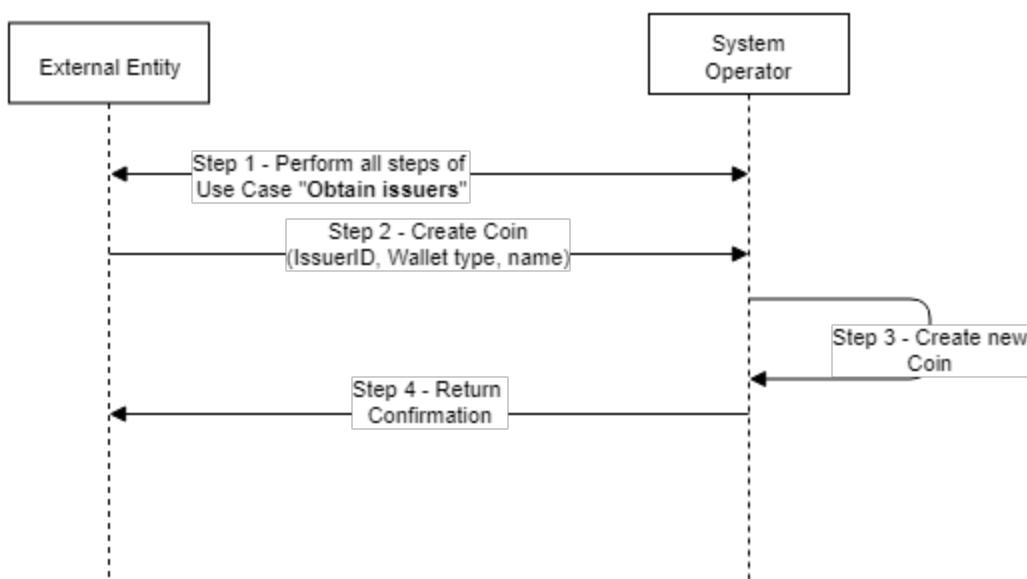
Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "coin": {  
        "serial": "string",  
        "name": "string",  
        "amount": 0,  
        "availableAmount": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "pin": "string"  
}
```

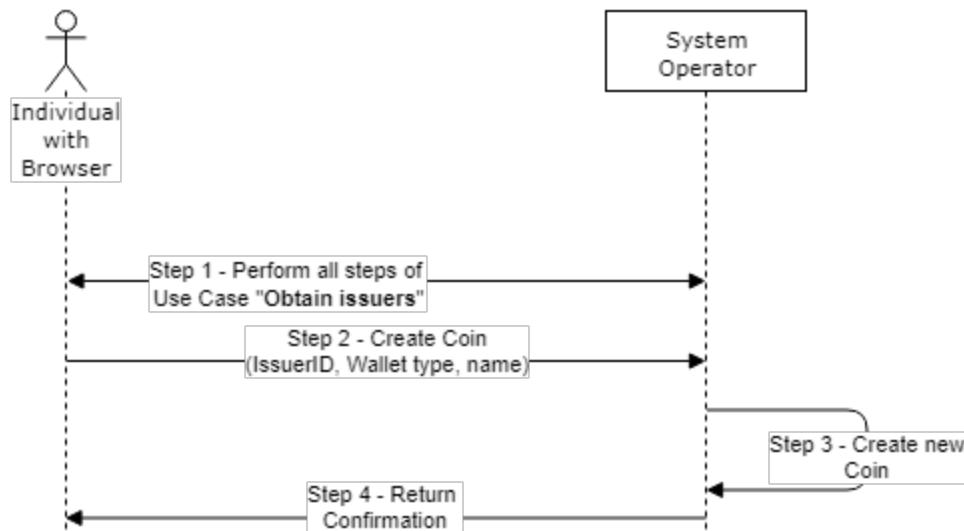
Create coin scheme

Use case: Create Coin

Basic FLow



Optional Web UI Flow



Delete coin description

Use Case Name

Delete coin

Brief Description

A user or External Entity API Caller with role permission COIN_OWNER goes through the Authentication process obtains a security token from System Operator and gets the access to System Operator functions. Then all User Wallets are displayed and Uses can select a particular Wallet for deletion. As a final step User send a request to System Operator to delete this Wallet in System Operator Data Repository.

Actors

1. The user or External Entity with role permission COIN_OWNER calling SDK.Finance API EndPoint.
2. System Operator using SDK.Finance.

Preconditions

1. There must be a profile defined at System Operator.
2. There must be zero balance on the Coin to be deleted.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. Send Request with a Coin to be deleted.

Endpoint URL: DELETE /coins/{serial}

Parameters:

```
{
  "serial": "string"
}
```

3. System Operator deletes specified Coin.

4. System Operator responds with operation result Status.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. Send Request with a Coin to be deleted.

Endpoint URL: DELETE /coins/{serial}

Parameters:

```
{
  "serial": "string"
}
```

3. System Operator deletes specified Coin.

4. System Operator responds with a web page containing operation result Status.

Post Conditions

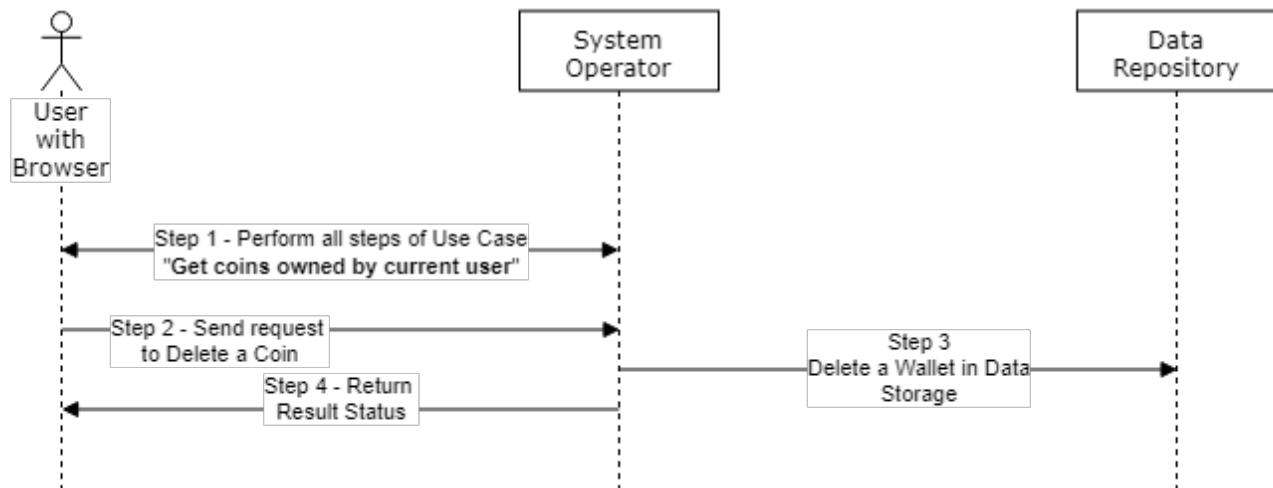
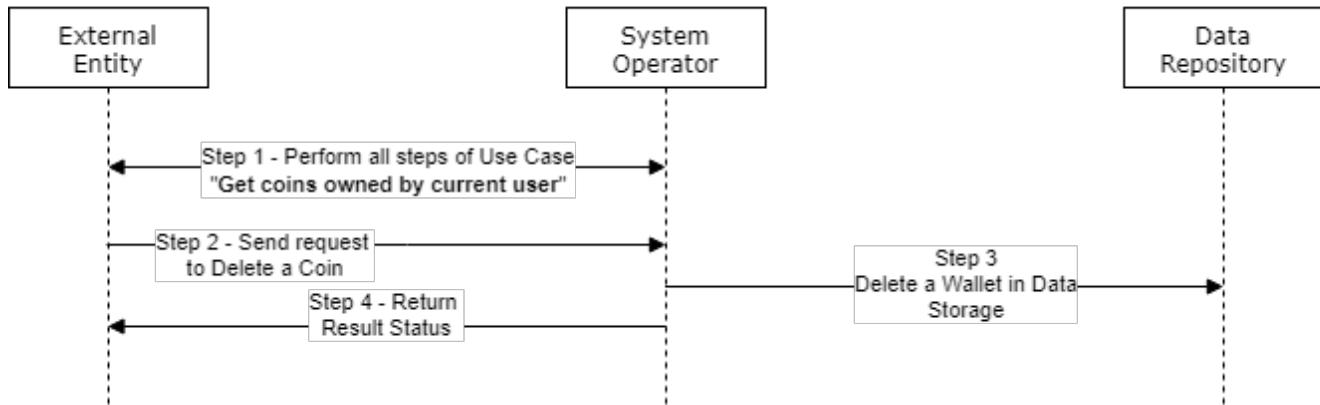
The Wallet is not displayed in User Cabinet.

Result example

```
{  
    "status": "ok / fail",  
    "message": "string"  
}
```

Delete coin scheme

Use case: Delete Coin



Get coins owned by current user description

Use Case Name

Get coins owned by the current user

Brief Description

A user of the system role permission COIN_OWNER sends a request to see all coins' information.

Actors

Individual with role permission COIN_OWNER.

Preconditions

An individual past authentication stage and has a valid security token.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to System Operator to see all coins which belong to a User Profile.

Endpoint URL: GET /coins

Parameters: TOKEN

3. Using token, System Operator retrieves all individual's coins
4. System Operator returns complete information about individual's coins.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. An individual sends a request to see all coins.
3. Using token System Operator retrieves all individuals coins
4. System Operator displays complete information about individual's coins.

Exception Flows

An individual has not created any coins yet. Corresponding information message will be displayed by the System Operator.

Post Conditions

All individual's coins are displayed.

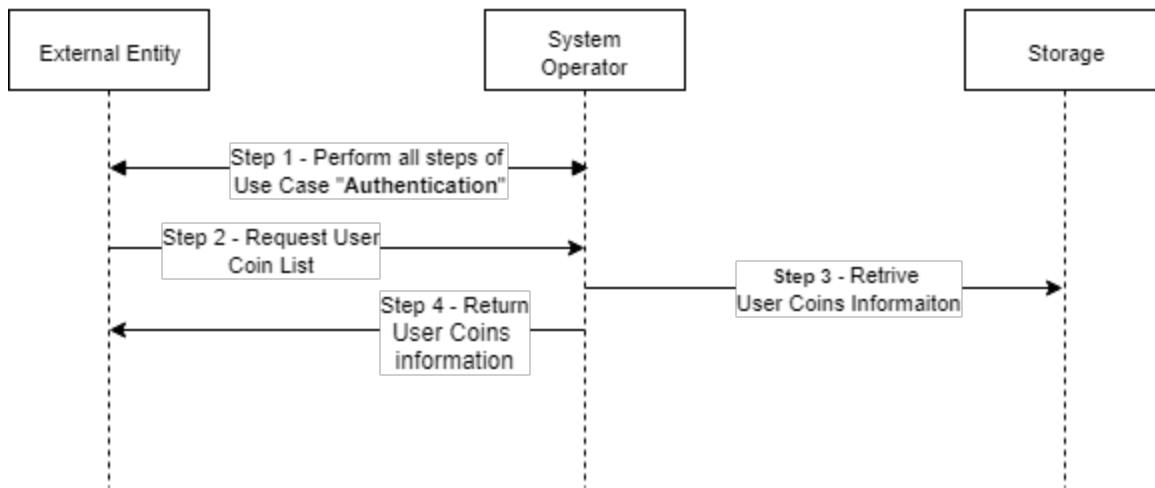
Result example

```
{  
    "coins": [  
        {  
            "serial": "932315898321",  
            "name": "Test UAH",  
            "amount": 9985,  
            "availableAmount": 9985,  
            "issuer": {  
                "id": "2b51bbc0-e112-4723-86c6-7bcb966d6846",  
                "sn": "UAH",  
                "currency": "UAH"  
            },  
            "active": true,  
            "type": "client"  
        },  
        {  
            "serial": "156407768719",  
            "name": "EUR ",  
            "amount": 9863.98,  
            "availableAmount": 9863.98,  
            "issuer": {  
                "id": "ae8fa896-750a-4459-a7a6-e6a3ff2852ee",  
                "sn": "EUR",  
                "currency": "EUR"  
            },  
            "active": true,  
            "type": "client"  
        }  
    ],  
    "status": "ok",  
    "message": "processed successfully"  
}
```

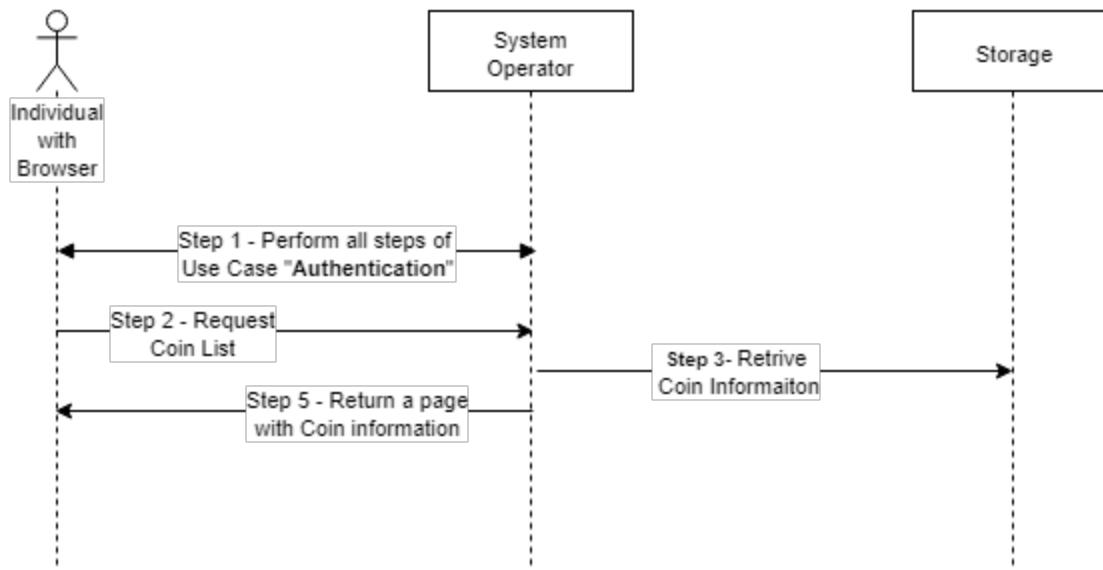
Get coins owned by current user scheme

Use case: Get coins owned by current user

Basic FLow



Optional Web UI Flow



Set coin as main description

Use Case Name

Set coin as main

Brief Description

User or API Caller with role permission COIN_OWNER goes through the Authentication process obtains a security token from System Operator and gets the access to System Operator functions. Then all User Coins are displayed and Uses can set one of the Coins to be the main Coin. As a final step User send a request to System Operator to record this update in System Operator Data Repository.

Actors

1. External Entity or User with role permission COIN_OWNER calling API.
2. System Operator using SDK.Finance.

Preconditions

1. The user must have a profile inside System Operator Datasource.
2. There must have more than one Coin.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. External Entity sends a request to set one Coin as main.

Endpoint URL: POST /coins/set-main

Parameters:

```
{
  "serial": "string"
}
```

3. System Operator updates Coin information in Data Storage.
4. System Operator responds with operation confirmation.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. The individual sends a web request to set one Coin as main.

Endpoint URL: POST /coins/set-main

Parameters:

```
{
  "serial": "string"
}
```

3. System Operator updates Coin information in Data Storage.
4. System Operator responds with operation confirmation.

Post Conditions

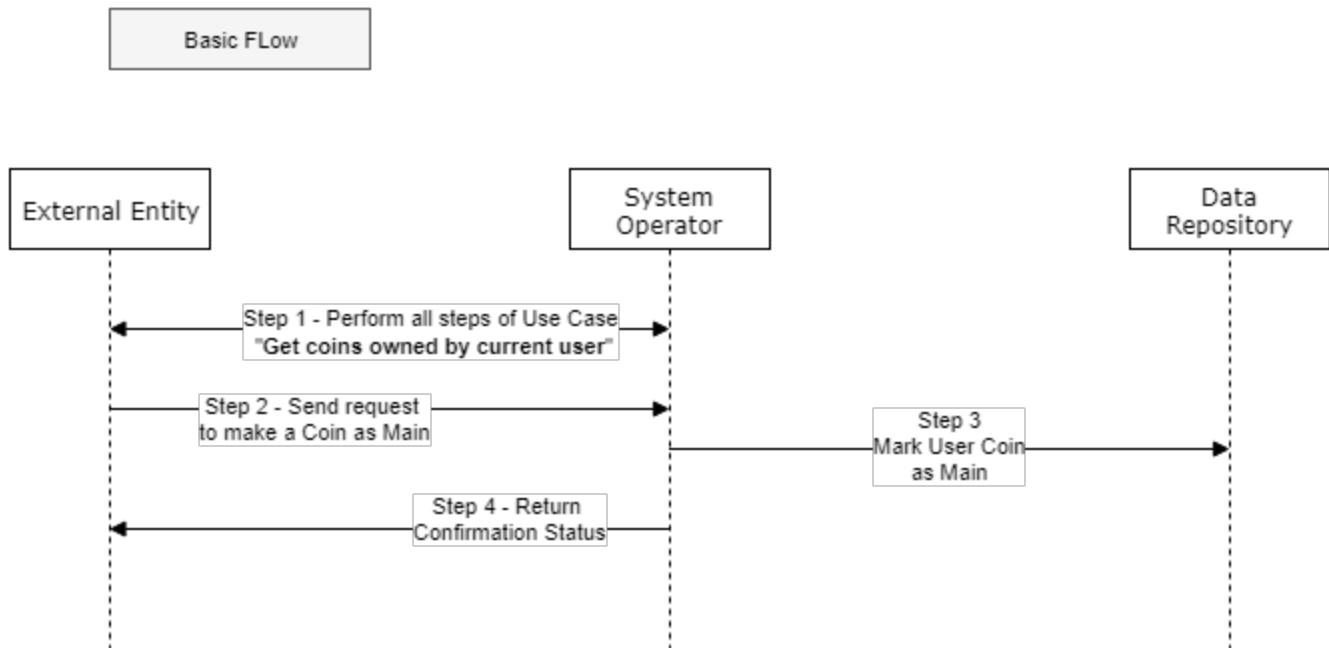
The Coin marked becomes main.

Result example

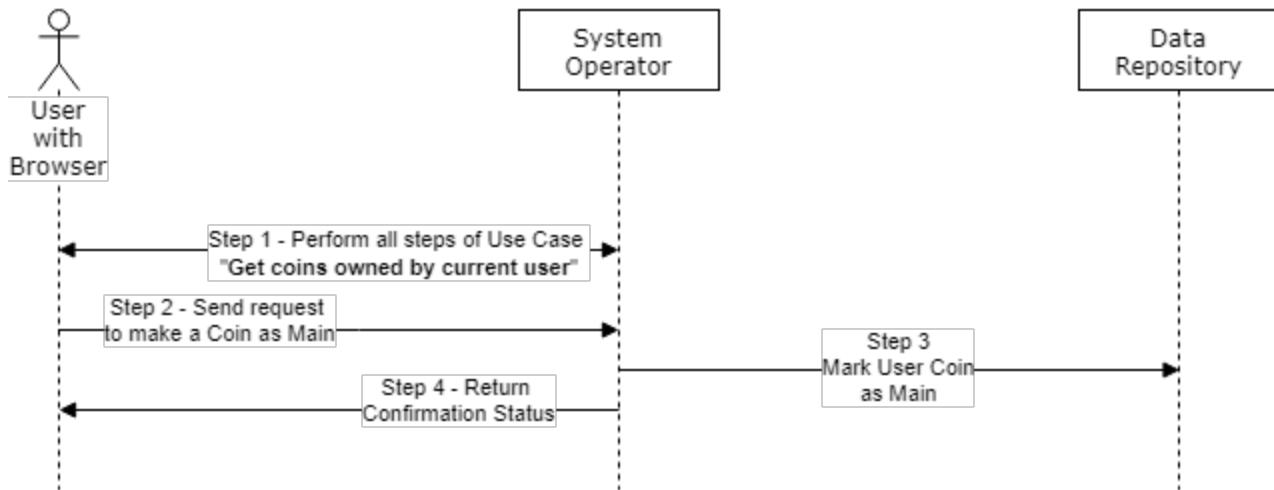
```
{  
    "status": "ok / fail",  
    "message": "string"  
}
```

Set coin as main scheme

Use case: Set coin as main



Optional Web UI Flow



Toggle coin status description

Use Case Name

Toggle coin status

Brief Description

In this Use Case, an authenticated User with role permission COIN_MANAGER sends a request to get all Coins. Then selects a Coin and sends a request to change the status of the Coin. If it was active, it becomes inactive and vice-versa.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with role permission COIN_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator valid profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. External Entity selects a Coin and sends a request to update the status of that Coin.

Endpoint URL: PATCH /coins/{coinID}/status

Parameter: Security TOKEN.

3. System Operator returns a verification message. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. A user selects a Coin and sends a request to update the status of that Coin.

Endpoint URL: PATCH /coins/{coinID}/status

Parameter: Security TOKEN.

3. System Operator returns a verification message.. (See Result example below)

Post Conditions

Status changes to opposite one.

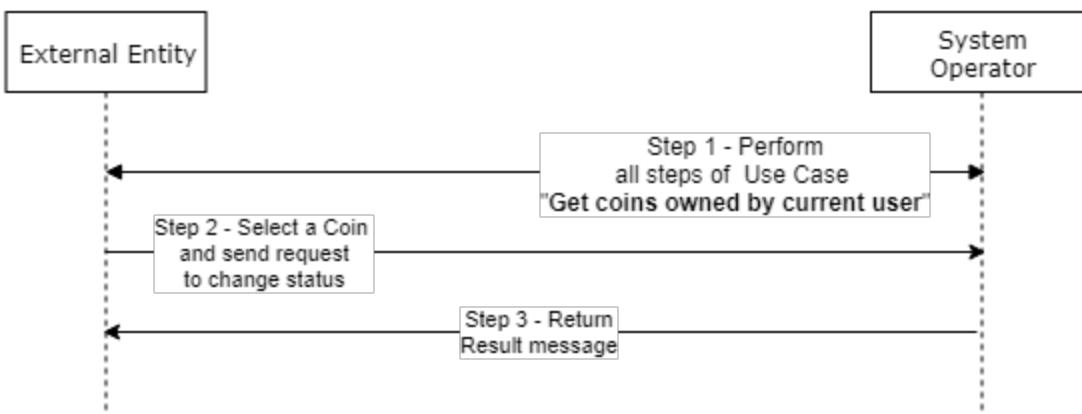
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

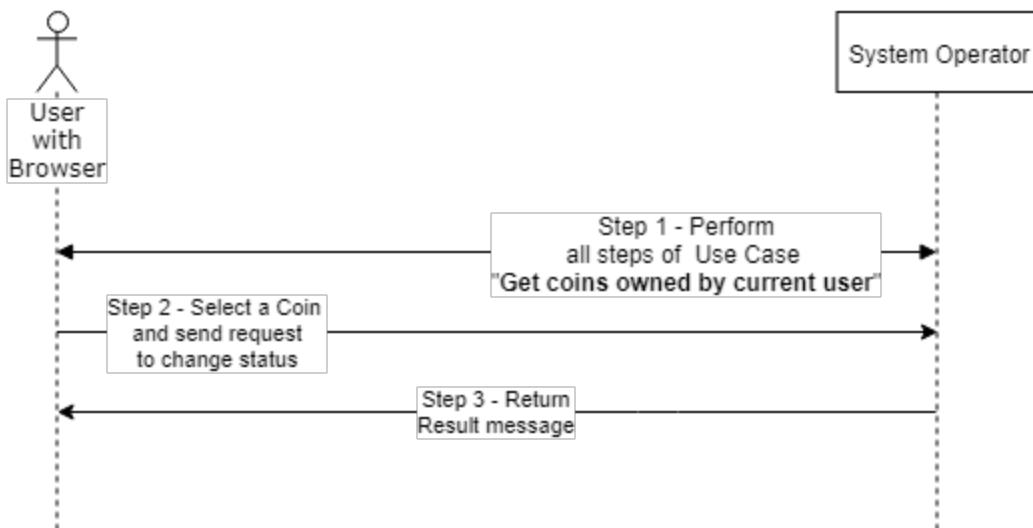
Toggle coin status scheme

Use case: Toggle coin status

Basic FLow



Optional Web UI Flow



Update an existing coin description

Use Case Name

Update an existing coin

Brief Description

User or API Caller with role permission COIN_OWNER goes through the Authentication process obtains a security token from System Operator and gets the access to System Operator functions. Then all User Coins are displayed and Uses can select a particular Coin Details for a needed update. The user makes changes to Coin information and as a final step User sends a request to System Operator to record made changes in the System Operator Data Repository.

Actors

1. External Entity or User of API with role permission COIN_OWNER.
2. System Operator using SDK.Finance

Preconditions

1. The user must have a profile at System Operator.
2. There must be at least one Coin in User's cabinet.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. The user makes changes to Coin Name and sends to System Operator to permanently record the changes.

Endpoint URL: PATCH /coins/{serial}

Parameters:

```
{
  "name" : "string"
}
```

3. System Operator updates specified Coin in System Operator Data Storage.

4. System Operator responds with operation confirmation status.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. The user makes changes to Coin details and sends to System Operator to permanently record the changes.

Endpoint URL: PATCH /coins/{serial}

Parameters:

```
{
  "name" : "string"
}
```

3. System Operator updates specified Coin in System Operator Data Storage.

4. System Operator responds with operation confirmation status.

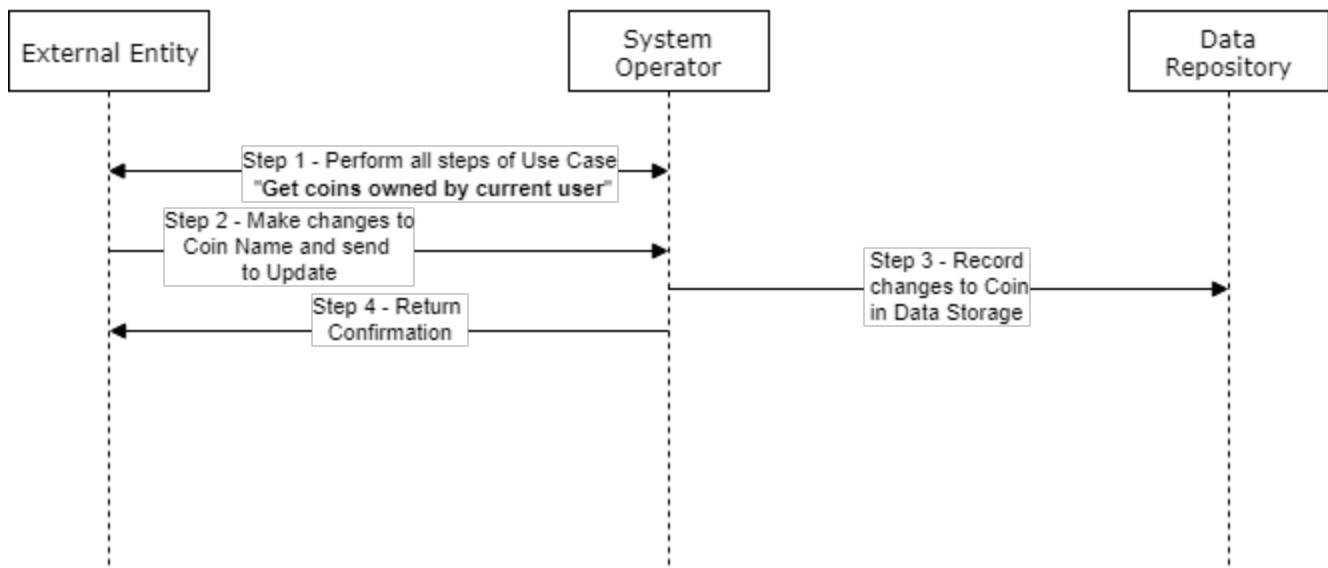
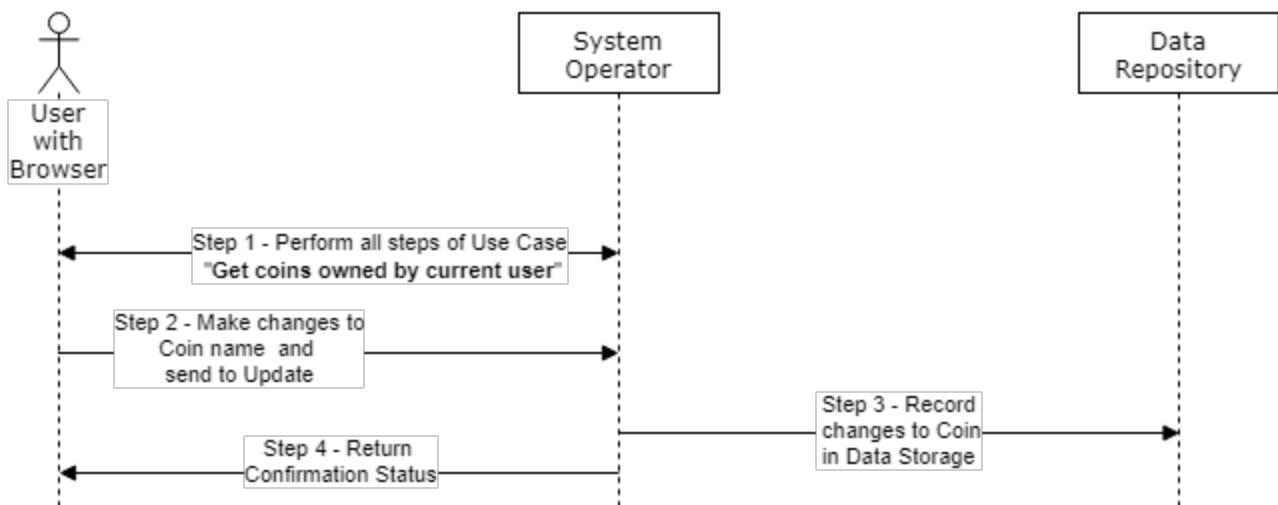
Post Conditions

All changes made to a Coin have taken effect.

Result example

```
{  
    "status": "ok / fail",  
    "message": "string"  
}
```

Update an existing coin scheme

Use case: Updater Wallet**Basic FLow****Optional Web UI Flow**

Authentication

Authentication description

Use Case Name

Authentication

Brief Description

In the context of computer systems, authentication is a process that confirms a user's identity. A User who was already boarded to System Operator makes the first contact with the System Operator site and System Operator confirms that the user has rights to access the system.

Actors

1. External Entity that can interact with System Operator API acting as a Registered User.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. There is a valid registered System Operator User.

Basic Flow

By default, this flow assumes that an External Entity sends Requests to System Operator Endpoints and the System Operator sends back Responses to External Entity.

1. External Entity sends a Request to System Operator API Endpoint URL.

API ENDPOINT <https://sandbox.sdk.finance/api/ui#!/Authorization/login>

Request body:

```
{
  "login": "user1",
  "password": "password11"
}
```

2. System Operator validates credentials.
3. System Operator generates a security token and stores it for Authorization.
4. System Operator sends a Response to External Entity with the following parameters

Response parameters: status, security token, expiration date, role, permissions, etc.

Optional Flow with Web Browser UI

1. An individual using a web browser or mobile application requests a Login page.
2. System Operator sends back a page with input fields for credentials:

Login and Password

3. An individual enters credentials and sends to the System Operator.

Endpoint URL: POST /authorization

Request body:

```
{  
  "login": "string",  
  "password": "string"  
}
```

4. System Operator validates credentials. If they are NOT valid, the logic flow returns to step 2.
5. System Operator generates a unique TOKEN valid for the duration of this interaction and saves the token for future validation.
6. System Operator sends the token back to the Wallet Owner and includes the token in each of the following requests.

Exception Flow

1. Execute Step 1 from Basic Flow.
2. System Operator validates credentials and validation fails three times.
3. System Operator puts the User on Hold and Administrator needs to get involved to generate temp password and to release the User.

Post Conditions

The user is given the Token to interact with the system.

Result example

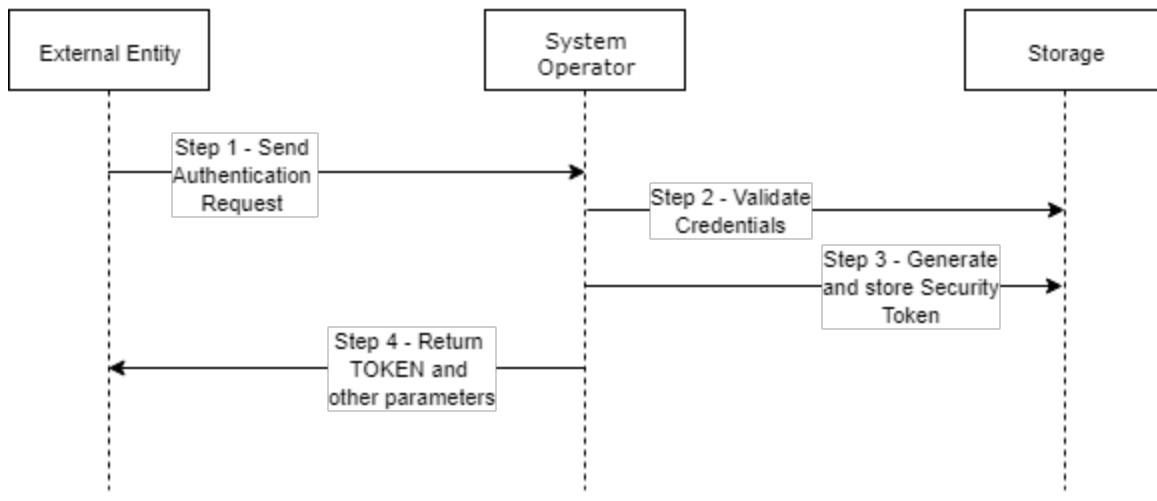
Example of System Operator response:

```
{  
    "status": "ok",  
    "message": "string",  
    "action": "TOKEN_CREATED",  
    "authorizationToken": {  
        "token": "string",  
        "expiresAt": "2018-10-30T08:59:14.386Z"  
    },  
    "members": [  
        {  
            "role": "string",  
            "user": {  
                "id": "string",  
                "name": "string"  
            },  
            "organization": {  
                "id": "string",  
                "type": "string",  
                "name": "string",  
                "identificationStatus": "string",  
                "contract_info": {  
                    "id": "string",  
                    "personType": "base"  
                }  
            },  
            "permissions": [  
                "string"  
            ],  
            "token": {  
                "token": "string",  
                "expiresAt": "2018-10-30T08:59:14.386Z"  
            }  
        }  
    ]  
}
```

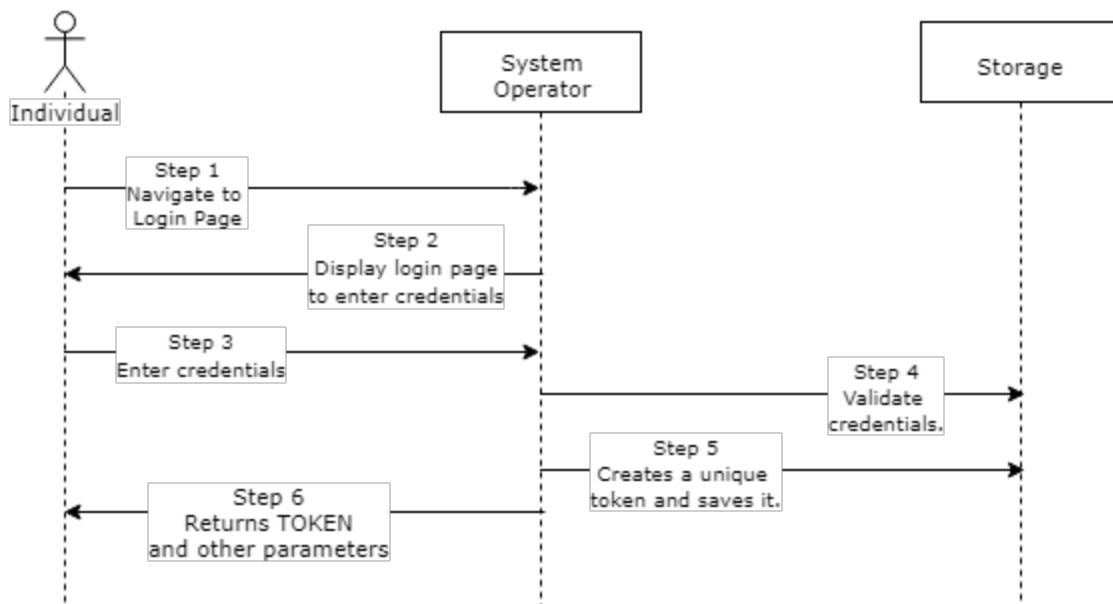
Authentication scheme

Use case: Authentication

Basic FLow



Optional Web UI Flow



Authentication with One Time Password (OTP) description

Use Case Name

Authentication with One Time Password (OTP).

Brief Description

Authentication with OTP, aka two-factor authentication (2FA) is an additional feature provided by System Operator using SDK.Finance. It is a superset of a traditional user/password authentication and could be switched ON and OFF as desired. To provide this type of authentication the system needs to have some email capability or have access to telephony.

With this type of authentication, the user is not immediately served with the Authorization token. Instead, System Operator generates an additional random one-time password and delivers it to the user either to the email address or as a cell phone number provided during initial registration.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running SDK.Finance software and exposing portfolio of financial APIs.
3. Provider of email or telephone communication channel.

Preconditions

1. There is a valid registered System User in System Operator Data Repository.
2. There is a registered telephone number or email address for the System Operator User.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. External Entity sends an Authentication request to System Operator API Endpoint.

Endpoint URL: POST /authorization

Request BODY

```
{
  "login": "string",
  "password": "string"
}
```

2. System Operator performs credentials validation. If credentials are NOT valid, corresponding error message is sent back
3. If credentials are valid, System Operator generates a random One Time Password (OTP) and saves it.
4. System Operator sends OTP to an individual's email address or to the phone number registered during registration.
5. System Operator returns to External Entity generated OTP.
6. External Entity somehow obtains OTP. If OTP has not arrived, External Entity sends a request and Steps 4 and 5 repeated.
7. External Entity sends a request to System Operator with the OTP obtained from an external channel.

Request BODY

```
{
  "login": "user1",
  "password": "otp-password"
}
```

8. System Operator receives OTP from the External Entity and matches it with the earlier saved OTP.
9. If there is a match, System Operator generates and saves a Security TOKEN. If there is NO match, the corresponding error message is sent back.
10. System Operator returns Security TOKEN to External Entity.

Example of System Operator response:

```
{
  "status": "ok",
  "message": "string",
  "action": "TOKEN_CREATED",
  "authorizationToken": {
    "token": "string",
    "expiresAt": "2018-10-30T08:59:14.386Z"
  },
  "members": [
    {
      "role": "string",
      "user": {
        "id": "string",
        "name": "string"
      },
      "organization": {
        "id": "string",
        "type": "string",
        "name": "string",
        "identificationStatus": "string",
        "contract_info": {
          "id": "string",
          "personType": "base"
        }
      },
      "permissions": [
        "string"
      ],
      "token": {
        "token": "string",
        "expiresAt": "2018-10-30T08:59:14.386Z"
      }
    }
  ]
}
```

Optional Flow with Web Browser UI

1. An individual points the web browser to the endpoint URL (/authorization).

API ENDPOINT <https://sandbox.sdk.finance/api/ui#!/Authorization/login>

2. System Operator returns a page with 2 input fields - user ID and password.

3. An individual enters credentials - user ID and password and sends to SO.

Endpoint URL: POST /authorization

Request BODY

```
{
  "login": "string",
  "password": "string"
}
```

4. System Operator performs the validation. If credentials are NOT valid, step 2 is repeated.

5. If credentials are valid, OS generates a random One Time Password (OTP) and saves it.

6. The OTP is sent to an individual's email address or to the phone number.

7. System Operator sends to the browser the page to input OTP.

8. An individual checks Email or Telephone number and gets the OTP. If OTP has not arrived, User sends a request and Steps 5 and 6 repeated.

9. An individual enters OTP on the page and sends to SO.

Request BODY

```
{
  "login": "user1",
  "password": "otp-password"
}
```

10. System Operator receives OTP and matches it with the saved OTP.

11. If there is a match, System Operator generates and saves the security TOKEN.

12. Send Security TOKEN to the browser to use for Authorization of all requests.

Alternative Flow

Steps 1 through 7 get executed.

8. Individual requests to resend OTP the execution logic returns to step 6 and continues as usual.

Post Conditions

Valid Security TOKEN is available for conducting future operations.

Result example

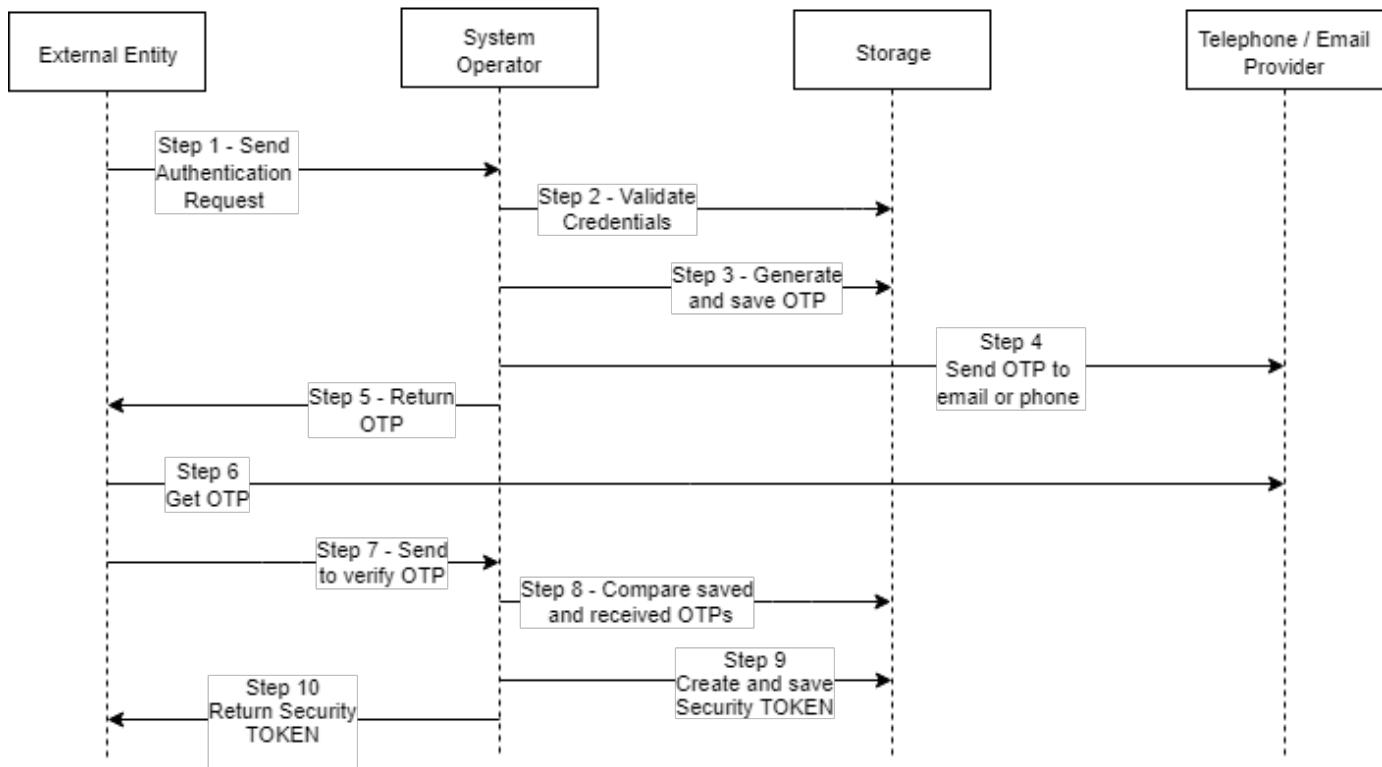
Example of System Operator response:

```
{  
    "status": "ok",  
    "message": "string",  
    "action": "TOKEN_CREATED",  
    "authorizationToken": {  
        "token": "string",  
        "expiresAt": "2018-10-30T08:59:14.386Z"  
    },  
    "members": [  
        {  
            "role": "string",  
            "user": {  
                "id": "string",  
                "name": "string"  
            },  
            "organization": {  
                "id": "string",  
                "type": "string",  
                "name": "string",  
                "identificationStatus": "string",  
                "contract_info": {  
                    "id": "string",  
                    "personType": "base"  
                }  
            },  
            "permissions": [  
                "string"  
            ],  
            "token": {  
                "token": "string",  
                "expiresAt": "2018-10-30T08:59:14.386Z"  
            }  
        }  
    ]  
}
```

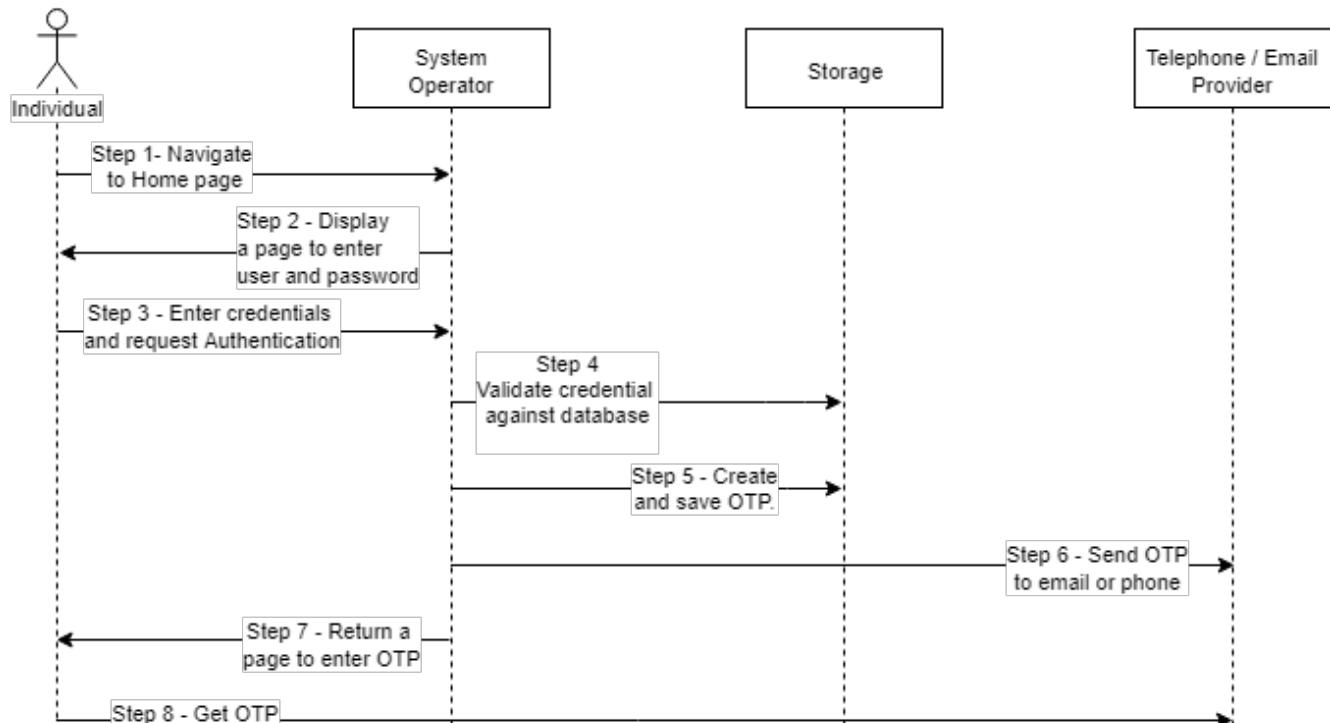
Authentication with OTP scheme

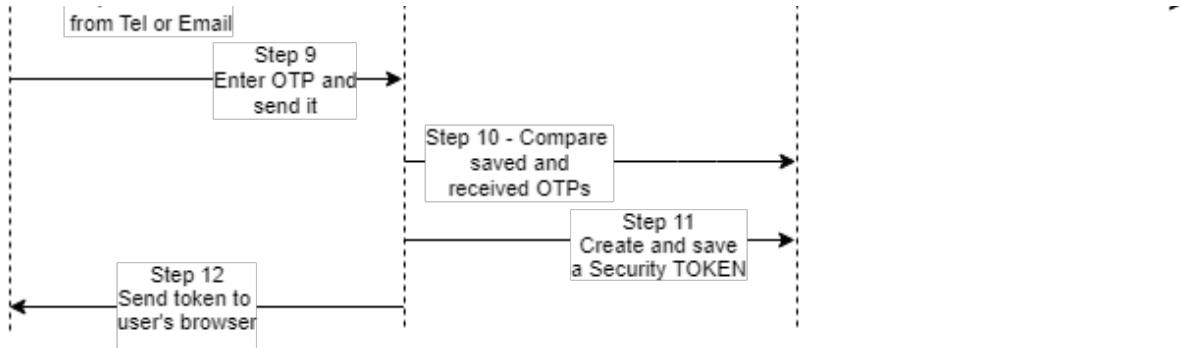
Use case: Authentication with OTP

Basic FLow



Optional Web UI Flow





Password recovery

Send request to restore a password description

Use Case Name

Send request to restore a password

Brief Description

A User of any role forgot the password. The user sends a request to recover password via OTP.

System operator generates and saves OTP, sends email or SMS with OTP and returns information about OTP location to User. User finds OTP and sends it to System Operator for matching with saved OTP.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.
2. The user must have a confirmed email address or phone number.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. External Entity sends a request to recover a password.

Endpoint URL: POST /password/recovery

Parameter:

```
{
    "login": "string"
}
```

2. System Operator generates an OTP, saves it, and sends it to the User via email or SMS.

3. System Operator returns a message on how to get OTP.

Response:

```
{
  "status": "ok",
  "message": "string",
  "action": "EMAIL_SENT"
}
```

4. External Entity acting as a User obtains OTP from email or SMS and sends a request to Endpoint “Confirmation of password restoring using one time password” of System Operator for matching OTP with the saved one.

Endpoint URL: POST /password/recovery/confirm

Parameter:

```
{
  "login": "string",
  "otp": "string",
  "newUserPassword": "string"
}
```

5. System Operator returns a confirmation message. (See Result example below)

Optional Flow with Web Browser UI

1. User sends a request to recover a password.

Endpoint URL: POST /password/recovery

Parameter:

```
{
  "login": "string"
}
```

2. System Operator generates an OTP, saves it, and sends it to the User via email or SMS.

3. System Operator returns a message on how to get OTP.

Response:

```
{
  "status": "ok",
  "message": "string",
  "action": "EMAIL_SENT"
}
```

4. User acting obtains OTP from email or SMS and sends a request to Endpoint “Confirmation of password restoring using one-time password” of System Operator for matching OTP with the saved one.

Endpoint URL: POST /password/recovery/confirm

Parameter:

```
{
    "login": "string",
    "otp": "string",
    "newUserPassword": "string"
}
```

5. System Operator returns a confirmation message. (See Result example below)

Post Conditions

The user gets into the system.

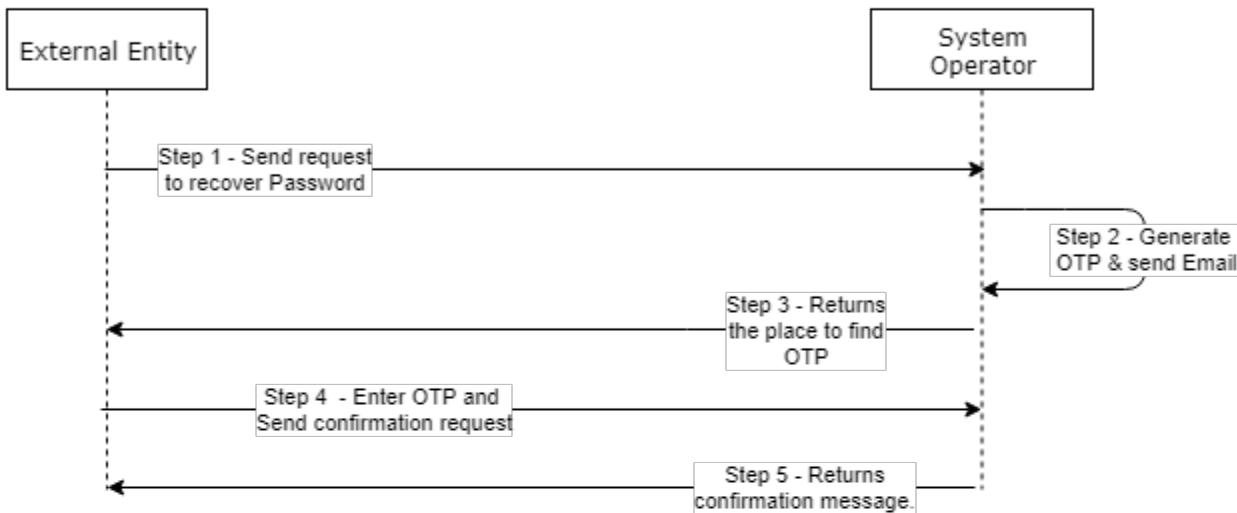
Result example

```
{
    "status": "ok",
    "message": "string",
    "action": "TOKEN_CREATED",
    "authorizationToken": {
        "token": "string",
        "expiresAt": "2018-07-25T12:12:33.818Z"
    },
    "members": [
        {
            "role": "string",
            "user": {
                "id": "string",
                "name": "string"
            },
            "organization": {
                "id": "string",
                "type": "string",
                "name": "string"
            },
            "permissions": [
                "string"
            ],
            "token": {
                "token": "string",
                "expiresAt": "2018-07-25T12:12:33.818Z"
            }
        }
    ]
}
```

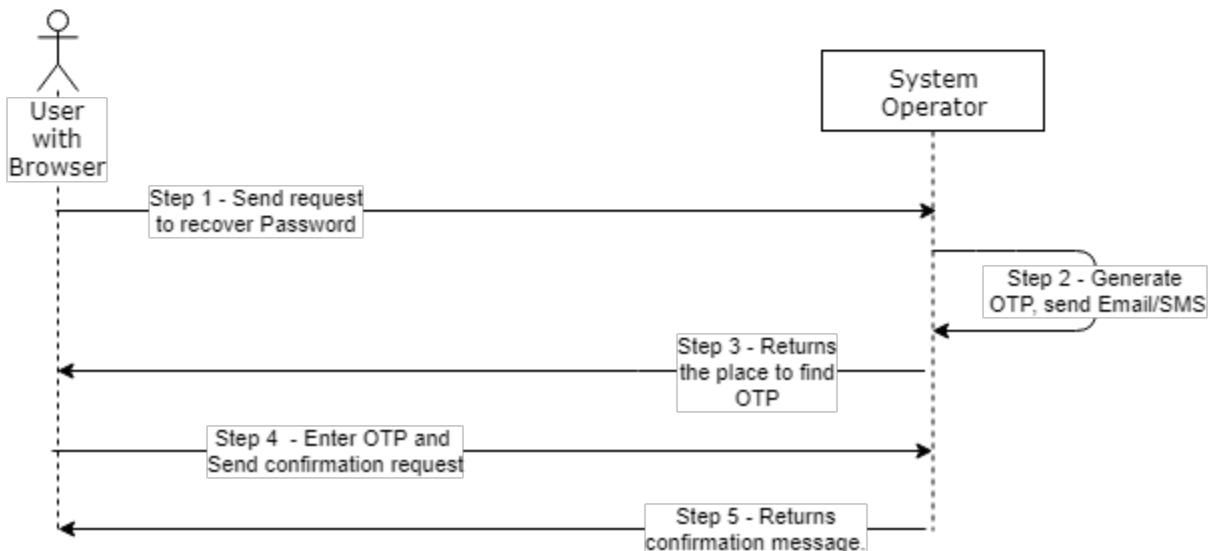
Send request to restore a password scheme

Use case: Send request to restore a password

Basic FLow



Optional Web UI Flow



Profile - Master PIN

Confirmation of ban removal description

Use Case Name

Confirmation of ban removal

Brief Description

A User or External Entity on behalf of a User with role permission Anonymous will go through all steps of “Authentication” Use Case, retrieve OTP from SMS or email, and then send a request to Endpoint “Confirmation of ban removal”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: Anonymous.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
3. A communication service provider - Mobile service company or Email server.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Confirmation of ban removal”.

Endpoint URL: POST /profiles/my/master-pin/confirm-ban-removal

Parameters:

```
{
  "login": "string",
  "otp": "string"
}
```

3. Retrieve OTP sent in SMS or in Email at the time an admin user unbanned this user.
4. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Confirmation of ban removal”.

Endpoint URL: POST /profiles/my/master-pin/confirm-ban-removal

Parameters:

```
{
  "login": "string",
  "otp": "string"
}
```

3. Retrieve OTP sent in SMS or in Email at the time an admin user unbanned this user.
4. System Operator returns result information to User (See Result example below).

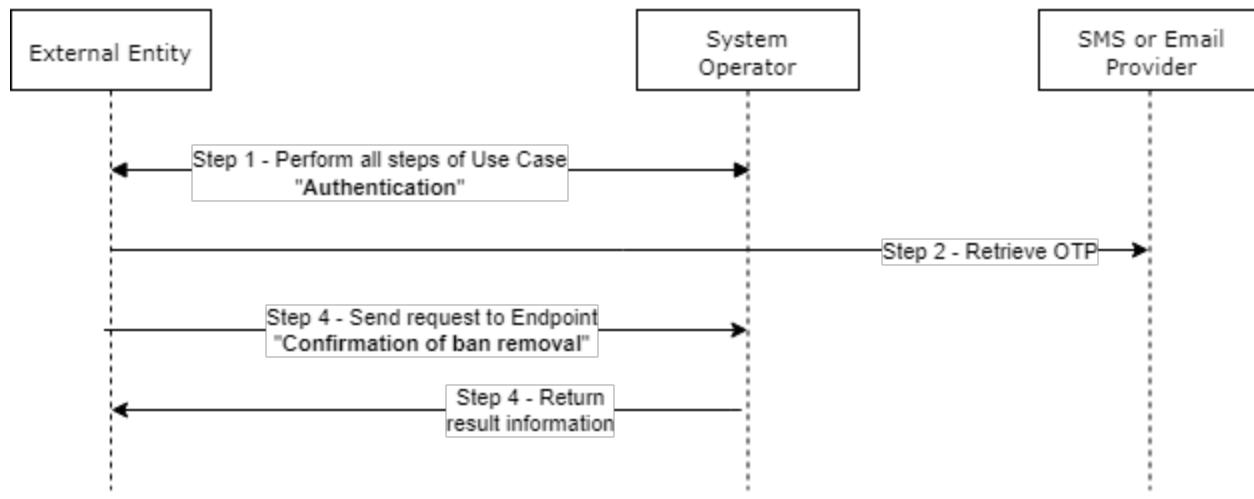
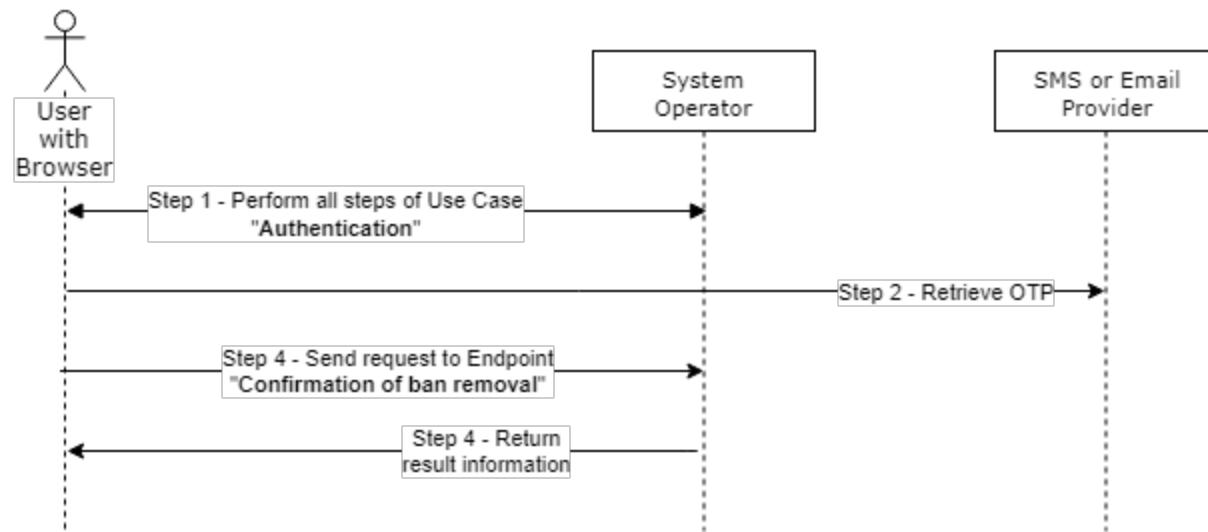
Post Conditions

Ban removal is confirmed.

Result example

```
{  
    "status": "ok",  
    "message": "string"  
}
```

Confirmation of ban removal scheme

Use case: Confirmation of ban removal**Basic FLow****Optional Web UI Flow**

Get master PIN belonging to current user description

Use Case Name

Get master PIN belonging to the current user

Note: On the contrary of the name, the Use Case does not provide actual PIN value. It only verifies PIN existence. The issue was added to JIRA: Issue SDKFIN-1419 - "Get master PIN belonging to current user" in the misleading name for this Endpoint.

Brief Description

A User or External Entity on behalf of a User with role permission MASTER_PIN_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get master PIN belonging to current user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MASTER_PIN_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get master PIN belonging to current user”.

Endpoint URL: GET /profiles/my/master-pin

Body parameters: Security TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get master PIN belonging to current user”.

Endpoint URL: GET /profiles/my/master-pin

Body parameters: Security TOKEN

3. System Operator returns result information to User (See Result example below).

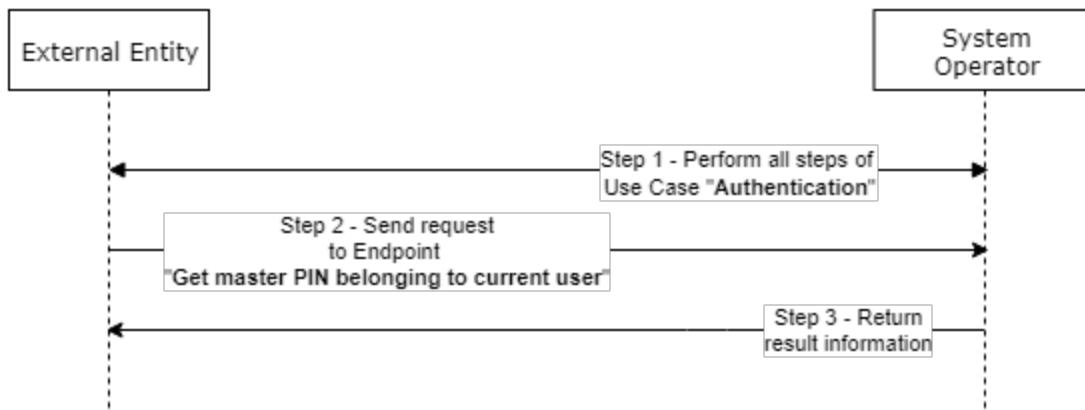
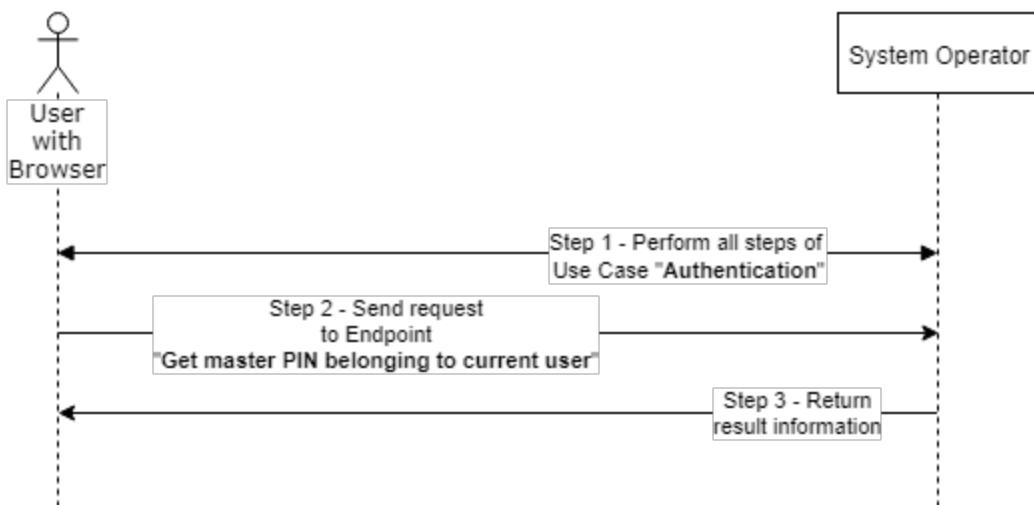
Post Conditions

Requested information is available

Result example

```
{  
  "pin": {  
    "createdAt": "2018-08-07T14:16:19.170Z",  
    "updatedAt": "2018-08-07T14:16:19.170Z",  
    "active": false  
  },  
  "status": "ok",  
  "message": "string"  
}
```

Get master PIN belonging to current user scheme

Use case: Get master PIN belonging to current user**Basic FFlow****Optional Web UI Flow**

Restore an existing master PIN by generating a new one description

Use Case Name

Restore an existing master PIN by generating a new one

Brief Description

A User or External Entity on behalf of an Anonymous User will send a request to recover a PIN. In the request, there will be a contact information in the form of the phone number or the email address. This contact information is verified and if correct, a new PIN will be generated, permanently saved, and sent to a User's selected channel.

Actors

1. External Entity that will contact System Operator as Anonymous User.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.
3. Email or SMS service provider.

Preconditions

1. The user must have a System Operator profile with valid and verified contact information.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. External Entity sends a request to Endpoint "Restore an existing master PIN by generating a new one".

Endpoint URL: POST /profiles/my/master-pin/restore

Parameters:

```
{
  "contact": "string"
}
```

"contact": "string" - Contact where the generated PIN will be sent to

2. System Operator generates a new PIN, saves it, and sends it via email or SMS.
3. System Operator returns result information to External Entity. (See Result example below)
4. External Entity retrieves new PIN from email or receives it via SMS.

Optional Flow with Web Browser UI

1. A user sends a request to Endpoint "Restore an existing master PIN by generating a new one".

Endpoint URL: POST /profiles/my/master-pin/restore

Parameters:

```
{
  "contact": "string"
}
```

"contact": "string" - Contact where the generated PIN will be sent to

2. System Operator generates a new PIN, saves it, and sends it via email or SMS.

3. System Operator returns result information to External Entity. (See Result example below)

4. A user retrieves new PIN from email or receives it via SMS.

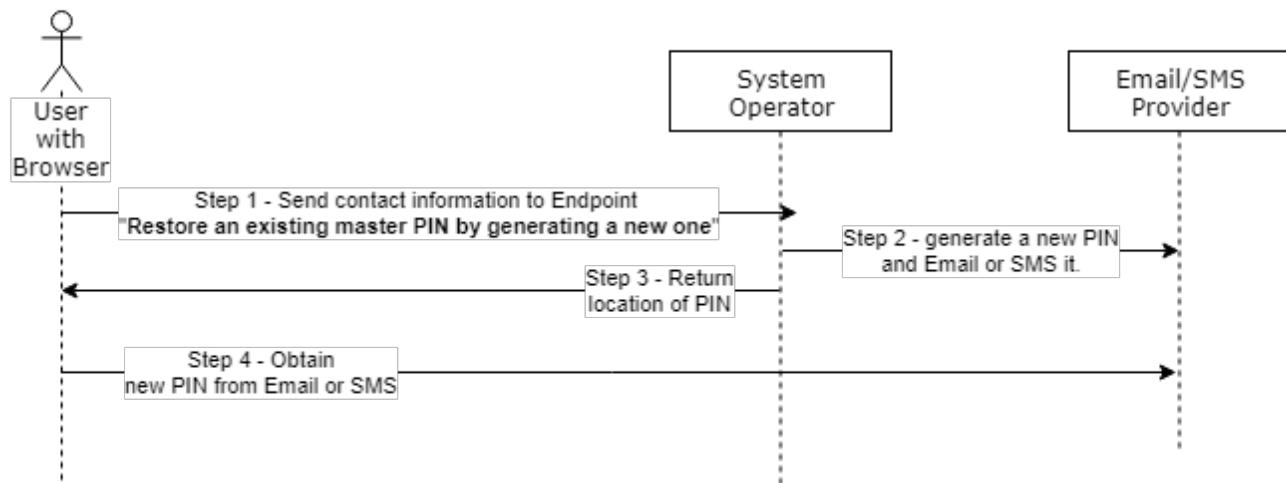
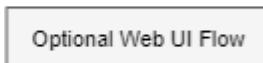
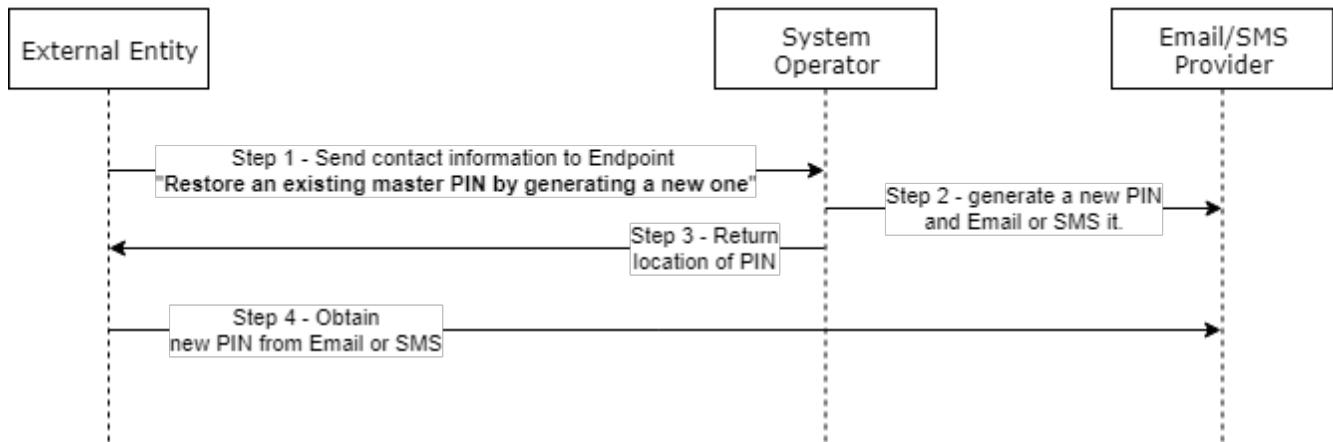
Post Conditions

As a result of a User again has access to the system.

Result example

```
{  
  "pin": {  
    "createdAt": "2018-08-07T14:16:19.184Z",  
    "updatedAt": "2018-08-07T14:16:19.184Z",  
    "active": false  
  },  
  "status": "ok",  
  "message": "string"  
}
```

Restore an existing master PIN by generating a new one scheme

Use case: Restore an existing master PIN by generating a new one

Set up a new or replace an existing master PIN description

Use Case Name

Set up a new or replace an existing master PIN

Brief Description

A User or External Entity on behalf of a User with role permission MASTER_PIN_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Set up a new or replace an existing master PIN”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MASTER_PIN_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get master PIN belonging to current user”.
2. External Entity sends a request to Endpoint “Set up a new or replace an existing master PIN”.

Endpoint URL: PUT /profiles/my/master-pin

Parameters:

```
{
  "oldPin": "string",
  "newPin": "string",
  "active": false
}
```

“oldPin”: “string”, - null without quotes if for the first time.

“newPin”: “string”, - make your own PIN. Ex. 1234

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get master PIN belonging to current user”.
2. A user sends a request to Endpoint “Set up a new or replace an existing master PIN”.

Endpoint URL: PUT /profiles/my/master-pin

Parameters:

```
{
  "oldPin": "string",
  "newPin": "string",
  "active": false
}
```

“oldPin”: “string”, - null without quotes if for the first time.

"newPin": "string", - make your own PIN. Ex. 1234

3. System Operator returns result information to User (See Result example below).

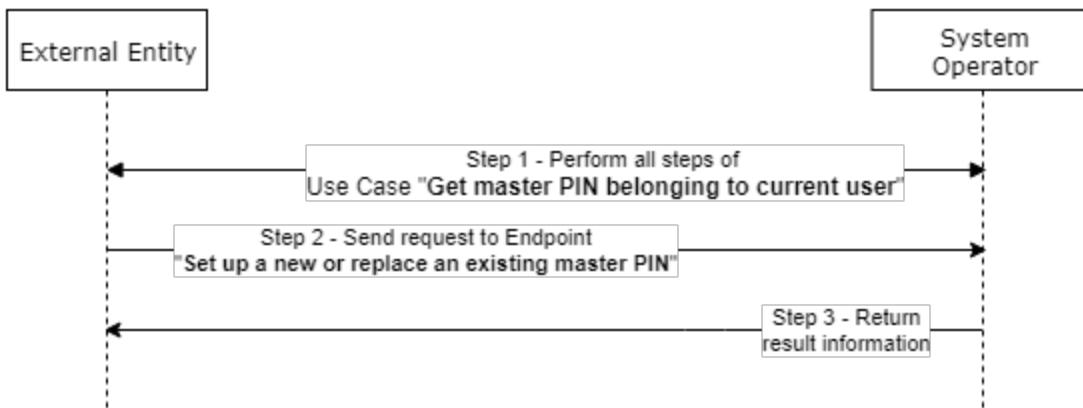
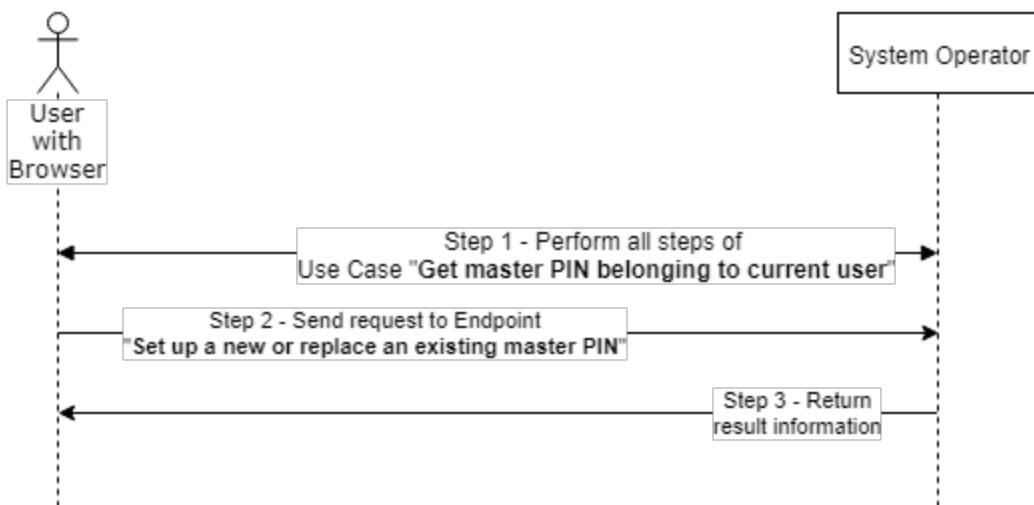
Post Conditions

The new PIN is available.

Result example

```
{  
  "pin": {  
    "createdAt": "2018-08-07T14:16:19.175Z",  
    "updatedAt": "2018-08-07T14:16:19.175Z",  
    "active": false  
  },  
  "status": "ok",  
  "message": "string"  
}
```

Set up a new or replace an existing master PIN scheme

Use case: Set up a new or replace an existing master PIN**Basic FFlow****Optional Web UI Flow**

Bank accounts management

Approve bank account as compliance description

Use Case Name

Approve bank account as compliance

Brief Description

Area of Compliance functionality, which allows to view, approve or reject Bank Account. In this case the Bank Account is the account that a wallet owner defines as the account associated with the Wallet. Then, a representative of System Operator reviews information about the Bank and Account and approves or rejects it. This can only be done by the logged in user. This Use Case details the steps of a positive scenario where the Bank Account is successfully approved.

Actors

1. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
2. External Entity that can interact with System Operator API acting as a registered System Operator user as Accountant.

Preconditions

1. The owner of the Wallet added a Bank Account to the profile.
2. Accountant received a notification about a new Bank Account in “Pending approval” status.
3. The accountant should complete Authentication Use Case.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to EE.

1. Perform all steps of Use Case “View bank accounts as compliance”.
2. After reviewing the Bank Account information External Entity sends a request with approval instruction.

Endpoint URL: POST /bank-accounts/{bankAccountId}/approve

3. System Operator platform sends a verification about operation result - APPROVED.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View bank accounts as compliance”.
2. After reviewing Bank Account details Compliance Specialist sends to System Operator a request to approve this Account.
3. System Operator returns a confirmation page with status APPROVED.

Post Conditions

Bank Account is ready to be used by User.

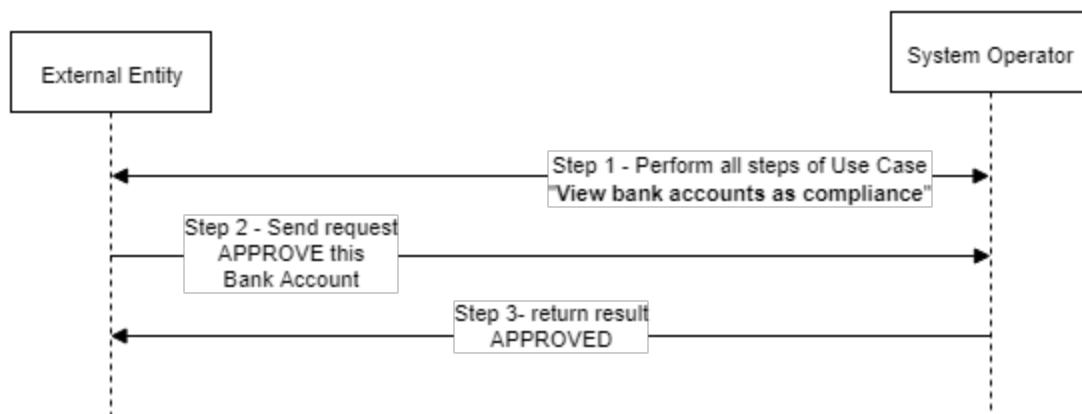
Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "createdAt": "2018-06-22T16:53:54.733Z",  
            "updatedAt": "2018-06-22T16:53:54.733Z",  
            "status": "APPROVED",  
            "details": {  
                "fullName": "string",  
                "account": "string",  
                "iban": "string",  
                "bic": "string",  
                "swift": "string",  
                "name": "string",  
                "address": "string"  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

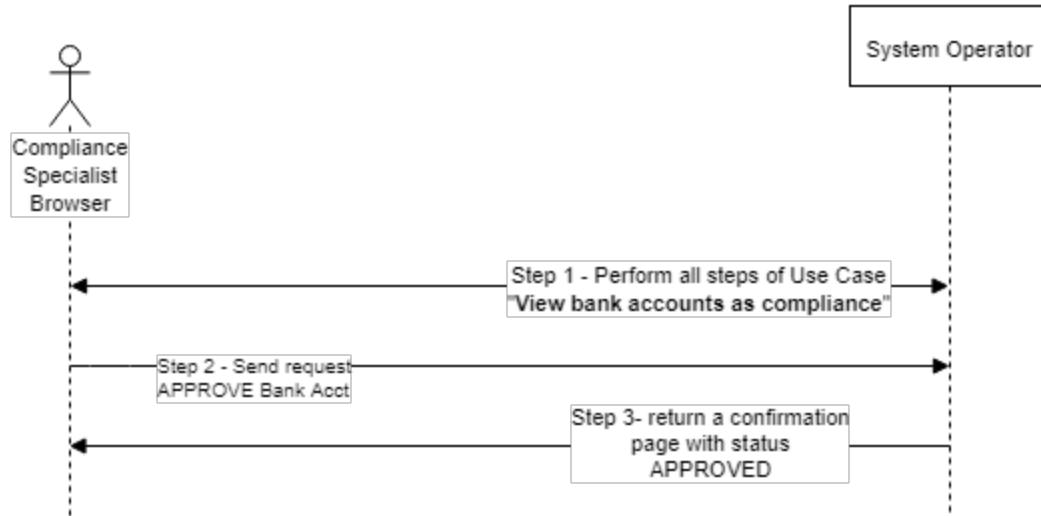
Approve bank account as compliance scheme

Use case: Approve bank account as compliance

Basic FLow



Optional Web UI Flow



Reject bank account as compliance description

Use Case Name

Reject bank account as compliance

Brief Description

Area of Compliance functionality, which allows to view, approve or reject Bank Account. In this case the Bank Account is the account that a wallet owner defines as the account associated with the Wallet. Then, a representative of System Operator with roles: BANK_ACCOUNT_MANAGER reviews information about the Bank and Account and approves or rejects it. This can only be done by the logged in user. This Use Case details the steps of a negative scenario where the Bank Account is rejected.

Actors

1. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
2. External Entity that can interact with System Operator API acting as a registered System Operator with assigned roles:
BANK_ACCOUNT_MANAGER user as Accountant.

Preconditions

1. The owner of the Wallet added a Bank Account to the profile.
2. Accountant received a notification about a new Bank Account in “Pending approval” status.
3. The accountant should complete Authentication Use Case.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to EE.

1. Perform all steps of Use Case “View bank accounts as compliance”.
2. After reviewing the Bank Account information External Entity sends a request with approval instruction.

Endpoint URL: POST /bank-accounts/{bankAccountId}/reject

3. System Operator platform sends a verification about operation result - REJECTED.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View bank accounts as compliance”.
2. After reviewing Bank Account details Compliance Specialist sends to System Operator a request to reject this Account.
3. System Operator returns a confirmation page with status REJECTED.

Post Conditions

Bank Account is NOT ready to be used by User.

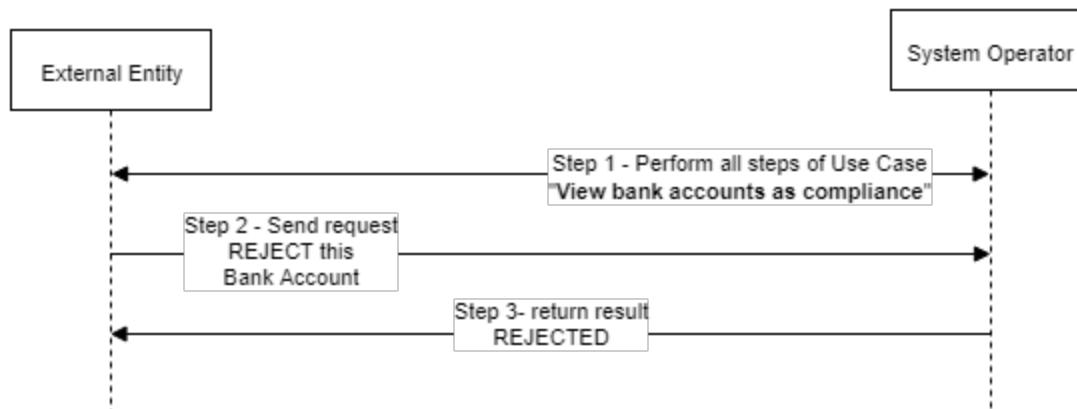
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-06-22T16:53:54.733Z",  
      "updatedAt": "2018-06-22T16:53:54.733Z",  
      "status": "REJECTED",  
      "details": {  
        "fullName": "string",  
        "account": "string",  
        "iban": "string",  
        "bic": "string",  
        "swift": "string",  
        "name": "string",  
        "address": "string"  
      }  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

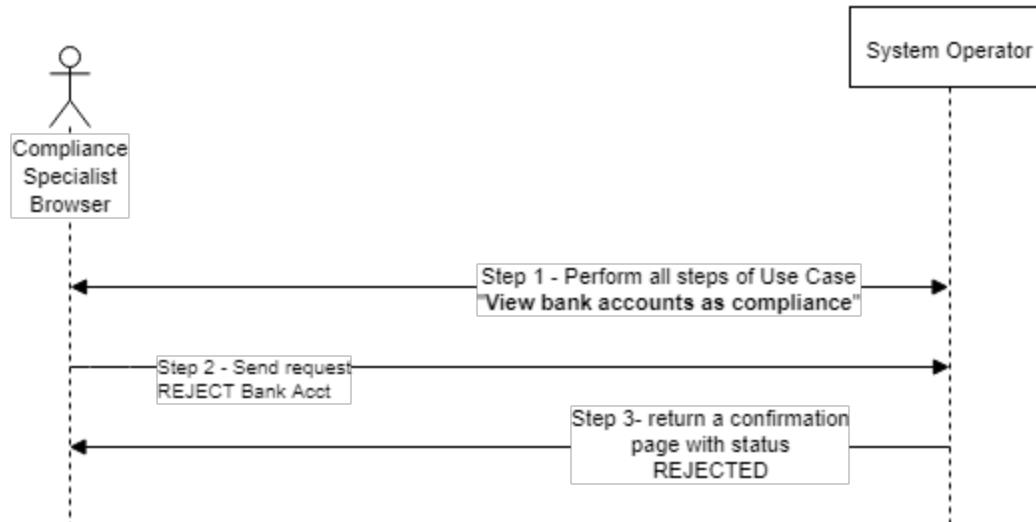
Reject bank account as compliance scheme

Use case: Reject bank account as compliance

Basic FFlow



Optional Web UI Flow



View bank accounts as compliance description

Use Case Name

View bank accounts as compliance

Brief Description

Area of Compliance functionality, which allows to view, approve or reject Bank Account. In this Use Case, the Bank Account is the account that a Wallet owner defines as the account associated with the Wallet. Then, a representative of System Operator with the role BANK_ACCOUNT_MANAGER reviews information about the Bank and Account and approves or rejects it. This can only be done by the logged in user. This Use Case details only a situation when Accountant wants to View the information without any further action.

Actors

1. System Operator using “SDK.Finance” software and exposing portfolio of financial APIs.
2. External Entity that can interact with System Operator API acting as a registered System Operator User as Accountant with role BANK_ACCOUNT_MANAGER.

Preconditions

1. The Owner of the Coin added a Bank Account to the profile.
2. Accountant received a notification about a new Bank Account in “Pending approval” status.
3. The accountant should complete Authentication Use Case.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to System Operator to get Bank Account details.

Endpoint URL: /bank-accounts/view

Parameters:

```
{
    "pageNumber": 0,
    "pageSize": 0,
    "filter": {
        "statuses": [
            "PENDING"
        ],
        "account": "string",
        "iban": "string",
        "bic": "string",
        "swift": "string",
        "bankName": "string"
    },
    "sort": {
        "status": "asc",
        "bankName": "asc"
    }
}
```

3. System Operator returns all details for requested Bank Account.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. Compliance Specialist sends a request to System Operator to see Account details.

Endpoint URL: /bank-accounts/view

Parameters:

```
{  
    "pageNumber": 0,  
    "pageSize": 0,  
    "filter": {  
        "statuses": [  
            "PENDING"  
        ],  
        "account": "string",  
        "iban": "string",  
        "bic": "string",  
        "swift": "string",  
        "bankName": "string"  
    },  
    "sort": {  
        "status": "asc",  
        "bankName": "asc"  
    }  
}
```

3. System Operator retrieves Account details and sends to Compliance Specialist.

Post Conditions

Compliance Specialist has all information need to work with User's Bank Account.

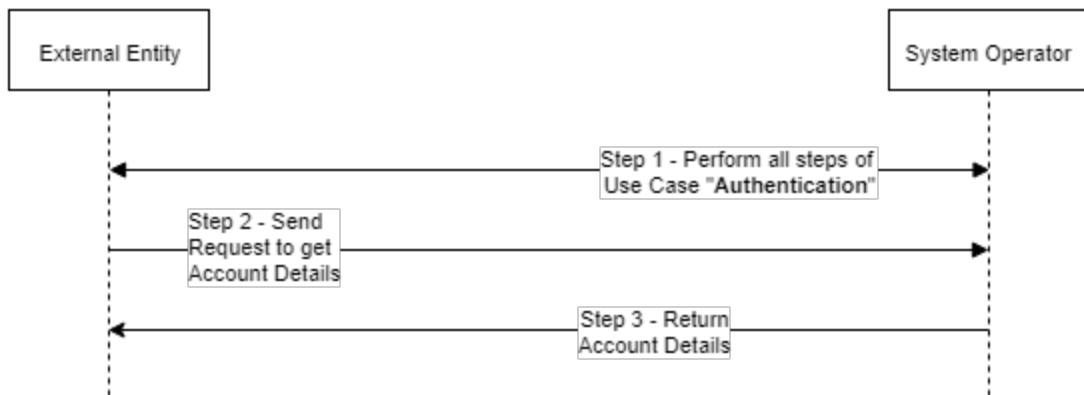
Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "createdAt": "2018-06-22T16:53:54.733Z",  
            "updatedAt": "2018-06-22T16:53:54.733Z",  
            "status": "PENDING",  
            "details": {  
                "fullName": "string",  
                "account": "string",  
                "iban": "string",  
                "bic": "string",  
                "swift": "string",  
                "name": "string",  
                "address": "string"  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

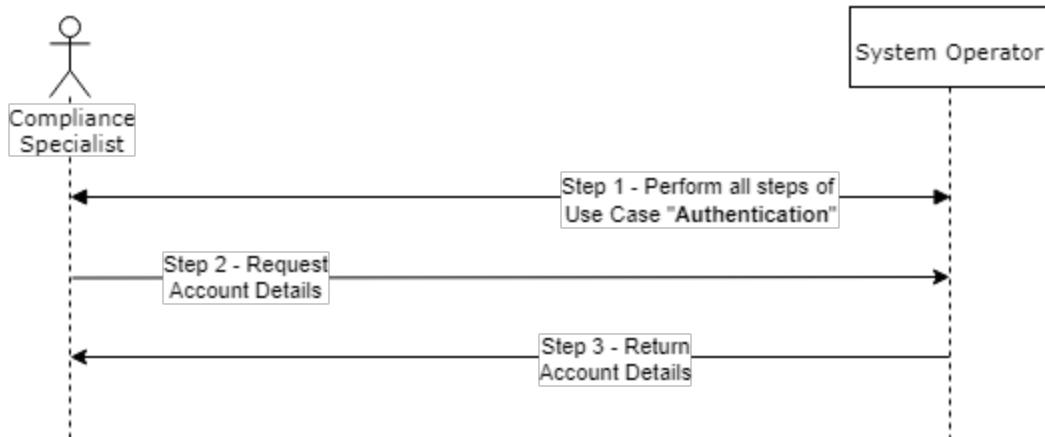
[View bank accounts as compliance scheme](#)

Use case: View bank accounts as compliance

Basic FLow



Optional Web UI Flow



Validation for coin

Validate a coin description

Use Case Name

Validate a coin

Brief Description

External Entity or User gets a list of all available Coins, then sends a request to check the Coin status.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to get a list of all Coins.

Endpoint URL: GET /coins

Parameter: Security TOKEN.

3. System Operator returns a List of User Coins to External Entity.
4. External Entity selects a Coin and sends a request to check Coin’s status.

Endpoint URL: POST /coins/validate

Parameters:

```
{
  "serial": "string"
}
```

5. System Operator returns Coin status information (see below in Result Example section)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The user sends a request to get a list of all Coins.

Endpoint URL: GET /coins

Parameter: Security TOKEN.

3. System Operator returns a List of User Coins.
4. The user selects a Coin and sends a request to check Coin’s status.

Endpoint URL: POST /coins/validate

Parameters:

```
{
  "serial": "string"
}
```

5. System Operator returns Coin status information (see below in Result Example section)

Post Conditions

External Entity or User can verify Coin status.

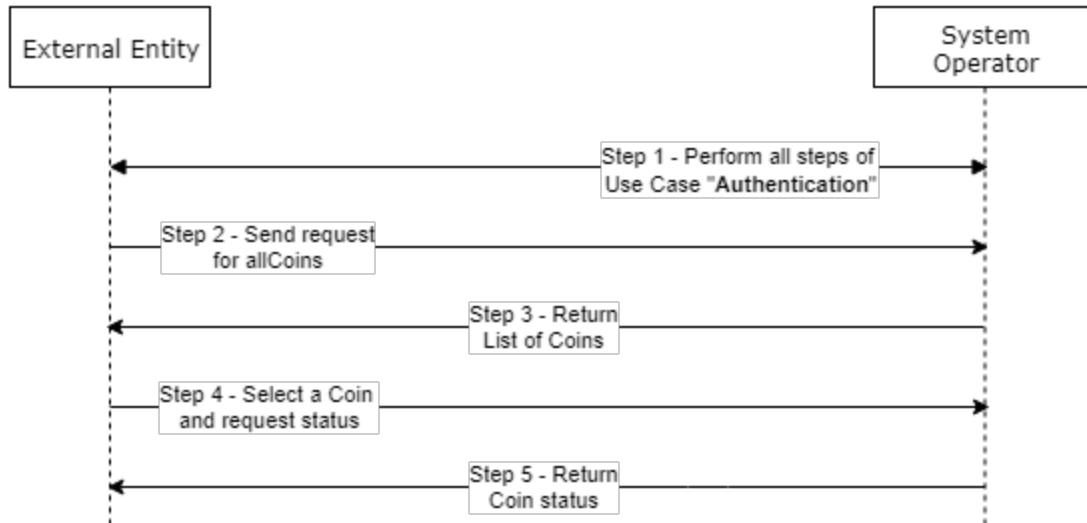
Result example

```
{  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    },  
    "owner": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

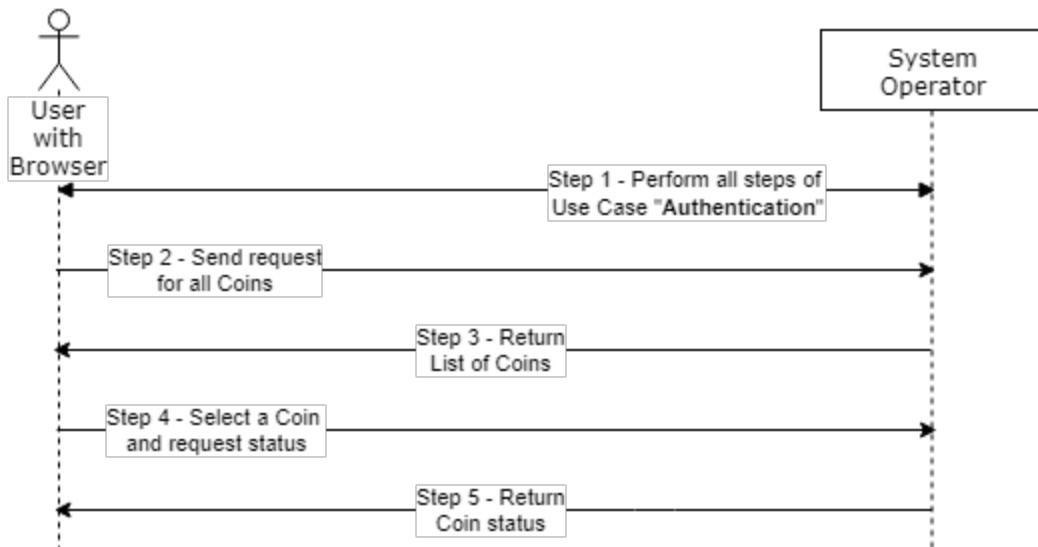
Validate a coin scheme

Use case: Validate a coin

Basic FLow



Optional Web UI Flow



Validation for user contact

Validate a contact description

Use Case Name

Validate a contact

Brief Description

Use Case could be executed by a User with a Role which has a privilege to view User information.

The purpose of this Use Case is to find a contact for a Use who is known by name.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to get the user’s contact information.

Endpoint URL: POST /contacts/validate

Parameter:

```
{
  "value": "string"
}
```

3. System Operator returns requested information (see Result Example section below).

System Operator returns a List of Issuers to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The user sends a request to get user’s contact information.

Endpoint URL: POST /contacts/validate

Parameter:

```
{
  "value": "string"
}
```

3. System Operator returns requested information (see Result Example section below).

Post Conditions

Contact information is available.

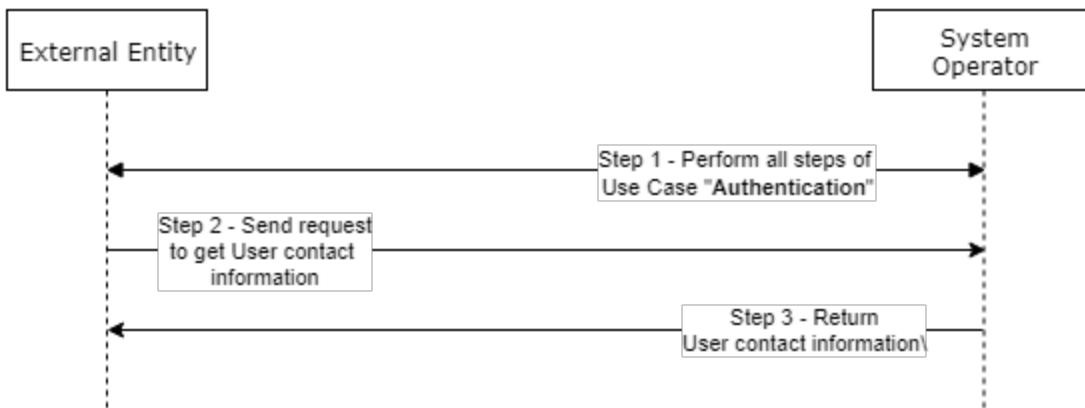
Result example

```
{  
    "user": {  
        "id": "string",  
        "name": "string"  
    },  
    "contactType": "phone",  
    "status": "ok",  
    "message": "string"  
}
```

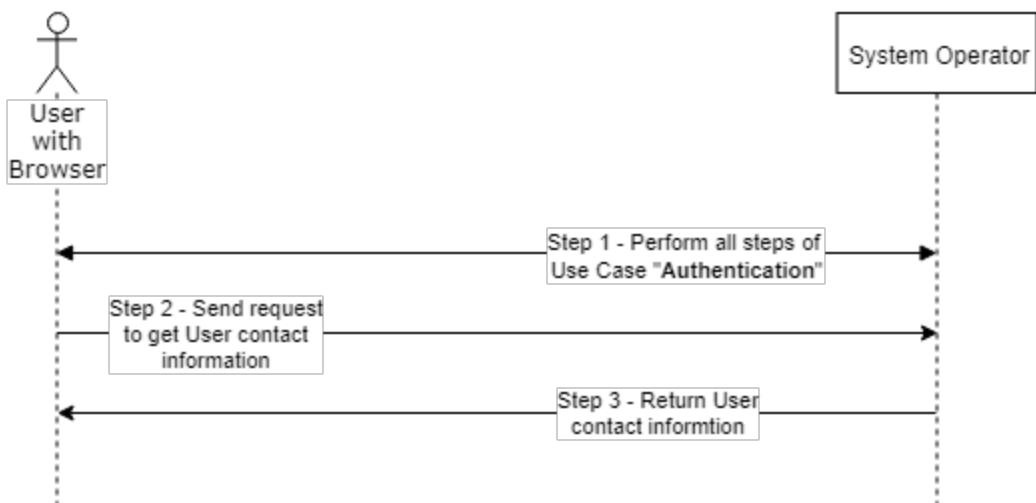
Validate a contact scheme

Use case: Validate a contact

Basic FLow



Optional Web UI Flow



User note API provider

Create note about user description

Use Case Name

Create note about user

Brief Description

A User or External Entity on behalf of a User with role permission "USER_NOTE_MANAGER" will go through all steps of "Get users" Use Case (to obtain user ID), and then send a request to Endpoint "Create note about user".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "USER_NOTE_MANAGER", e.g. head of compliance, administrator, compliance manager, or compliance specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get users".
2. External Entity sends a request to Endpoint "Create note about user".

Endpoint URL: POST /users/note

```
Parameters: {
  "userId": "string",
  "note": "string"
}
```

1. System Operator returns verification note was created to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get users".
2. A user sends a request to Endpoint "Create note about user".

Endpoint URL: POST /users/note

```
Parameters: {
  "userId": "string",
  "note": "string"
}
```

1. System Operator returns verification note was created to User (See Result example below).

Post Conditions

User ID exists.

Result example

```
{
  "status": "ok",
  "message": "string",
  "noteDto": {
    "id": "string",
    "note": "string",
    "createdAt": "2018-08-22T09:12:26.706Z"
  }
}
```

Create note about user scheme

Get all user notes description

Use Case Name

Get all user notes

Brief Description

A User or External Entity on behalf of a User with role permission "USER_NOTE_MANAGER"

will go through all steps of "Get users" Use Case (to obtain user ID), and then send a request to Endpoint "Get all user notes".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "USER_NOTE_MANAGER", e.g. head of compliance, administrator, compliance manager, or compliance specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Get all user notes”.

Endpoint URL: GET /users/note/{userId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns all notes for a user to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Get all user notes”.

Endpoint URL: GET /users/note/{userId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns all notes for a user to User (See Result example below).

Post Conditions

User ID exists.

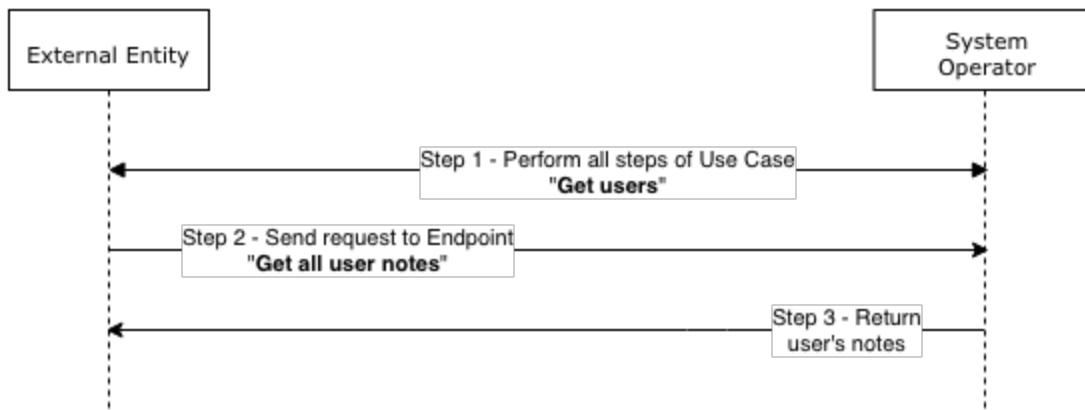
Result example

```
{
  "notes": [
    {
      "id": "string",
      "note": "string",
      "createdAt": "2018-08-22T09:12:26.708Z"
    }
  ]
}
```

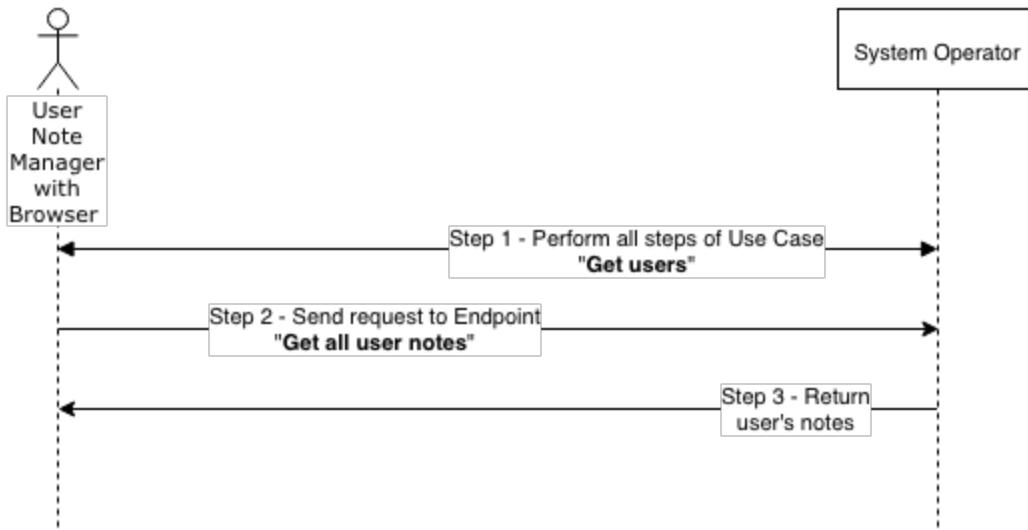
Get all user notes scheme

Use case: Get all user notes

Basic FFlow



Optional Web UI Flow



Investment via cashdesk

Accept investment request 1 description

Use Case Name

Accept investment request

Brief Description

A User or External Entity on behalf of a User with role permission "PROVIDER_INVESTMENT_VALIDATOR" will go through all steps of "View cash investment details" Use Case (to obtain business request identifier), and then send a request to Endpoint "Accept investment request".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "PROVIDER_INVESTMENT_VALIDATOR", e.g. CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View cash investment details".
2. External Entity sends a request to Endpoint "Accept investment request".

Endpoint URL: POST /investments/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

1. System Operator returns details of accepted request to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View cash investment details".
2. A user sends a request to Endpoint "Accept investment request".

Endpoint URL: POST /investments/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

1. System Operator returns details of accepted request to User (See Result example below).

Post Conditions

Business request identifier is available.

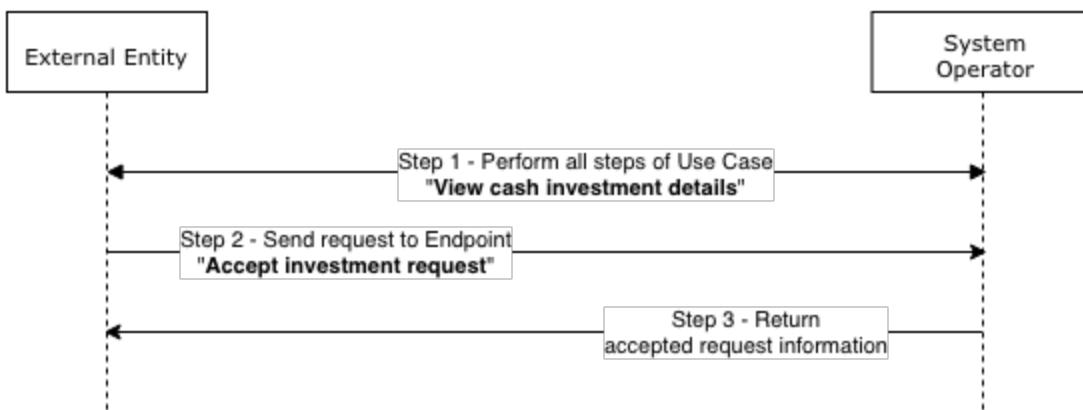
Result example

```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-09-17T11:07:13.989Z",  
    "updatedAt": "2018-09-17T11:07:13.989Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        }  
      }  
    ]  
  }  
}
```

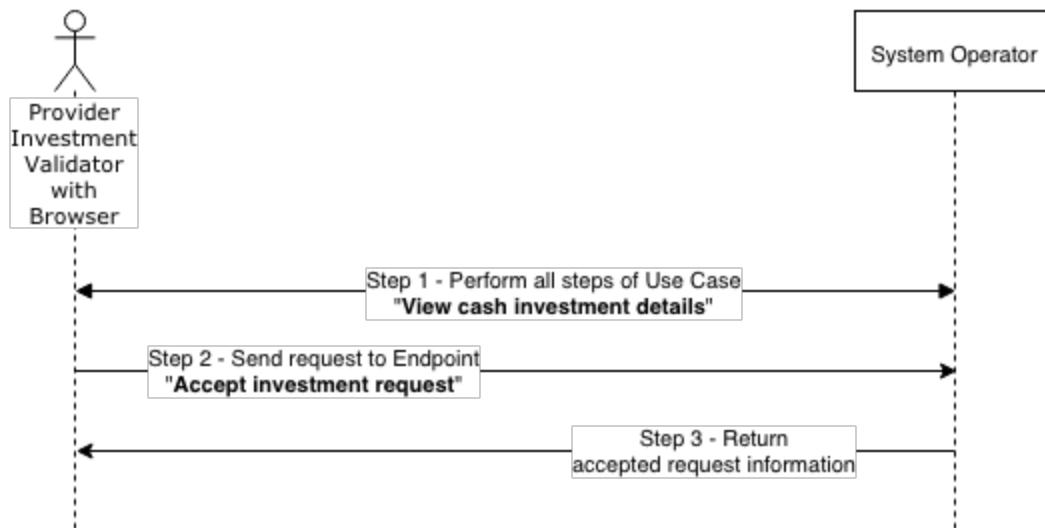
```
 },
 "amount": 0,
 "performedAt": "2018-09-17T11:07:13.990Z",
 "issuer": {
   "id": "string",
   "sn": "string",
   "currency": "string"
 },
 },
 ],
 "children": [
   {
     "errorMessage": "string"
   },
   "status": "ok",
   "message": "string"
 }
}
Accept investment request 1 scheme
```

Use case: Accept investment request

Basic FFlow



Optional Web UI Flow



Top up of authorized capital with cash description

Use Case Name

Top up of authorized capital with cash

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_INVESTMENT_EXECUTOR" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Top up of authorized capital with cash".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_INVESTMENT_EXECUTOR", eg. accountant or CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to User (See Result example below).

Post Conditions

Serial and authorized person are available.

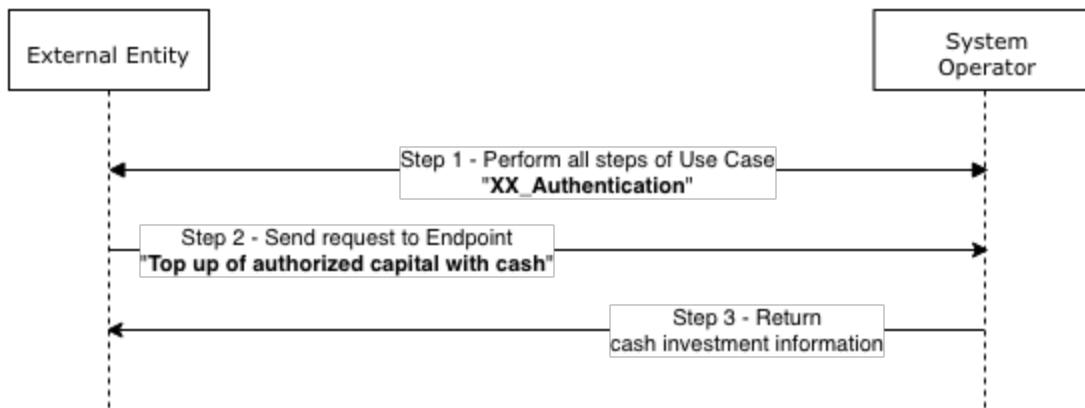
Result example

```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-09-17T11:07:13.977Z",  
    "updatedAt": "2018-09-17T11:07:13.977Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "amount": 0,  
        "performedAt": "2018-09-17T11:07:13.977Z"  
      }  
    ]  
  }  
}
```

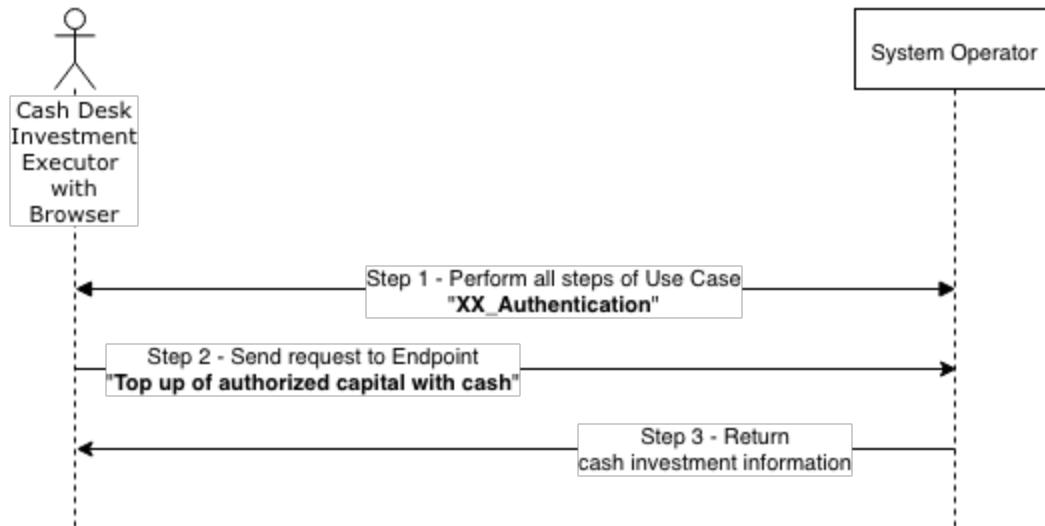
```
"issuer": {  
    "id": "string",  
    "sn": "string",  
    "currency": "string"  
},  
}  
],  
"children": [  
    {}  
],  
"errorMessage": "string"  
,  
"status": "ok",  
"message": "string"  
}  
Top up of authorized capital with cash scheme
```

Use case: Top up of authorized capital with cash

Basic FFlow



Optional Web UI Flow



View cash investment details 1 description

Use Case Name

Top up of authorized capital with cash

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_INVESTMENT_EXECUTOR" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Top up of authorized capital with cash".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_INVESTMENT_EXECUTOR", eg. accountant or CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to User (See Result example below).

Post Conditions

Serial and authorized person are available.

Result example

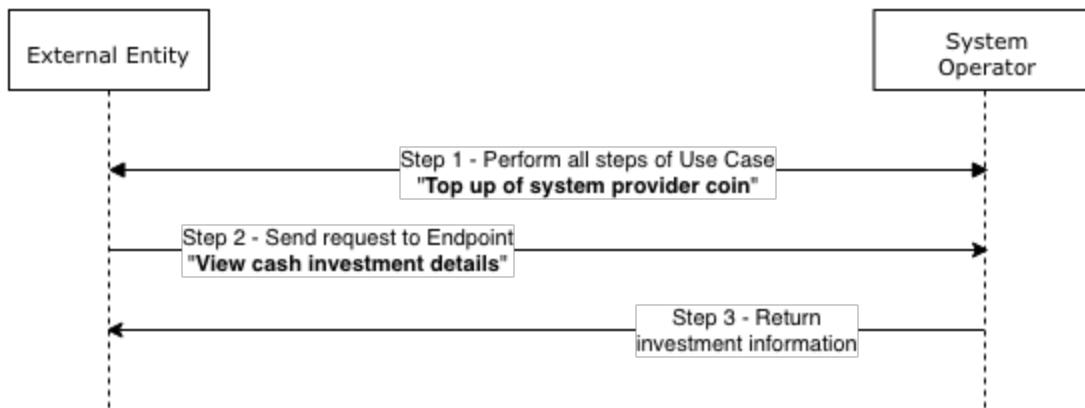
```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-09-17T11:07:13.977Z",  
    "updatedAt": "2018-09-17T11:07:13.977Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "amount": 0,  
        "performedAt": "2018-09-17T11:07:13.977Z",  
        "status": "success",  
        "error": null  
      }  
    ]  
  }  
}
```

```
"issuer": {  
    "id": "string",  
    "sn": "string",  
    "currency": "string"  
},  
}  
],  
"children": [  
    {}  
],  
"errorMessage": "string"  
,  
"status": "ok",  
"message": "string"  
}
```

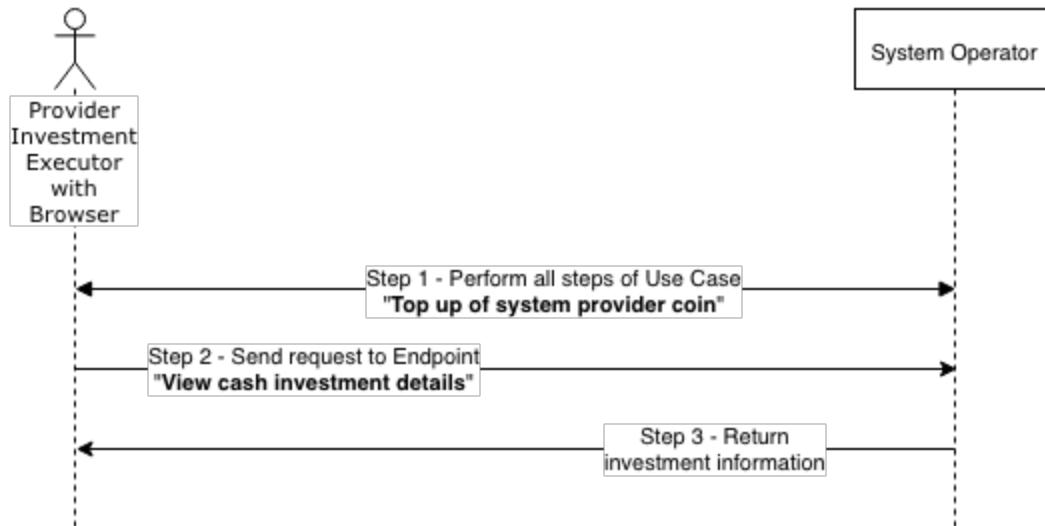
View cash investment details 1 scheme

Use case: View cash investment details

Basic FFlow



Optional Web UI Flow



View cash investment details2 description

Use Case Name

Top up of authorized capital with cash

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_INVESTMENT_EXECUTOR" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Top up of authorized capital with cash".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_INVESTMENT_EXECUTOR", eg. accountant or CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to User (See Result example below).

Post Conditions

Serial and authorized person are available.

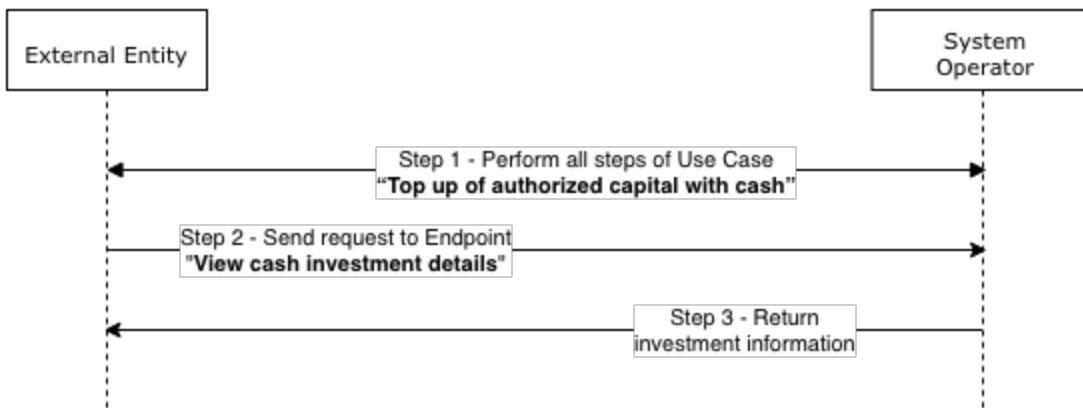
Result example

```
"performedAt": "2018-09-17T11:07:13.977Z",
"issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
},
],
"children": [
    {}
],
"errorMessage": "string"
},
"status": "ok",
"message": "string"
}
```

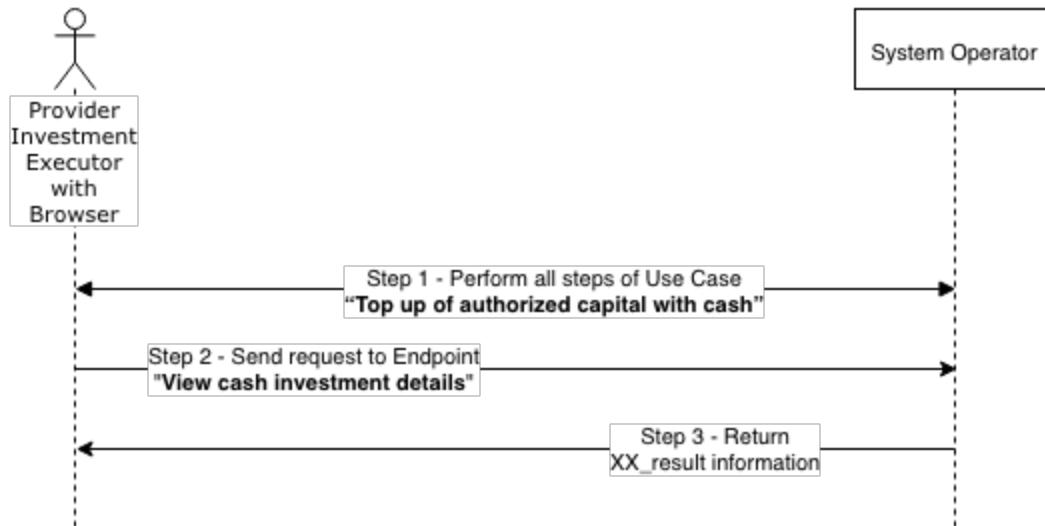
View cash investment details2 scheme

Use case: View cash investment details

Basic FLow



Optional Web UI Flow



Input to cash desk operations

Accept input request description

Use Case Name

Accept input request

Brief Description

A User or External Entity on behalf of a User with role permission "CASHIER" will go through all steps of "View cash input details" Use Case, and then send a request to Endpoint "Accept input request". This request requires a requestIdentifier parameter which is obtained in the "Input collected cash to cash desk" Use Case. The "View cash input details" Use Case provides further details.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASHIER".
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View cash input details".
2. External Entity sends a request to Endpoint "Accept input request".

Endpoint URL: POST /inputs/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

1. System Operator returns request and transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View cash input details".
2. A user sends a request to Endpoint "Accept input request".

Endpoint URL: POST /inputs/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

1. System Operator returns request and transaction information including to User (See Result example below).

Post Conditions

Cash input request is available.

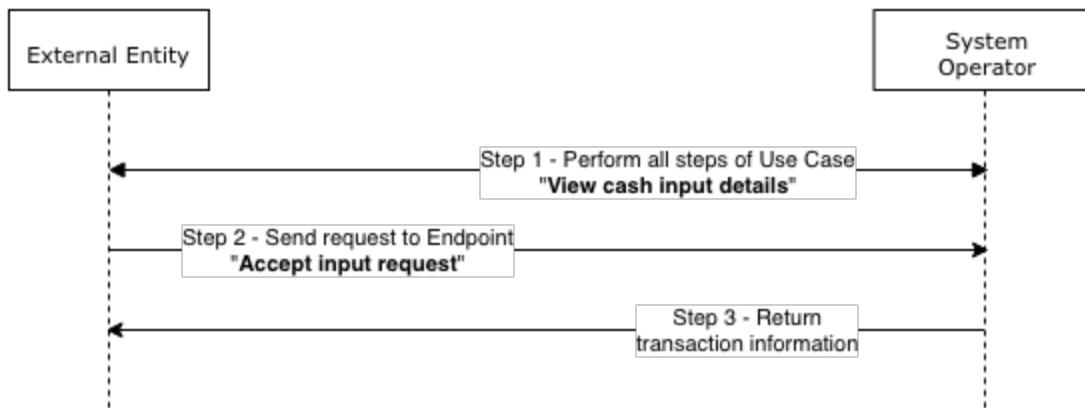
Result example

```
"performedAt": "2018-08-31T14:37:03.018Z",
"issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
},
],
"children": [
    {}
],
"errorMessage": "string"
},
"status": "ok",
"message": "string"
}
```

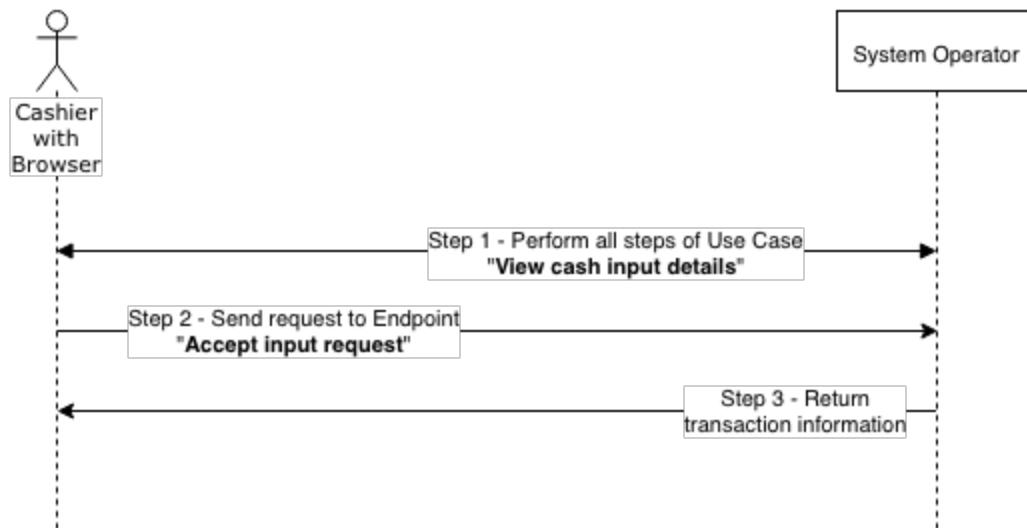
Accept input request scheme

Use case: Accept input request

Basic FLow



Optional Web UI Flow



Input collected cash to cash desk description

Use Case Name

Input collected cash to cash desk

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_INPUT_EXECUTOR" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Input collected cash to cash desk". System returns requestID used in other Endpoints in this Use Case. The input parameters will be known to the "CASH_DESK_INPUT_EXECUTOR".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_INPUT_EXECUTOR". The roles for this user include CFO and accountant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Input collected cash to cash desk".

Endpoint URL: POST /inputs

```
Parameters: {
  "amount": 0,
  "serial": "string",
  "fullName": "string"
}
```

1. System Operator returns request and transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Input collected cash to cash desk".

Endpoint URL: POST /inputs

```
Parameters: {
  "amount": 0,
  "serial": "string",
  "fullName": "string"
}
```

```
}
```

1. System Operator returns request and transaction information to User (See Result example below).

Post Conditions

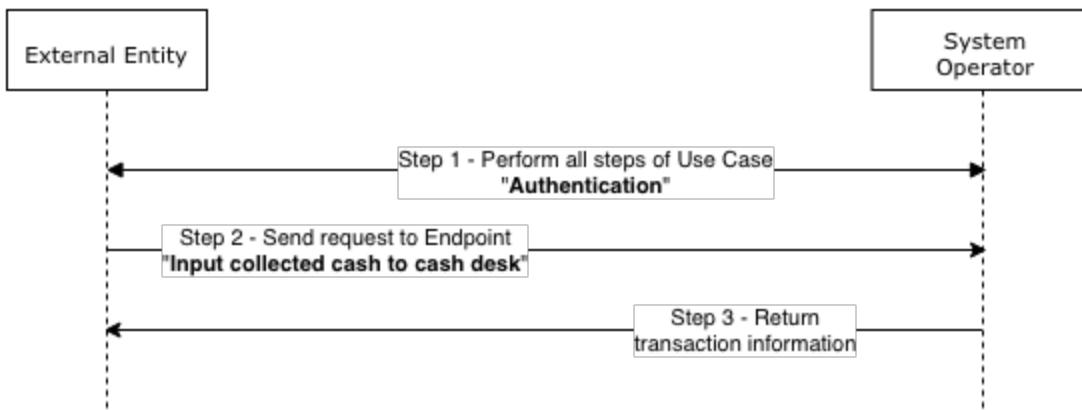
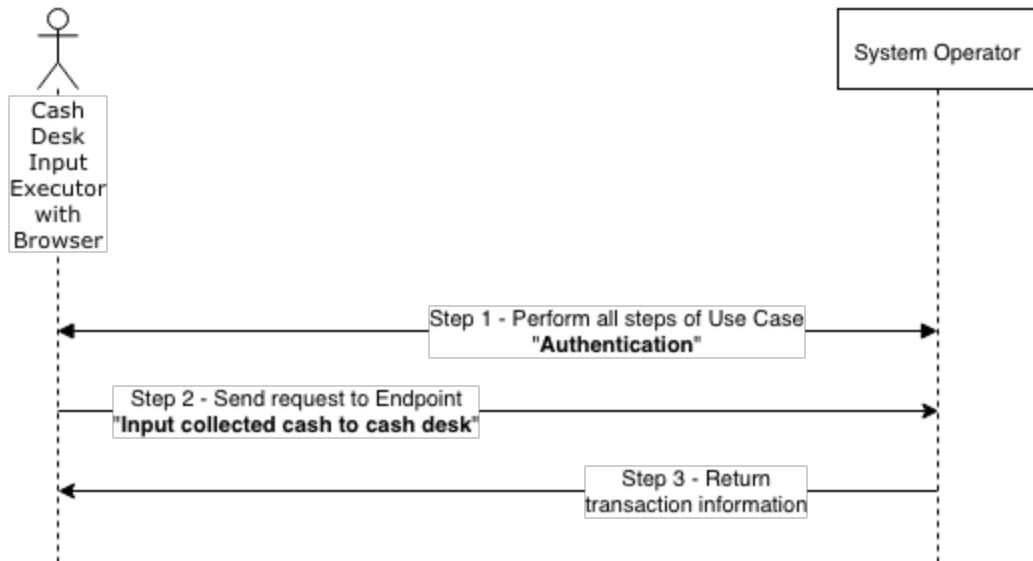
Cash with relevant serial numbers is available.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-08-31T14:37:02.883Z",
    "updatedAt": "2018-08-31T14:37:02.883Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "to": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
        }
      }
    ]
  }
}
```

```
"type": "regular_commission",
"issuer": {
  "id": "string",
  "sn": "string",
  "currency": "string"
},
"amount": 0,
"performedAt": "2018-08-31T14:37:02.887Z",
"issuer": {
  "id": "string",
  "sn": "string",
  "currency": "string"
},
},
],
"children": [
  {
    "errorMessage": "string"
  },
  {
    "status": "ok",
    "message": "string"
  }
]
```

Input collected cash to cash desk scheme

Use case: Input collected cash to cash desk**Basic FFlow****Optional Web UI Flow**

View cash input details description

Use Case Name

View cash input details

Brief Description

A User or External Entity on behalf of a User with role permission "CASHIER" will go through all steps of "Input collected cash to cash desk" Use Case, and then send a request to Endpoint "View cash input details". The requestIdentifier is obtained in the "Input collected cash to cash desk" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permission: "CASHIER".
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Input collected cash to cash desk".
2. External Entity sends a request to Endpoint "View cash input details".

Endpoint URL: GET /inputs/{requestIdentifier}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash input transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Input collected cash to cash desk".
2. A user sends a request to Endpoint "View cash input details".

Endpoint URL: GET /inputs/{requestIdentifier}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash input transaction information to User (See Result example below).

Post Conditions

Cash has been given to Cashier and a transaction record is available.

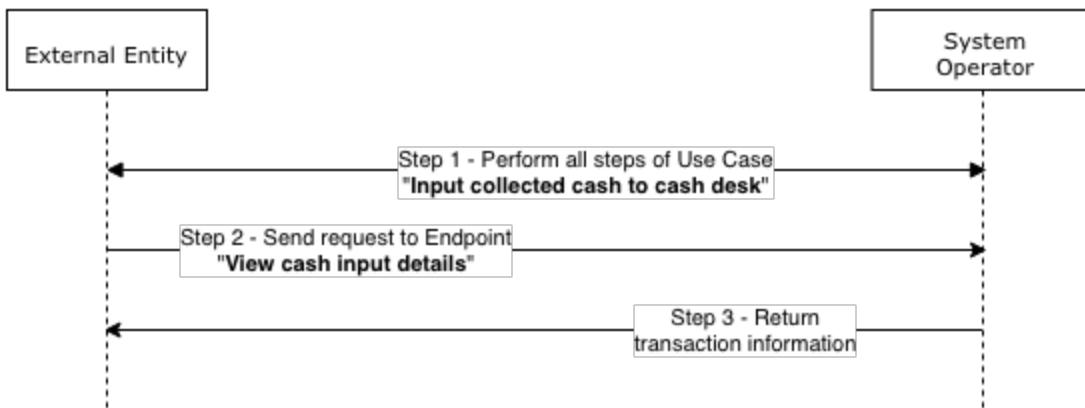
Result example

```
{  
  "id": "string",  
  "amount": 0,  
  "currency": {  
    "code": "string",  
    "digitalCode": "string",  
    "symbol": "string",  
    "name": "string",  
    "description": "string"  
  },  
  "cashDeskSerial": "string",  
  "fullName": "string",  
  "processStatus": "string",  
  "type": "string",  
  "status": "ok",  
  "message": "string"  
}
```

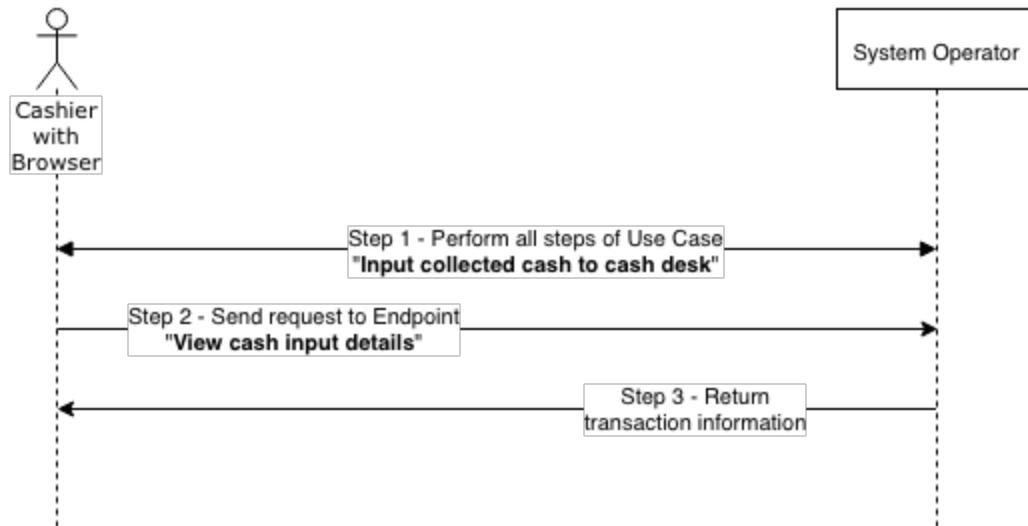
View cash input details scheme

Use case: View cash input details

Basic FFlow



Optional Web UI Flow



Cashdesk - working day management

Accept closing of working day description

Use Case Name

Accept closing of working day

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_WORKING_DAYS_MANAGER" will go through all steps of "Get the cashdesk withdrawal requests which are requested for closing with Pagination" (to get organization IDs for cash desks that have requested to close) Use Case, and then send a request to Endpoint "Accept closing of working day".

The organizationId/cash desk ID is the cash desk organization information.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_WORKING_DAYS_MANAGER", e.g. accountant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get the cashdesk withdrawal requests which are requested for closing with Pagination".
2. External Entity sends a request to Endpoint "Accept closing of working day".

Endpoint URL: POST /working-days/confirm-request-closing

```
Parameters: {
  "organizationId": "string"
}
```

1. System Operator returns cashier, status and cash desk information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get the cashdesk withdrawal requests which are requested for closing with Pagination".
2. A user sends a request to Endpoint "Accept closing of working day".

Endpoint URL: POST /working-days/confirm-request-closing

```
Parameters: {
  "organizationId": "string"
}
```

1. System Operator returns cashier, status and cash desk information to User (See Result example below).

Post Conditions

Working day record is available.

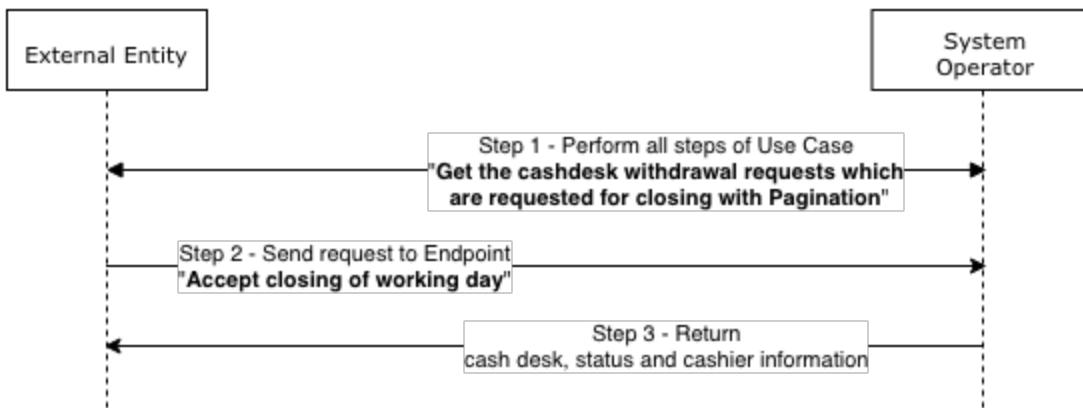
Result example

```
{  
  "workingDay": {  
    "id": "string",  
    "cashdesk": {  
      "id": "string",  
      "type": "string",  
      "name": "string"  
    },  
    "cashier": {  
      "id": "string",  
      "name": "string"  
    },  
    "status": "opened",  
    "createdAt": "2018-08-31T14:37:08.714Z",  
    "start": "2018-08-31T14:37:08.714Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

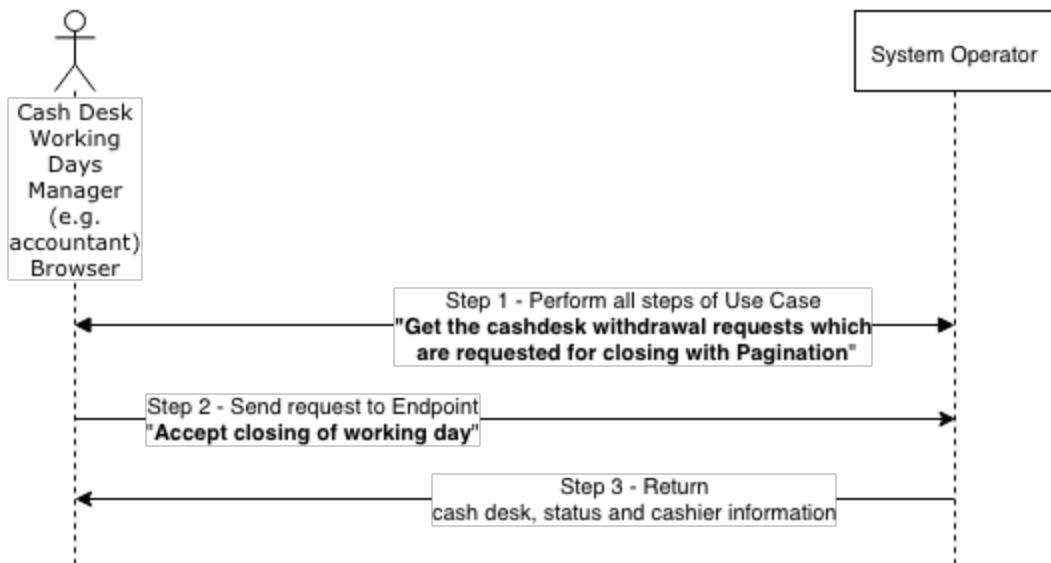
Accept closing of working day scheme

Use case: Accept closing of working day

Basic FFlow



Optional Web UI Flow



Get the cashdesk withdrawal requests which are requested for closing with Pagination description

Use Case Name

Get the cashdesk withdrawal requests which are requested for closing with Pagination

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_WORKING_DAYS_MANAGER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get the cashdesk withdrawal requests which are requested for closing with Pagination". This Use Case allows the user to sort the results on date and ascending/descending order. It also provides the input parameters for the "Accept closing of working day" EP.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_WORKING_DAYS_MANAGER", e.g. accountant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get the cashdesk withdrawal requests which are requested for closing with Pagination".

Endpoint URL: POST /working-days/request-closing/view

Parameters: {

```
"filter": {},  
"sort": {  
    "date": "asc"  
},  
"pageNumber": 0,  
"pageSize": 0  
}
```

1. System Operator returns list of cash desks that have requested closure for the working day to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get the cashdesk withdrawal requests which are requested for closing with Pagination".

Endpoint URL: POST /working-days/request-closing/view

```
Parameters: {
  "filter": {},
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

1. System Operator returns list of cash desks that have requested closure for the working day to User (See Result example below).

Post Conditions

One or more requests to close a cash desk are available.

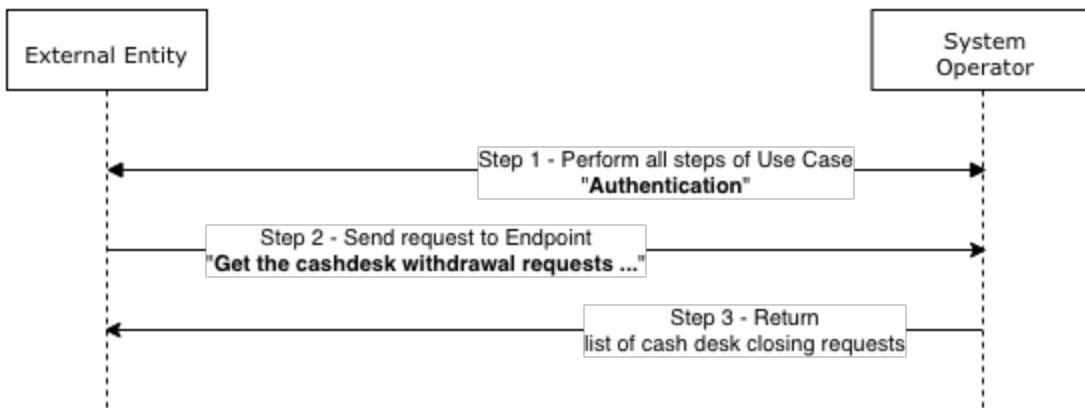
Result example

```
{
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "records": [
    {
      "id": "string",
      "cashdesk": {
        "id": "string",
        "type": "string",
        "name": "string"
      },
      "cashier": {
        "id": "string",
        "name": "string"
      },
      "status": "opened",
      "createdAt": "2018-08-31T14:37:08.726Z",
      "start": "2018-08-31T14:37:08.726Z"
    }
  ]
}
```

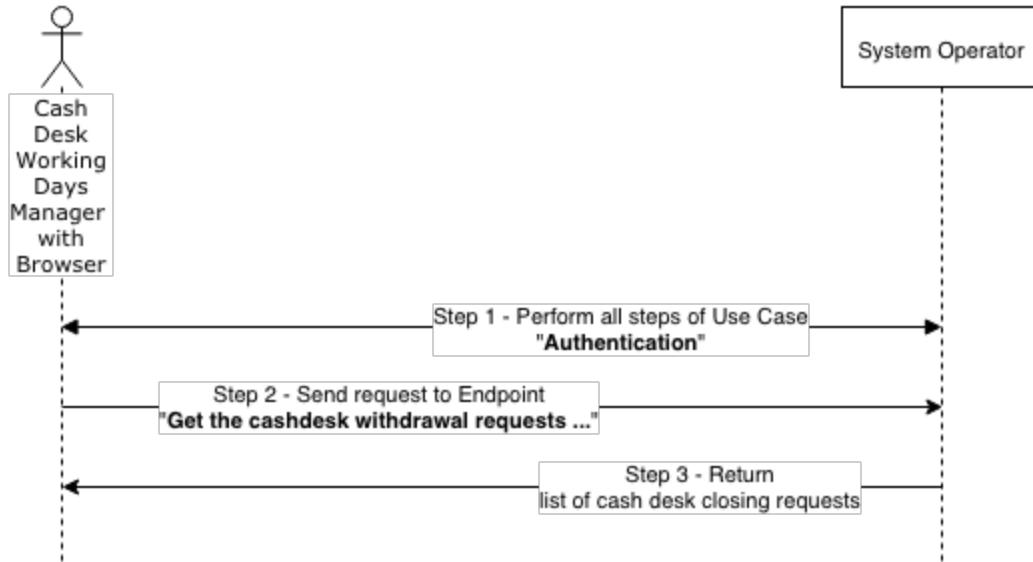
```
]  
}  
Get the cashdesk withdrawal requests which are requested for closing with Pagination scheme
```

Use case: Get the cashdesk withdrawal requests which are requested for closing with Pagination

Basic FFlow



Optional Web UI Flow



Send request to close a working day description

Use Case Name

Send request to close a working day

Brief Description

A User or External Entity on behalf of a User with role permission "CASHIER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Send request to close a working day".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASHIER".
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Send request to close a working day".

Endpoint URL: POST /working-days/request-closing

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cashier, cash desk and status (e.g. Closed) information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Send request to close a working day".

Endpoint URL: POST /working-days/request-closing

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cashier, cash desk and status (e.g. Closed) information to User (See Result example below).

Post Conditions

Token is available.

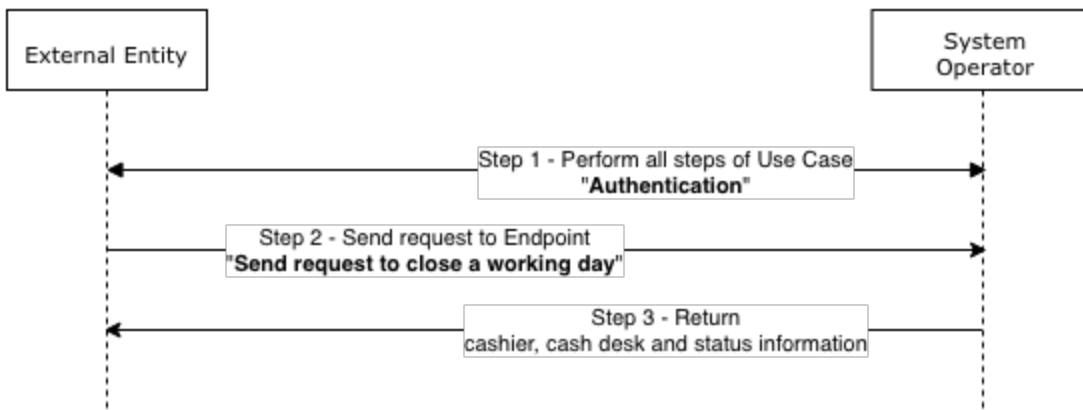
Result example

```
{  
  "workingDay": {  
    "id": "string",  
    "cashdesk": {  
      "id": "string",  
      "type": "string",  
      "name": "string"  
    },  
    "cashier": {  
      "id": "string",  
      "name": "string"  
    },  
    "status": "opened",  
    "createdAt": "2018-08-31T14:37:08.720Z",  
    "start": "2018-08-31T14:37:08.720Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

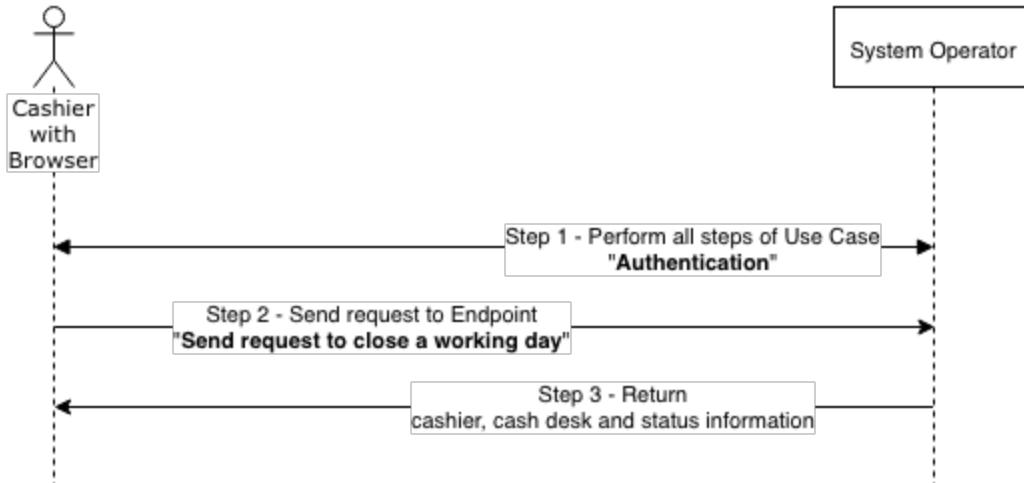
Send request to close a working day scheme

Use case: Send request to close a working day

Basic FFlow



Optional Web UI Flow



View the current working day description

Use Case Name

View the current working day

Brief Description

A User or External Entity on behalf of a User with role permission "CASHIER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View the current working day".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASHIER".
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View the current working day".

Endpoint URL: GET /working-days/current

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash desk, status (e.g. Opened) and cashier information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "View the current working day".

Endpoint URL: GET /working-days/current

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash desk, cashier and status (e.g. Opened) information to User (See Result example below).

Post Conditions

Token is available.

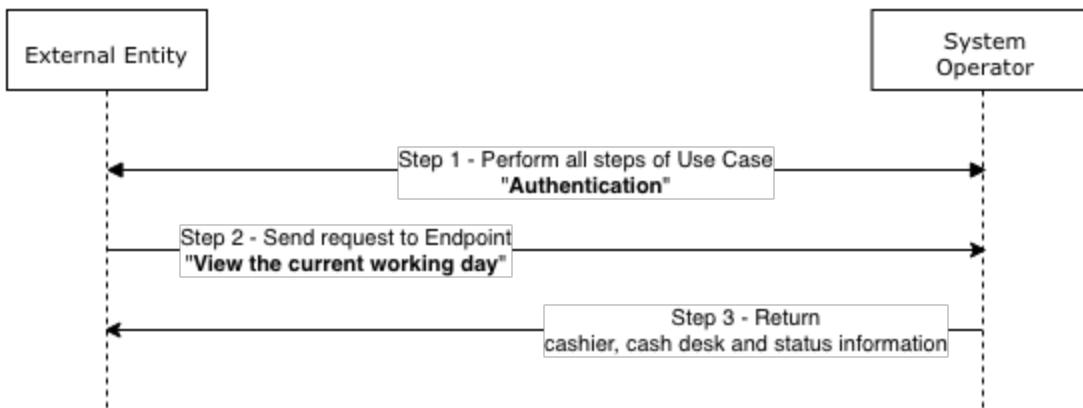
Result example

```
{  
  "workingDay": {  
    "id": "string",  
    "cashdesk": {  
      "id": "string",  
      "type": "string",  
      "name": "string"  
    },  
    "cashier": {  
      "id": "string",  
      "name": "string"  
    },  
    "status": "opened",  
    "createdAt": "2018-08-31T14:37:08.717Z",  
    "start": "2018-08-31T14:37:08.717Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

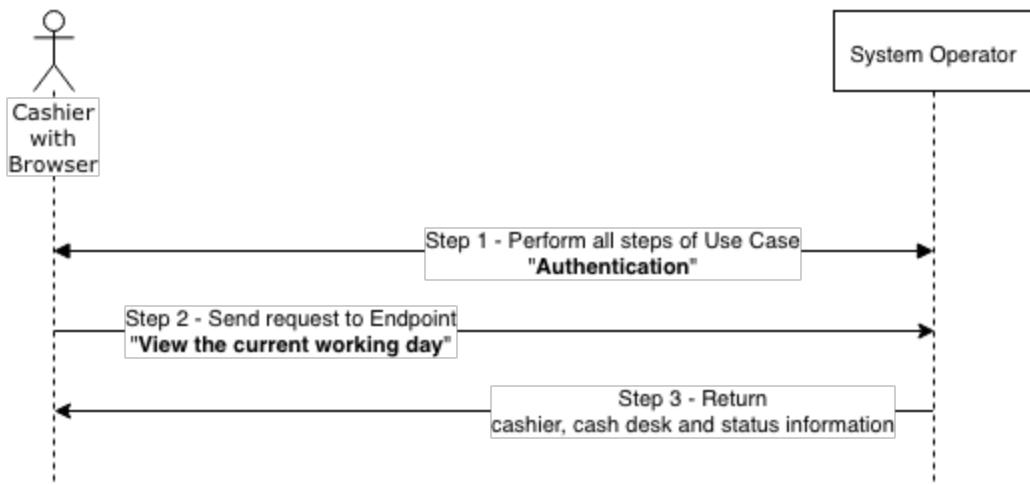
View the current working day scheme

Use case: View the current working day

Basic FLow



Optional Web UI Flow



Cash desk operations

Get cash desks for specified coin description

Use Case Name

Get cash desks for specified coin

Brief Description

External Entity acts as a registered User inside System Operator with role CASH_DESK_CLIENT.

System Operator is the company using “SDK.Finance” software.

External Entity calls System Operator API function with a valid Coin Identifier as a parameter.

Each Coin is associated with only one Currency.

System Operator determines a needed Currency from Coin parameter and returns to External Entity a list of all Cash Desks that operate in that Currency.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user with role CASH_DESK_CLIENT.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. Valid User Profile
2. Valid Coin

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity requests the list of Coins of Authenticated User.

Endpoint URL: GET /coins

3. System Operator returns a List of all Coins.



4. External Entity sends a request to System Operator to get a list of Cash Desks operating in specified Currency.

Endpoint URL: /cash-desks/view

Parameters:

```
{  
    "coinSerial": "string"  
}
```

5. System Operator returns a list of Cash Desks.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The individual selects a Coin from the Cabinet and sends a request to get Cash Desks for this Coin.

Endpoint URL: /cash-desks/view

Parameters:

```
{  
    "coinSerial": "string"  
}
```

3. System Operator returns a page with the list of Cash Desks.

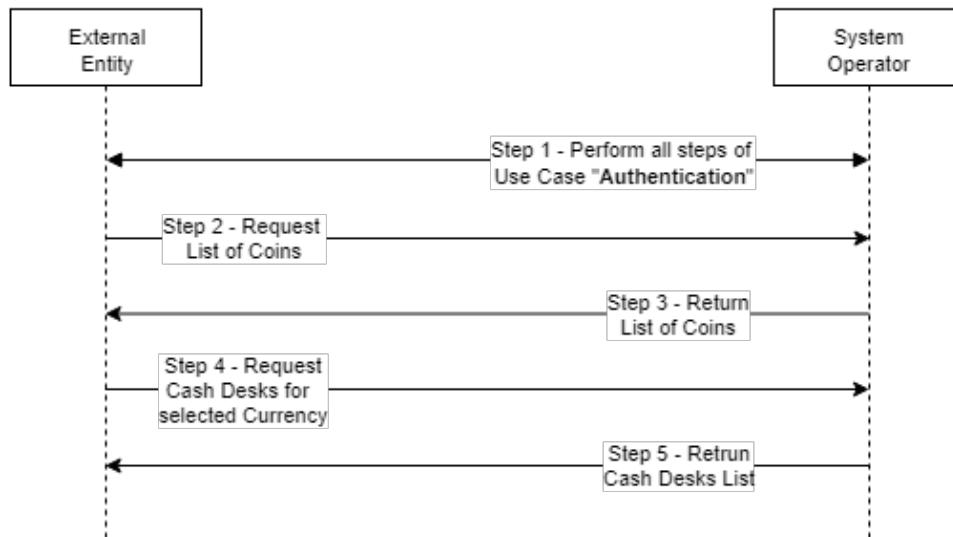
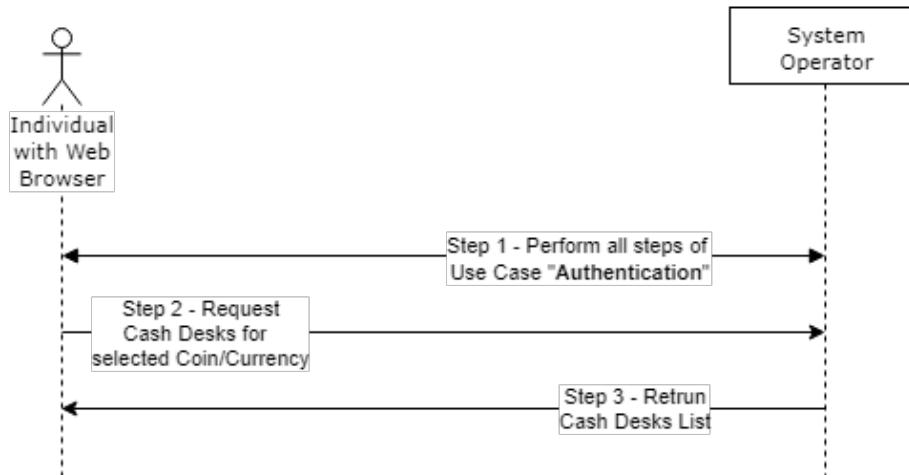
Post Conditions

List of Cash Desks operating in specified Currency is available.

Response example

```
{  
    "status": "ok",  
    "message": "string",  
    "records": [  
        {  
            "id": "string",  
            "name": "string",  
            "issuers": [  
                {  
                    "id": "string",  
                    "sn": "string",  
                    "name": "string",  
                    "description": "string",  
                    "orderNumber": 0,  
                    "orderQuote": 0,  
                    "active": false,  
                    "currency": {  
                        "code": "string",  
                        "digitalCode": "string",  
                        "symbol": "string",  
                        "name": "string",  
                        "description": "string"  
                    }  
                }  
            ],  
            "address": "string",  
            "type": "base",  
            "coordinate": {  
                "latitude": "string",  
                "longitude": "string"  
            }  
        }  
    ]  
}
```

Get cash desks for specified coin scheme

Use case: Get cash desks for specified coin**Basic FLow****Optional Web UI Flow**

[View cash desk turnover report description](#)

Use Case Name

[View cash desk turnover report](#)

Brief Description

External Entity acts as a registered Accountant inside System Operator.

System Operator is the company using “SDK.Finance” software.

External Entity calls System Operator API function to get the Cash desk turnover report.

System Operator selects cash desk transaction history and returns to a caller.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator Accountant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. Valid Accountant Profile
2. Valid cash desk

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity .

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to System Operator to get a Turnover report.

Endpoint URL: POST /cash-desks/report-turnover

Parameters:

```
{
  "filter": {
    "types": [
      "string"
    ]
  },
  "sort": {
    "creationDate": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns a report (see result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to System Operator to get a Turnover report.

Endpoint URL: POST /cash-desks/report-turnover

Parameters:

```
{
  "filter": {
    "types": [
      "string"
    ]
  },
  "sort": {
    "creationDate": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns a report (see result example below)

Post Conditions

Turnover report available.

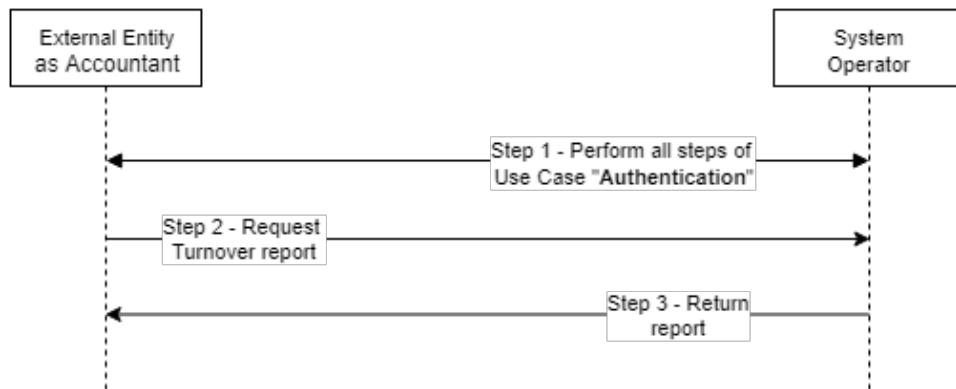
Result example

```
{
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "status": "ok",
  "message": "string",
  "records": [
    {
    }
  ]
}
```

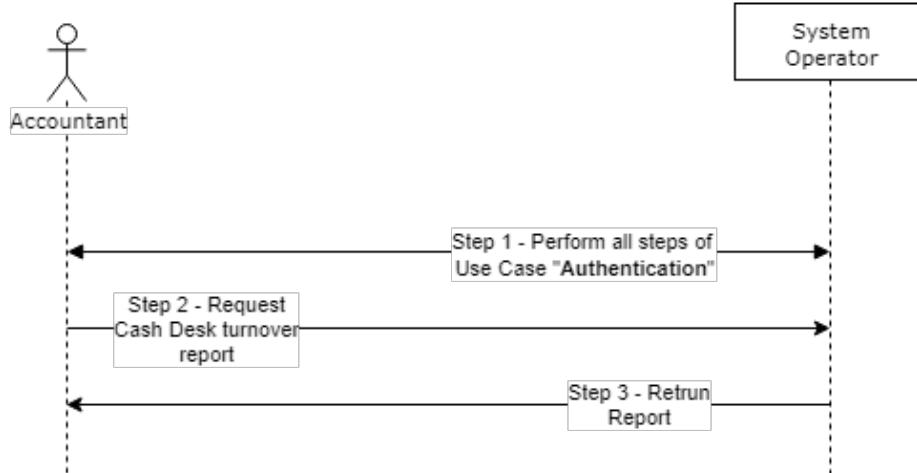
View cash desk turnover report scheme

Use case: View cash desk turnover report @

Basic FFlow



Optional Web UI Flow



Cash desk collect

Accept collect request description

Use Case Name

Accept collect request

Brief Description

A User or External Entity on behalf of a User with role permission “CASHIER” will go through all steps of “View transaction details” Use Case, and then send a request to Endpoint “Accept collect request”. Cashier gets “requestIdentifier” from the Executor in the “View transaction details” Use Case. The Executor will have received the “requestIdentifier” from the “Cash collect request” Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: “CASHIER”.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

3. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View transaction details”.
2. External Entity sends a request to Endpoint “Accept collect request”.

Endpoint URL: POST /collects/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

3. System Operator returns cash collect transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View transaction details”.
2. A user sends a request to Endpoint “Accept collect request”.

Endpoint URL: POST /collects/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

3. System Operator returns cash collect transaction information to User (See Result example below).

Post Conditions

Cash collect request is available.

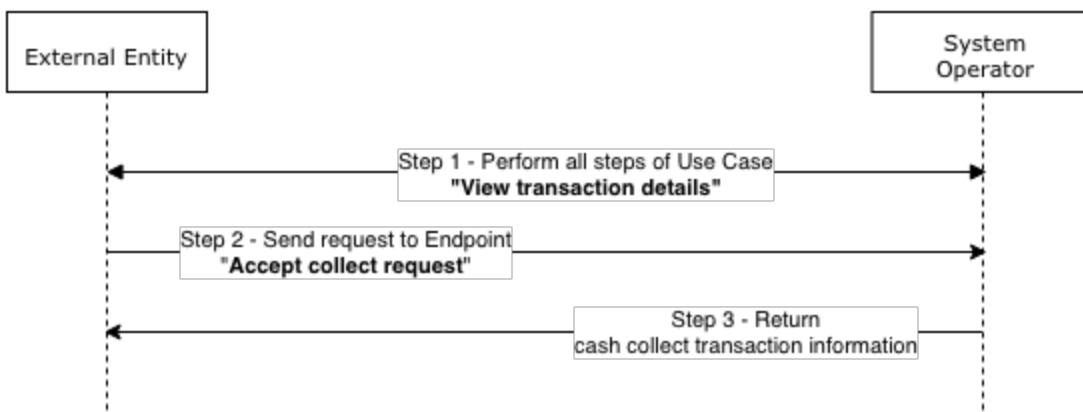
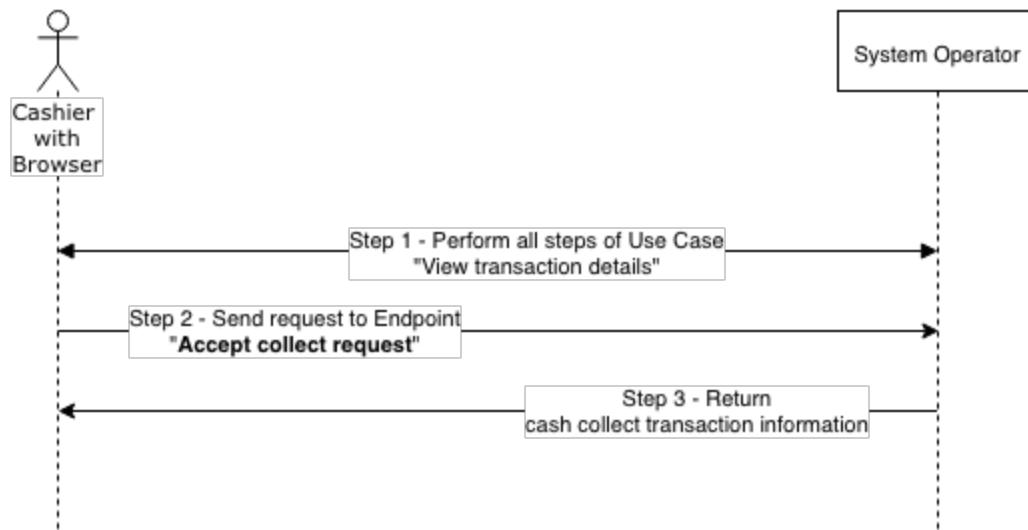
Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-08-29T10:07:45.706Z",
    "updatedAt": "2018-08-29T10:07:45.706Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "id": "string"
        }
      }
    ]
  }
}
```

```
        "serial": "string",
        "organizationId": "string",
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "to": {
        "serial": "string",
        "organizationId": "string",
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "amount": 0,
    "performedAt": "2018-08-29T10:07:45.706Z",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
},
"children": [
    {
        ...
    }
],
"errorMessage": "string"
},
```

```
"status": "ok",
"message": "string"
}
```

Accept collect request scheme

Use case: Accept collect request**Basic FFlow****Optional Web UI Flow**

Cash collect request description

Use Case Name

Cash collect request

Brief Description

A User or External Entity on behalf of a User with role permission CASH_DESK_COLLECT_EXECUTOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Cash collect request”. In this Use Case, a CFO or an Accountant with the role CASH_DESK_COLLECT_EXECUTOR goes to the Cash Desk to collect cash for some reason, e.g. to deposit in a partner bank. This Executor role must know the amount to collect, coin ID, and name of the cash receiver.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CASH_DESK_COLLECT_EXECUTOR, e.g. accountant or CFO.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Cash collect request”.

Endpoint URL: POST /collects

```
Parameters: {
  "amount": 0,
  "serial": "string",
  "fullName": "string"
}
```

1. System Operator returns cash collect request information including requestId to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Cash collect request”.

Endpoint URL: POST /collects

```
Parameters: {
  "amount": 0,
  "serial": "string",
```

```
"fullName": "string"  
}
```

1. System Operator returns cash collect request information including requestId to User (See Result example below).

Post Conditions

The Executor role must know the amount to collect, coin ID, and name of the cash receiver.

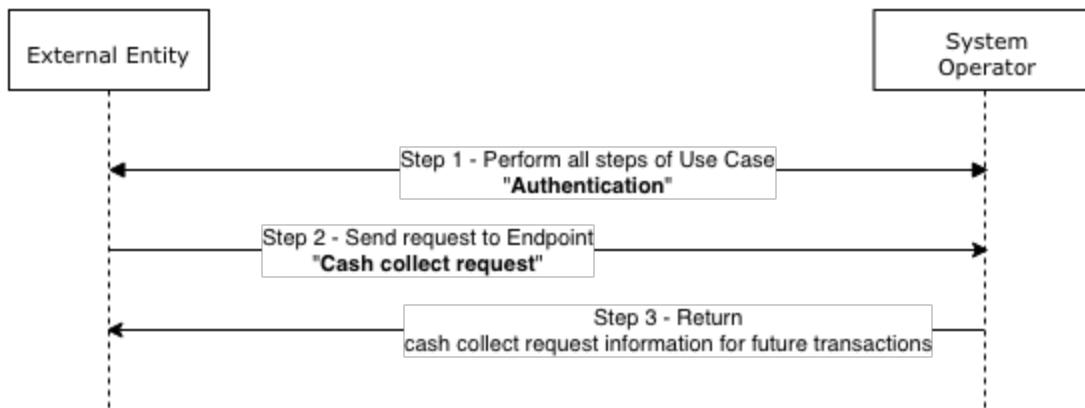
Result example

```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-08-29T10:07:45.669Z",  
    "updatedAt": "2018-08-29T10:07:45.669Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        }  
      }  
    ]  
  }  
}
```

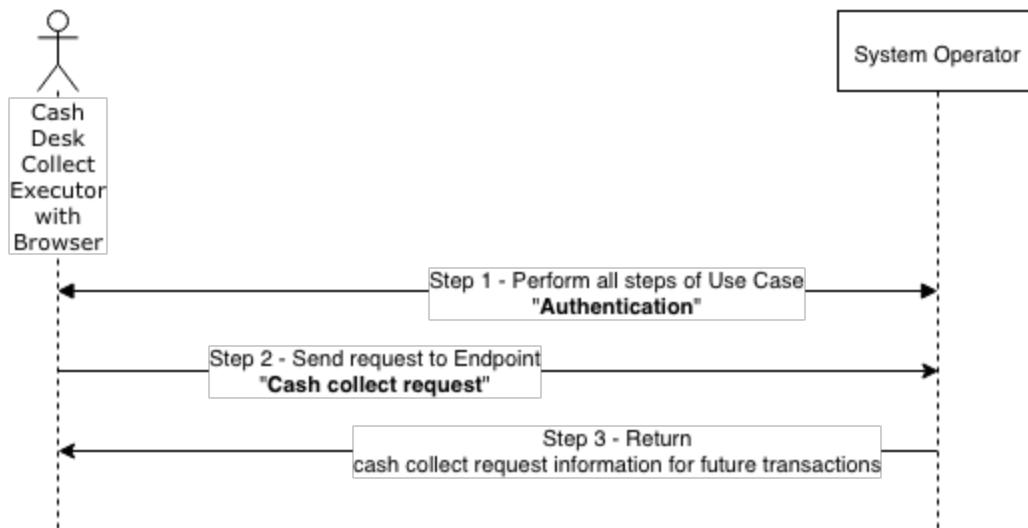
```
"technical": false,  
"type": "regular_commission",  
"issuer": {  
    "id": "string",  
    "sn": "string",  
    "currency": "string"  
}  
,  
"amount": 0,  
"performedAt": "2018-08-29T10:07:45.672Z",  
"issuer": {  
    "id": "string",  
    "sn": "string",  
    "currency": "string"  
}  
,  
]  
,  
"children": [  
{}  
,  
],  
"errorMessage": "string"  
,  
"status": "ok",  
"message": "string"  
}  
Cash collect request scheme
```

Use case: Cash collect request

Basic FFlow



Optional Web UI Flow



View transaction details description

Use Case Name

View transaction details

Brief Description

A User or External Entity on behalf of a User with role permission "CASHIER" will go through all steps of "Cash collect request" (to obtain a request ID) Use Case, and then send a request to Endpoint "View transaction details". The Executor from the "Cash collect request" Use Case gives the Cashier at the cash desk the "requestIdentifier" obtained.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASHIER".
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Cash collect request".
2. External Entity sends a request to Endpoint "View transaction details".

Endpoint URL: GET /collects/{requestIdentifier}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns transaction request details to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Cash collect request".
2. A user sends a request to Endpoint "View transaction details".

Endpoint URL: GET /collects/{requestIdentifier}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns transaction request details to User (See Result example below).

Post Conditions

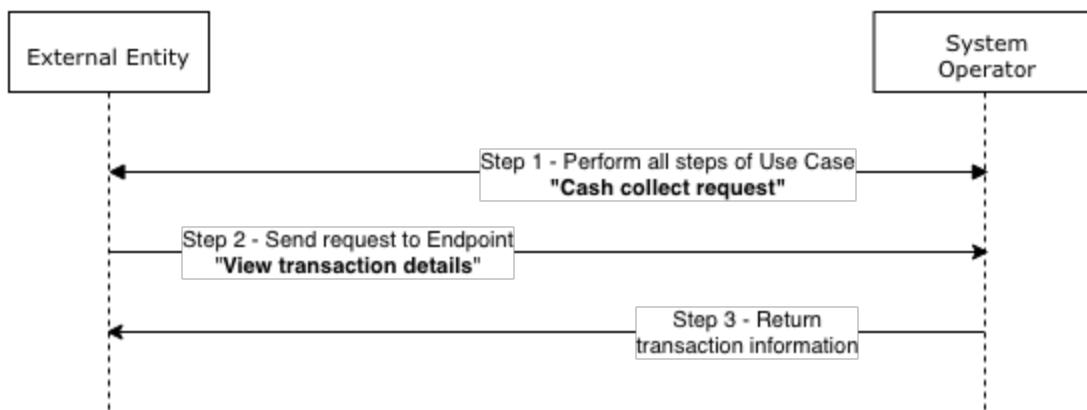
Transaction request identifier is available.

Result example

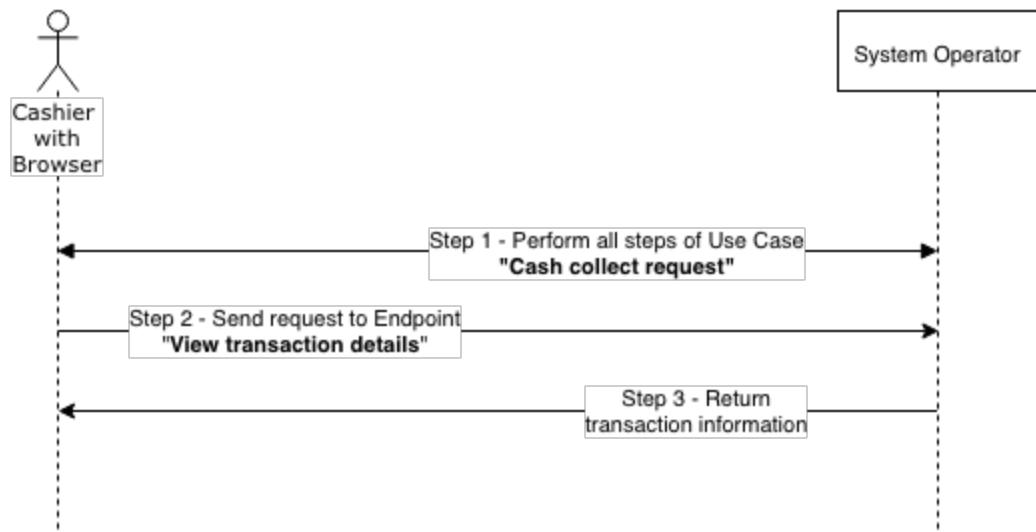
```
{  
    "id": "string",  
    "amount": 0,  
    "currency": {  
        "code": "string",  
        "digitalCode": "string",  
        "symbol": "string",  
        "name": "string",  
        "description": "string"  
    },  
    "cashDeskSerial": "string",  
    "fullName": "string",  
    "processStatus": "string",  
    "type": "string",  
    "status": "ok",  
    "message": "string"  
}  
View transaction details scheme
```

Use case: View transaction details

Basic FFlow



Optional Web UI Flow



Exchange

Calculate commission for exchange operation description

Use Case Name

Calculate commission for exchange operation

Brief Description

A User or External Entity on behalf of a User with role permission EXCHANGE_EXECUTOR will go through all steps of “View exchange rates” Use Case, and then send a request to Endpoint “Calculate commission for exchange operation”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: EXCHANGE_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View exchange rates”.
2. External Entity sends a request to Endpoint “Calculate commission for exchange operation”.

Endpoint URL: POST /exchange-rates/{rateId}/exchanges/calculate

Parameter:

```
{
  "inCoin": "string",
  "outCoin": "string",
  "inAmount": 0
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View exchange rates”.
2. A user sends a request to Endpoint “Calculate commission for exchange operation”.

Endpoint URL: POST /exchange-rates/{rateId}/exchanges/calculate

Parameter:

```
{
  "inCoin": "string",
  "outCoin": "string",
  "inAmount": 0
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

Commission amount is available

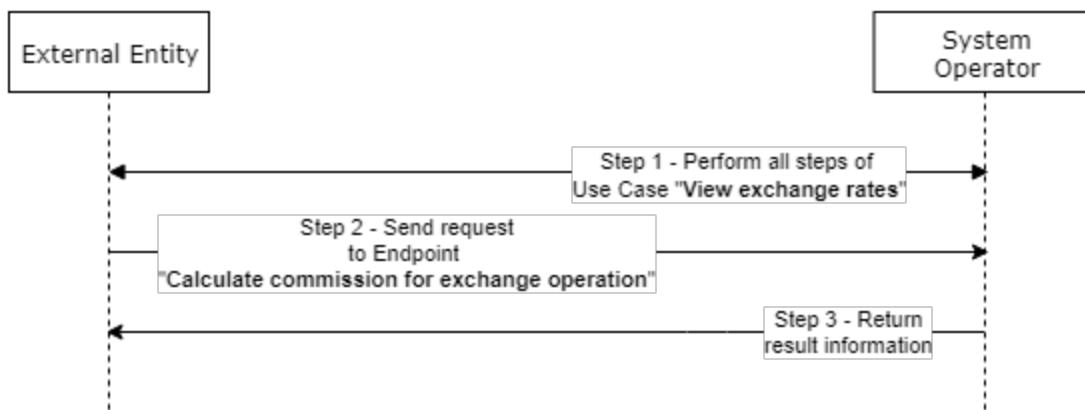
Result example

```
{  
    "in": {  
        "transactionAmount": 0,  
        "senderAmountPush": 0,  
        "recipientAmountPush": 0,  
        "commissionAmountPush": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        }  
    },  
    "out": {  
        "transactionAmount": 0,  
        "senderAmountPush": 0,  
        "recipientAmountPush": 0,  
        "commissionAmountPush": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

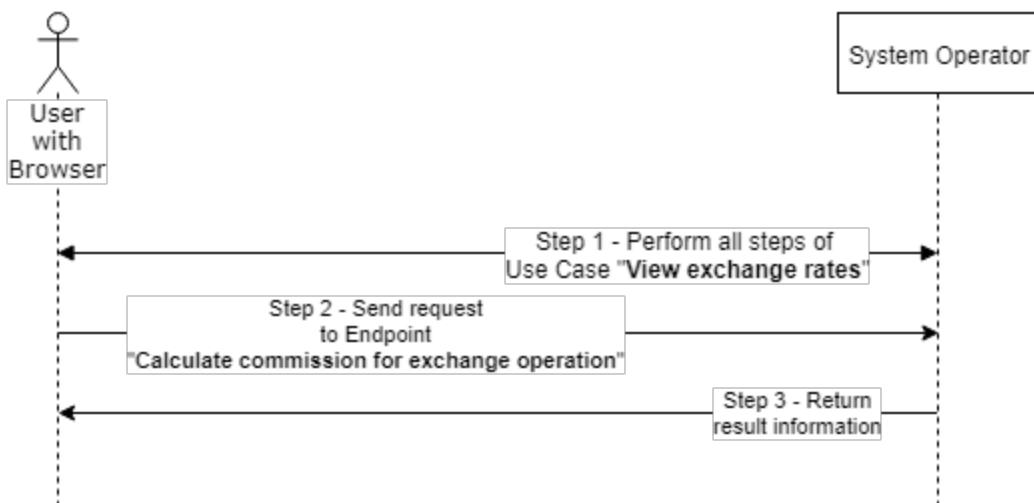
Calculate commission for exchange operation scheme

Use case: Calculate commission for exchange operation

Basic FFlow



Optional Web UI Flow



Change currency exchange rate status to opposite description

Use Case Name

Change currency exchange rate status to opposite

Brief Description

A User or External Entity on behalf of a User with role permission EXCHANGE_MANAGER will go through all steps of “View exchange rates” Use Case, and then send a request to Endpoint “Change currency exchange rate status to opposite”. As a result, the status will toggle the value of ACTIVE between false and true.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: EXCHANGE_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View exchange rates”.
2. External Entity sends a request to Endpoint “Change currency exchange rate status to opposite”.

Endpoint URL: POST /exchange-rates/{rateId}/status

Parameter: Security TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View exchange rates”.
2. A user sends a request to Endpoint “Change currency exchange rate status to opposite”.

Endpoint URL: POST /exchange-rates/{rateId}/status

Parameter: Security TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

If the Active value was TRUE, it becomes FALSE and vice-versa.

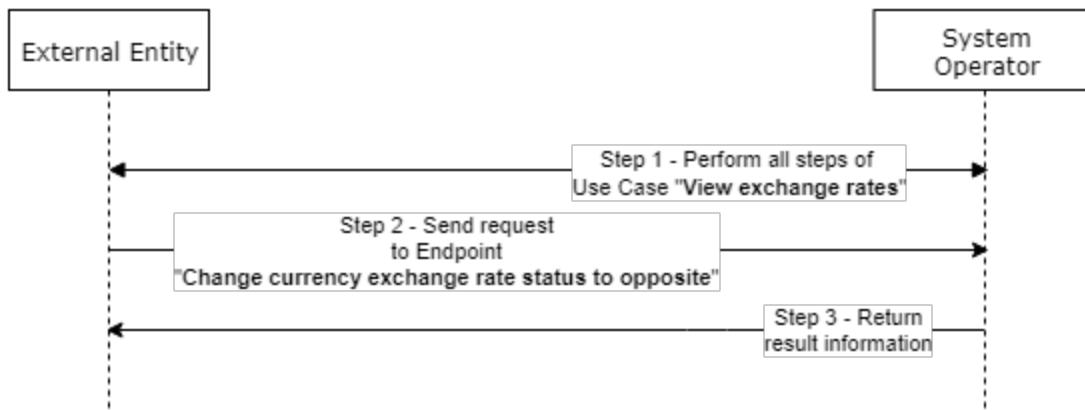
Result example

```
{  
    "rate": {  
        "id": "string",  
        "inIssuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "outIssuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "rate": 0,  
        "direction": "buy",  
        "exchanger": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "reserve": 0,  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

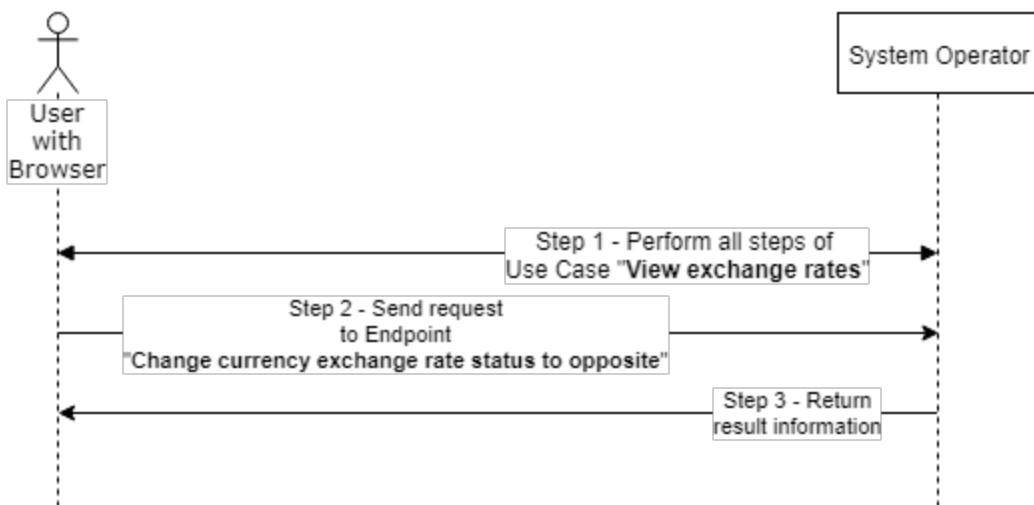
Change currency exchange rate status to opposite scheme

Use case: Change currency exchange rate status to opposite

Basic FFlow



Optional Web UI Flow



Execute exchange operation description

Use Case Name

Change currency exchange rate status to opposite

Brief Description

A User or External Entity on behalf of a User with role permission EXCHANGE_MANAGER will go through all steps of “View exchange rates” Use Case, and then send a request to Endpoint “Change currency exchange rate status to opposite”. As a result, the status will toggle the value of ACTIVE between false and true.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: EXCHANGE_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View exchange rates”.
2. External Entity sends a request to Endpoint “Change currency exchange rate status to opposite”.

Endpoint URL: POST /exchange-rates/{rateId}/status

Parameter: Security TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View exchange rates”.
2. A user sends a request to Endpoint “Change currency exchange rate status to opposite”.

Endpoint URL: POST /exchange-rates/{rateId}/status

Parameter: Security TOKEN

3. System Operator returns result information to User (See Result example below).

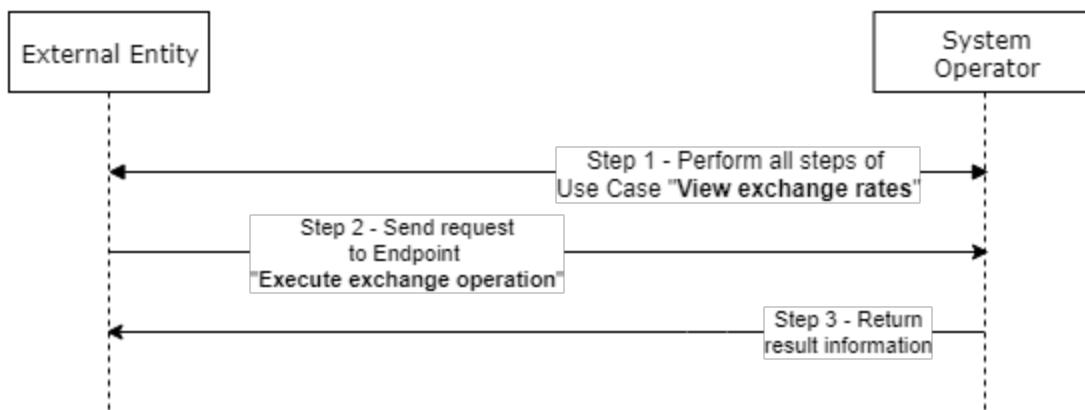
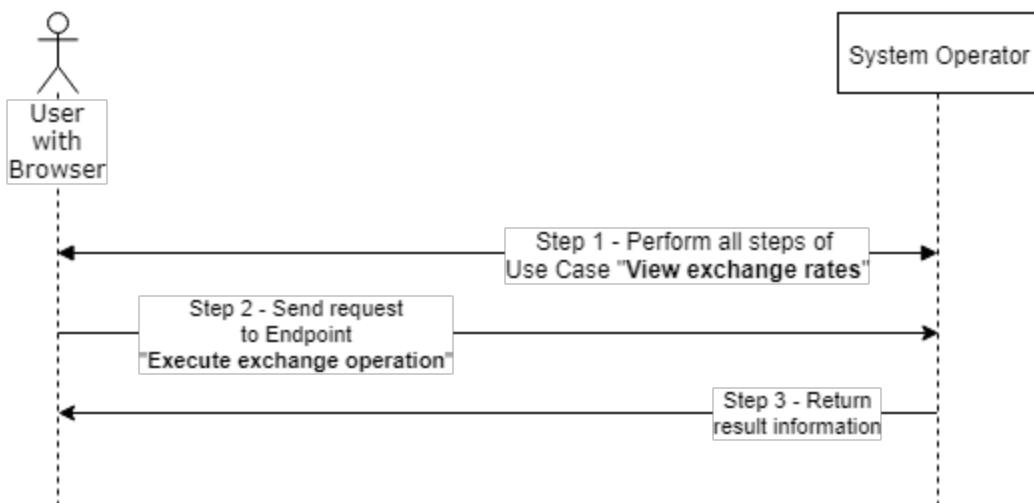
Post Conditions

If the Active value was TRUE, it becomes FALSE and vice-versa.

Result example

```
{  
    "rate": {  
        "id": "string",  
        "inIssuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "outIssuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "rate": 0,  
        "direction": "buy",  
        "exchanger": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "reserve": 0,  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Execute exchange operation scheme

Use case: Execute exchange operation**Basic FFlow****Optional Web UI Flow**

Set currency exchange rate description

Use Case Name

Set currency exchange rate

Brief Description

A User or External Entity on behalf of a User with role permission EXCHANGE_MANAGER will go through all steps of “Obtain issuers” Use Case, and then send a request to Endpoint “Set currency exchange rate”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: EXCHANGE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all the steps of Use Case “Obtain issuers”.
2. External Entity sends a request to Endpoint “Set currency exchange rate”.

Endpoint URL: POST /exchange-rates/rate

Parameter:

```
{
  "rate": 0,
  "inIssuerId": "string",
  "outIssuerId": "string",
  "direction": "buy",
  "active": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.
2. A user sends a request to Endpoint “Set currency exchange rate”.

Endpoint URL: POST /exchange-rates/rate

Parameter:

```
{
    "rate":0,
    "inIssuerId":"string",
    "outIssuerId":"string",
    "direction":"buy",
    "active":false
}
```

3. System Operator returns result information to User (See Result example below).

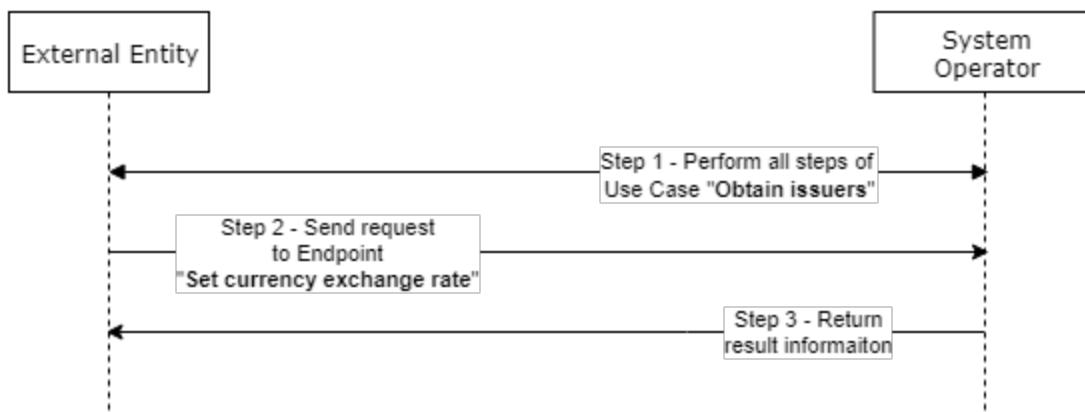
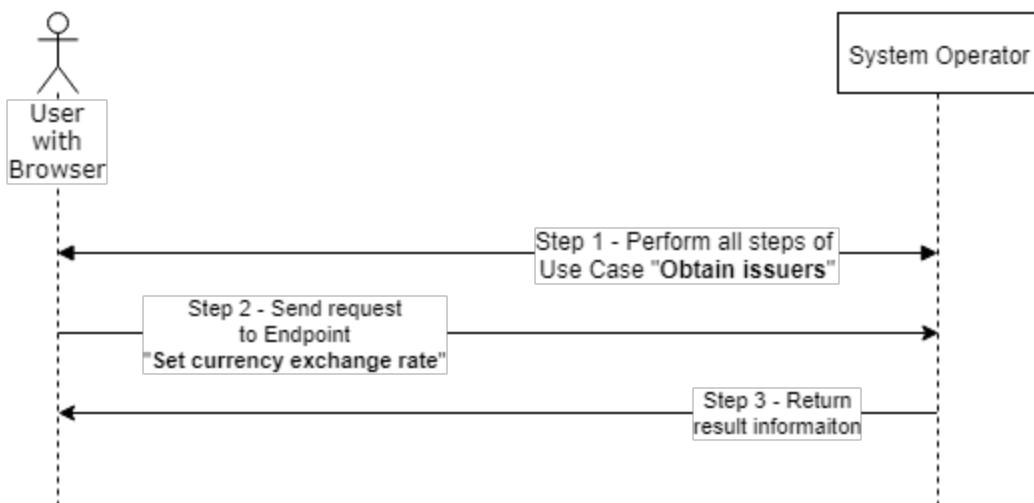
Post Conditions

Set rate took effect.

Result example

```
{
    "rate":{
        "id":"string",
        "inIssuer":{
            "id":"string",
            "sn":"string",
            "currency":"string"
        },
        "outIssuer":{
            "id":"string",
            "sn":"string",
            "currency":"string"
        },
        "rate":0,
        "direction":"buy",
        "exchanger":{
            "id":"string",
            "type":"string",
            "name":"string"
        },
        "reserve":0,
        "active":false
    },
    "status":"ok",
    "message":"string"
}
```

Set currency exchange rate scheme

Use case: Set currency exchange rate**Basic FFlow****Optional Web UI Flow**

|
View exchange rates description

Use Case Name

View exchange rates

Brief Description

A User or External Entity on behalf of a User with role permission EXCHANGE_MANAGER,
EXCHANGE_VIEWER will go through all steps of “Obtain issuers” Use Case, and then send a request to Endpoint “View exchange rates”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: EXCHANGE_MANAGER, EXCHANGE_VIEWER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all the steps of Use Case “Obtain issuers”.
2. External Entity sends a request to Endpoint “View exchange rates”.

Endpoint URL: POST /exchange-rates/view

Parameter:

```
{
  "inIssuerId": "string",
  "outIssuerId": "string"
}
```

3. System Operator returns a List of rates to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain issuers”.
2. A user sends a request to Endpoint “View exchange rates”.

Endpoint URL: POST /exchange-rates/view

Parameter:

```
{
  "inIssuerId": "string",
  "outIssuerId": "string"
}
```

3. System Operator returns a List of rates to User (See Result example below).

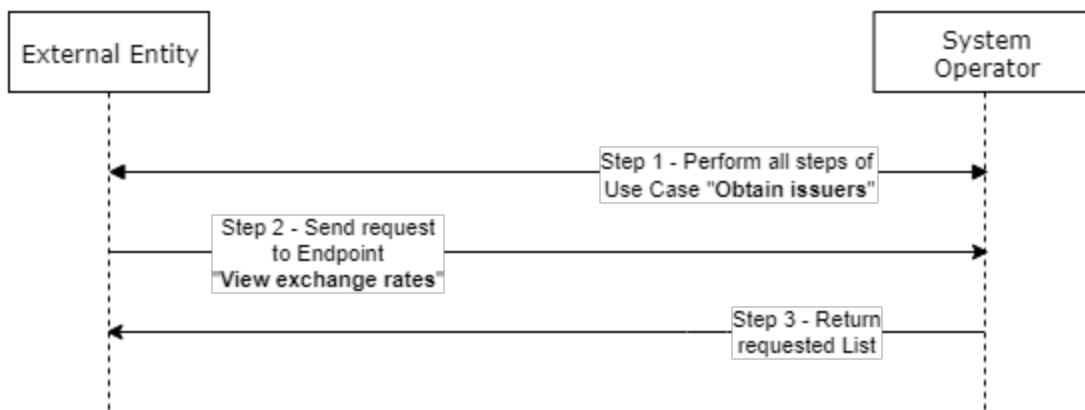
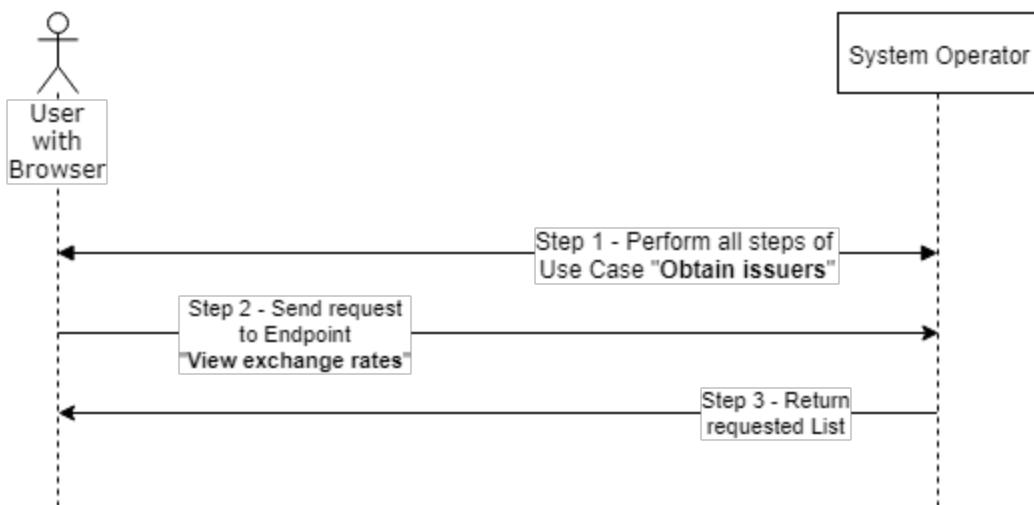
Post Conditions

A list is available

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "records": [  
        {  
            "id": "string",  
            "inIssuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            },  
            "outIssuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            },  
            "rate": 0,  
            "direction": "buy",  
            "exchanger": {  
                "id": "string",  
                "type": "string",  
                "name": "string"  
            },  
            "reserve": 0,  
            "active": false  
        }  
    ]  
}
```

[View exchange rates scheme](#)

Use case: View exchange rates**Basic FFlow****Optional Web UI Flow**

Gate provider account management

Update settings of provider account description

Use Case Name

Update settings of the provider account

Brief Description

A User or External Entity on behalf of a User with role permission GATE_PROVIDER_ACCOUNT_MANAGER will go through all steps of “View provider account settings” Use Case, and then send a request to Endpoint “Update settings of provider account”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: GATE_PROVIDER_ACCOUNT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View provider account settings”.
2. External Entity sends a request to Endpoint “Update settings of provider account”.

Endpoint URL: PATCH /gate-provider-accounts/{providerAccountId}/settings

Parameters:

```
{
  "settings": [
    {
      "name": "string",
      "value": {
        ...
      }
    }
  ]
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View provider account settings”.
2. A user sends a request to Endpoint “Update settings of provider account”.

Endpoint URL: PATCH /gate-provider-accounts/{providerAccountId}/settings

Parameters:

```
{  
    "settings": [  
        {  
            "name": "string",  
            "value": {  
                }  
        }  
    ]  
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

New information is available.

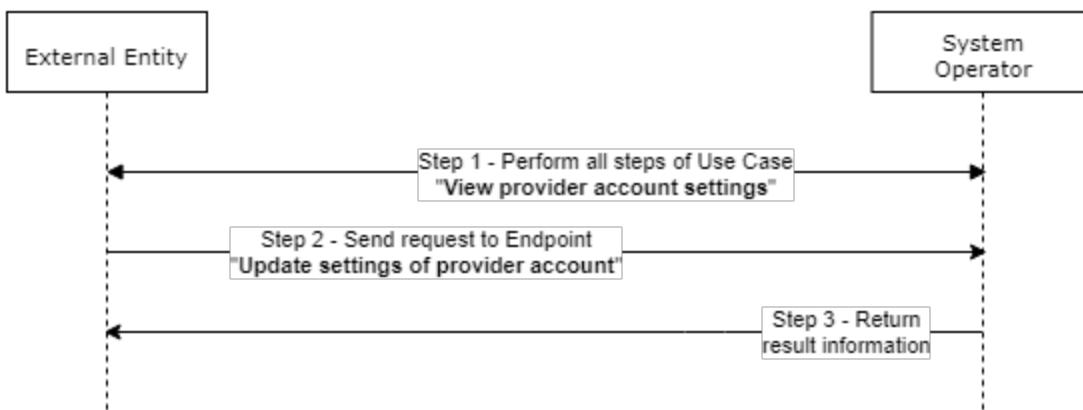
Result example

```
{  
    "settings": [  
        {  
            "name": "string",  
            "value": {  
                }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

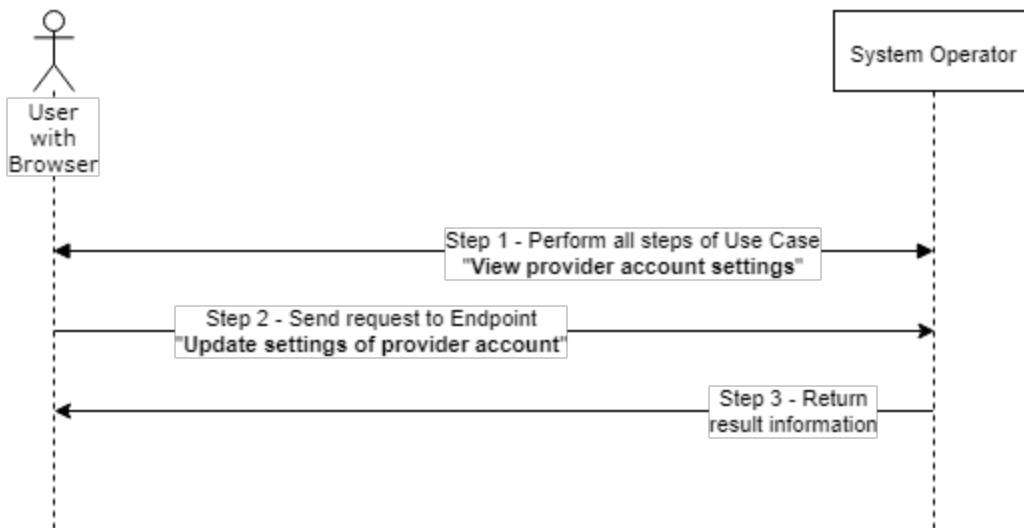
Update settings of provider account scheme

Use case: Update settings of provider account

Basic FFlow



Optional Web UI Flow



View provider account settings description

Use Case Name

View provider account settings

Brief Description

A User or External Entity on behalf of a User with role permission GATE_PROVIDER_ACCOUNT_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View provider account settings”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: GATE_PROVIDER_ACCOUNT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View provider account settings”.

Endpoint URL: GET /gate-provider-accounts/{providerAccountId}/settings

Parameters: TOKEN

3. System Operator returns List of Settings to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View provider account settings”.

Endpoint URL: GET /gate-provider-accounts/{providerAccountId}/settings

Parameters: TOKEN

3. System Operator returns List of Settings to User (See Result example below).

Post Conditions

List of Settings is available.

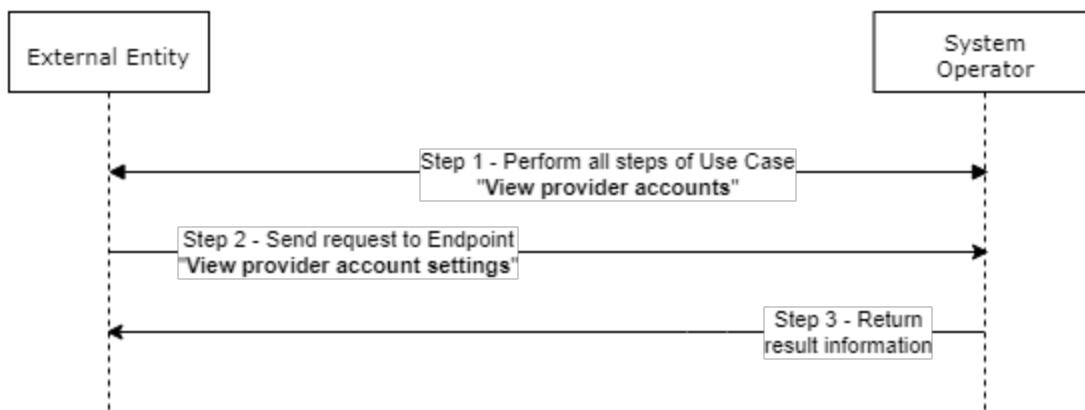
Result example

```
{
  "settings": [
    {
      "name": "string",
      "value": {
        ...
      }
    }
  ],
  "status": "ok",
  "message": "string"
}
```

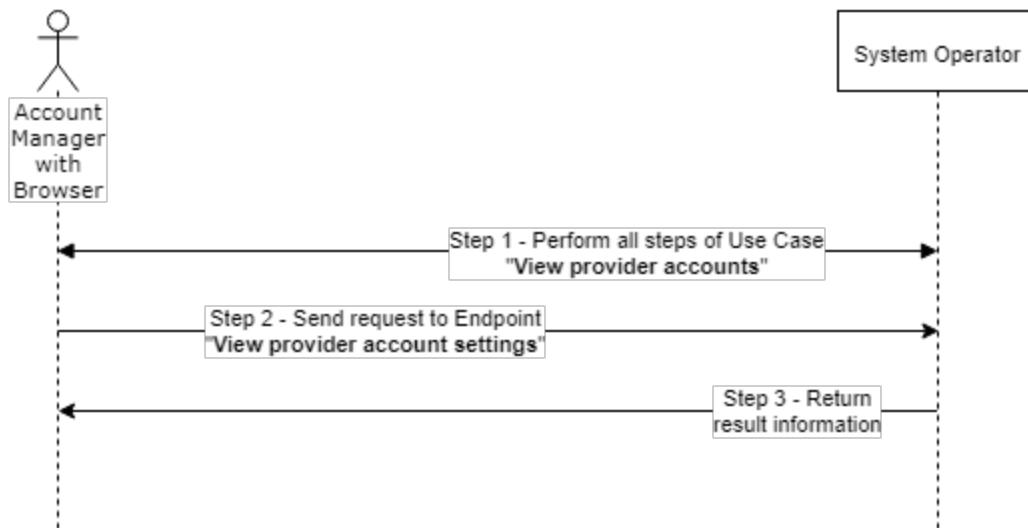
[View provider account settings scheme](#)

Use case: View provider account settings

Basic FFlow



Optional Web UI Flow



View provider accounts description

Use Case Name

View provider accounts

Brief Description

A User or External Entity on behalf of a User with role permission GATE_PROVIDER_ACCOUNT_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View provider accounts”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: GATE_PROVIDER_ACCOUNT_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View provider accounts”.

Endpoint URL: GET /gate-provider-accounts

Parameters: TOKEN

3. System Operator returns provider accounts to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View provider accounts”.

Endpoint URL: GET /gate-provider-accounts

Parameters: TOKEN

3. System Operator returns provider accounts to User (See Result example below).

Post Conditions

List of provider accounts is available.

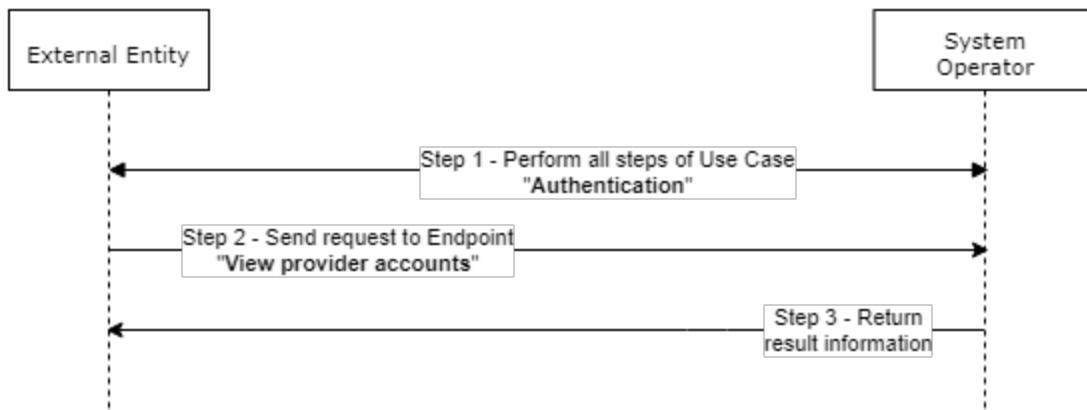
Result example

```
{  
  "provider-accounts": [  
    {  
      "id": "string",  
      "provider": {  
        "name": "string"  
      },  
      "organization": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
      },  
      "supportedTransactionTypes": [  
        "string"  
      ]  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

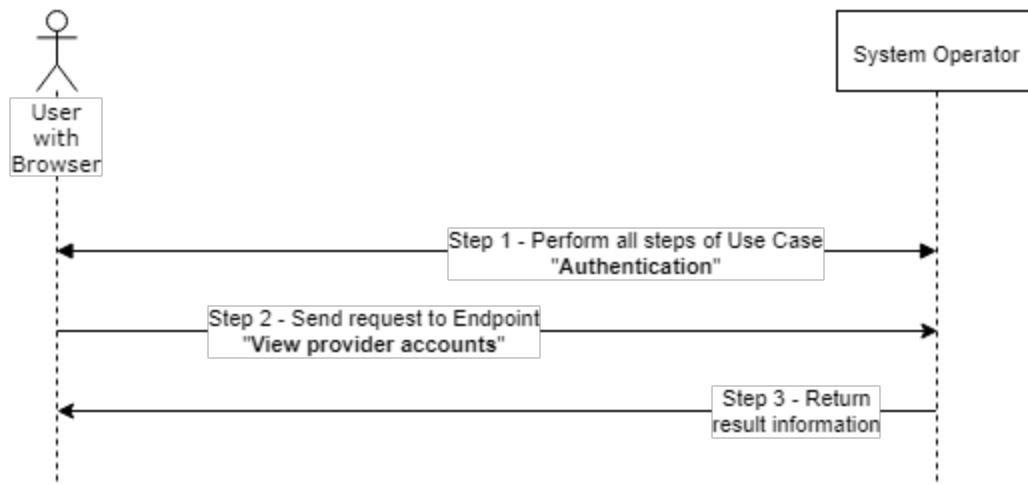
[View provider accounts scheme](#)

Use case: View provider accounts

Basic FFlow



Optional Web UI Flow



Provider coin management

View provider coins of organization description

Use Case Name

View provider coins of organization

Brief Description

A User or External Entity on behalf of a User with role permissions ORGANIZATION_VIEWER and COIN_MANAGER will go through all steps of “View organizations” Use Case, and then send a request to Endpoint “View provider coins of the organization”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ORGANIZATION_VIEWER and COIN_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View organizations”.
2. External Entity sends a request to Endpoint “View provider coins of the organization”.

Endpoint URL: GET /organizations/{id}/provider-coins

Parameters: TOKEN

3. System Operator returns provider coins information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View organizations”.
2. A user sends a request to Endpoint “View provider coins of the organization”.

Endpoint URL: GET /organizations/{id}/provider-coins

Parameters: TOKEN

3. System Operator returns provider coins information to User (See Result example below).

Post Conditions

Provider coins are available.

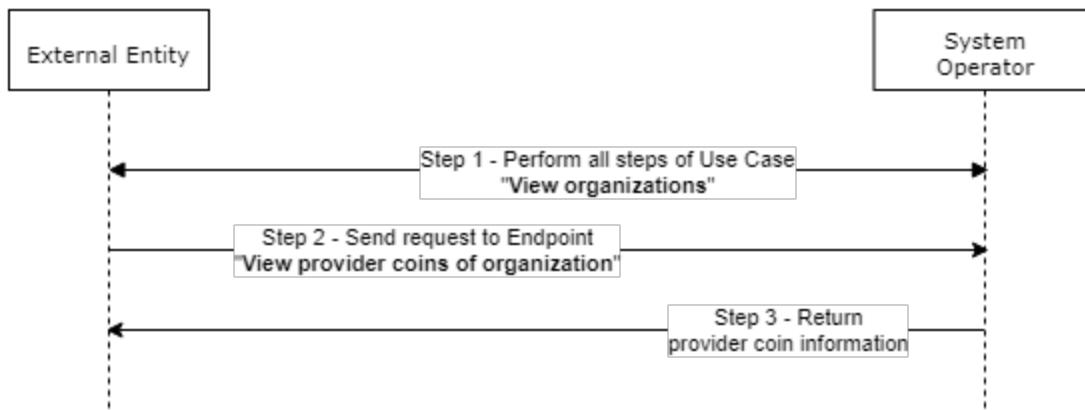
Result example

```
{  
    "coins": [  
        {  
            "serial": "string",  
            "type": "regular_commission",  
            "amount": 0,  
            "availableAmount": 0,  
            "issuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

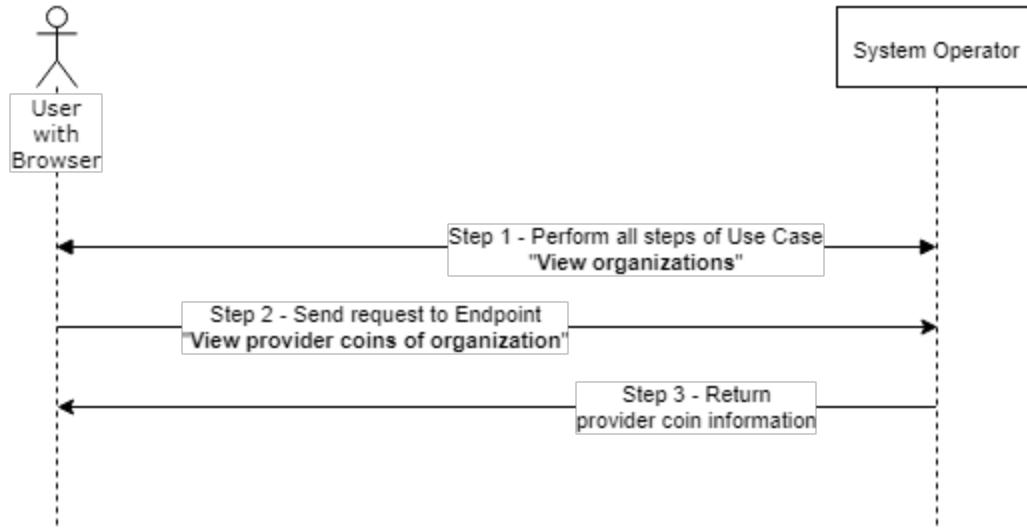
View provider coins of organization scheme

Use case: View provider coins of organization

Basic FFlow



Optional Web UI Flow



Gate investment operations

Accept investment request description

Use Case Name

Accept investment request

Brief Description

A User or External Entity on behalf of a User with role permission PROVIDER_INVESTMENT_VALIDATOR will go through all steps of “View cash investment details” Use Case, and then send a request to Endpoint “Accept investment request”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROVIDER_INVESTMENT_VALIDATOR.
 2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View cash investment details”.
 2. External Entity sends a request to Endpoint “Accept investment request”.

Endpoint URL: POST /gate-investments/{requestIdentifier}/accept

Parameters: TOKEN

3. System Operator returns XX result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View cash investment details”.
 2. A user sends a request to Endpoint “Accept investment request”.

Endpoint URL: POST /gate-investments/{requestIdentifier}/accept

Parameters: TOKEN

3. System Operator returns XX result information to User (See Result example below).

Post Conditions

The request is accepted.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-08-10T14:07:38.964Z",
    "updatedAt": "2018-08-10T14:07:38.964Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "type": "string",
        "status": "limited",
        "amount": 0,
        "date": "2018-08-10T14:07:38.964Z"
      }
    ]
  }
}
```

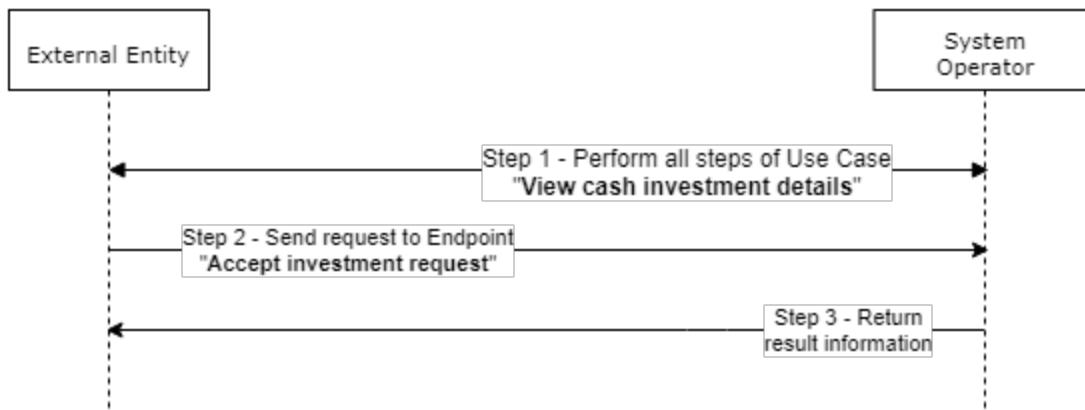
```
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        },
        "to": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        },
        "amount": 0,
        "performedAt": "2018-08-10T14:07:38.964Z",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    }
],
"children": [
    {
        ...
    },
    ...
],
"errorMessage": "string"
},
```

```
"status": "ok",
"message": "string"
}
```

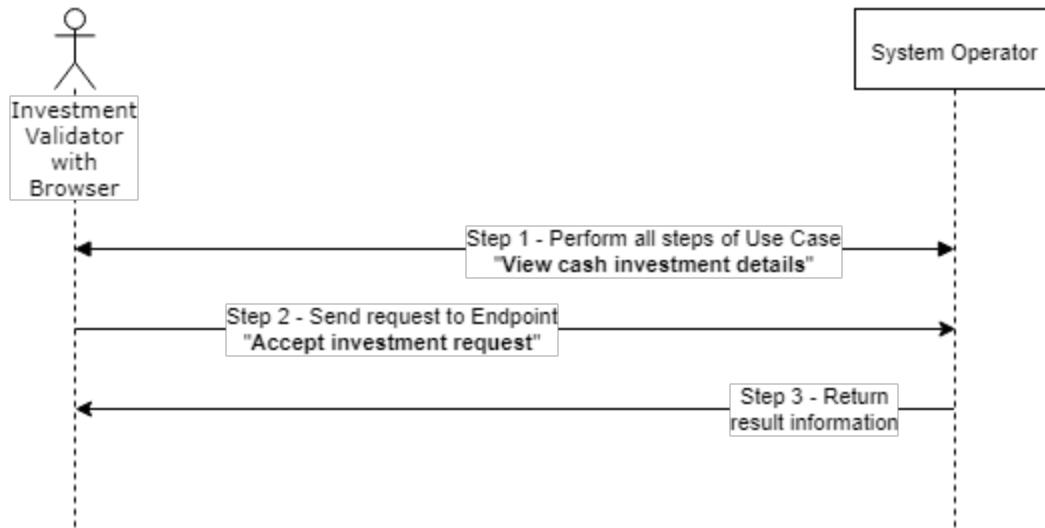
Accept investment request scheme

Use case: Accept investment request

Basic FFlow



Optional Web UI Flow



Top up of system provider coin description

Use Case Name

Top up of system provider coin

Brief Description

A User or External Entity on behalf of a User with role permission PROVIDER_INVESTMENT_EXECUTOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint

“Top up of system provider coin”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROVIDER_INVESTMENT_EXECUTOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Top up of system provider coin”.

Endpoint URL: POST /gate-investments

Parameters:

```
{
  "amount": 0,
  "coin-serial": "string",
  "fullName": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Top up of system provider coin”.

Endpoint URL: POST /gate-investments

Parameters:

```
{
  "amount": 0,
  "coin-serial": "string",
  "fullName": "string"
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

Request identifier is available for further processing.

Result example

```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-08-10T14:07:38.931Z",  
    "updatedAt": "2018-08-10T14:07:38.931Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "amount": 0,  
        "performedAt": "2018-08-10T14:07:38.932Z",  
        "issuer": {  
          "id": "string",  
          "sn": "string",  
          "currency": "string"  
        }  
      }  
    ],  
  }  
}
```

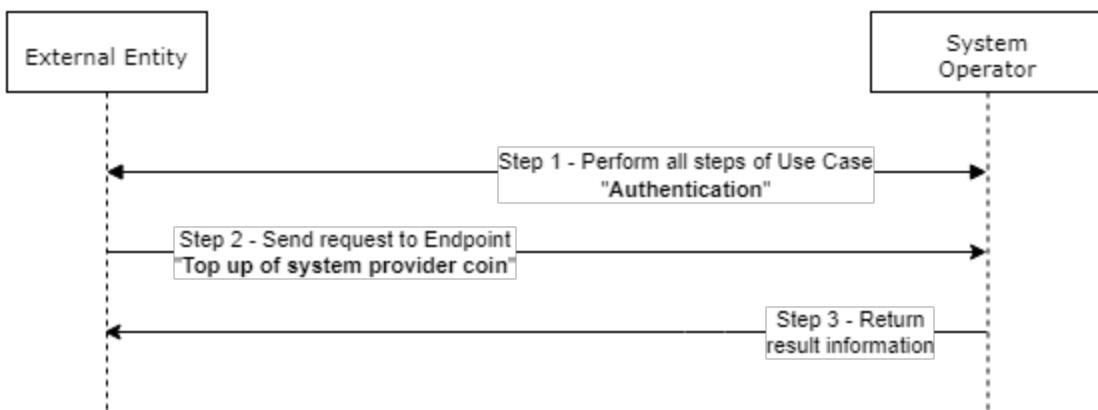
```
"children": [
  {
    }
  ],
  "errorMessage": "string"
},
```

```
"status": "ok",  
"message": "string"  
}
```

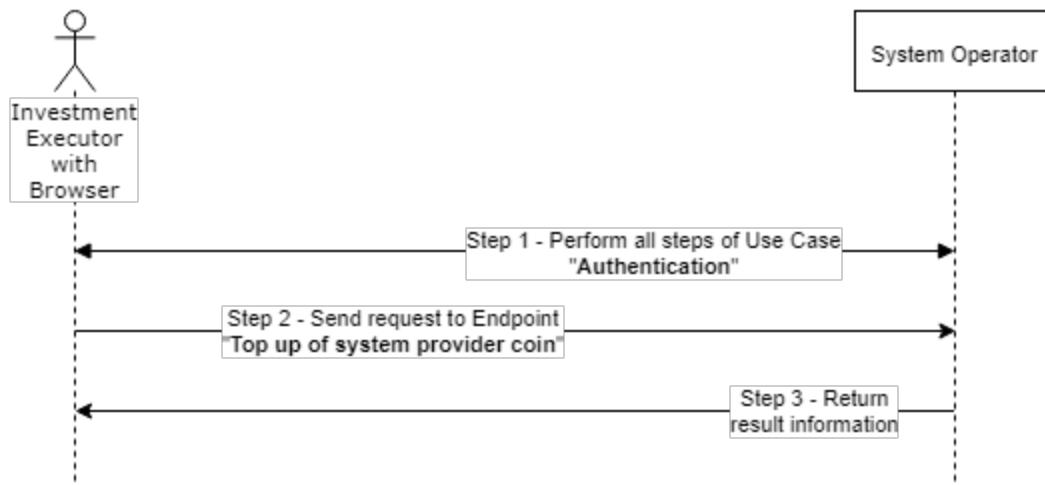
Top up of system provider coin scheme

Use case: Top up of system provider coin

Basic FFlow



Optional Web UI Flow



View cash investment details description

Use Case Name

View cash investment details

Brief Description

A User or External Entity on behalf of a User with role permission PROVIDER_INVESTMENT_VALIDATOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View cash investment details”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROVIDER_INVESTMENT_VALIDATOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View cash investment details”.

Endpoint URL:

GET /gate-investments/{requestIdentifier} - provided by PROVIDER_INVESTMENT_EXECUTOR

Parameters: TOKEN

3. System Operator returns cash investment details to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View cash investment details”.

Endpoint URL:

GET /gate-investments/{requestIdentifier} - provided by PROVIDER_INVESTMENT_EXECUTOR

Parameters: TOKEN

3. System Operator returns cash investment details to User (See Result example below).

Post Conditions

Cash investment details are available.

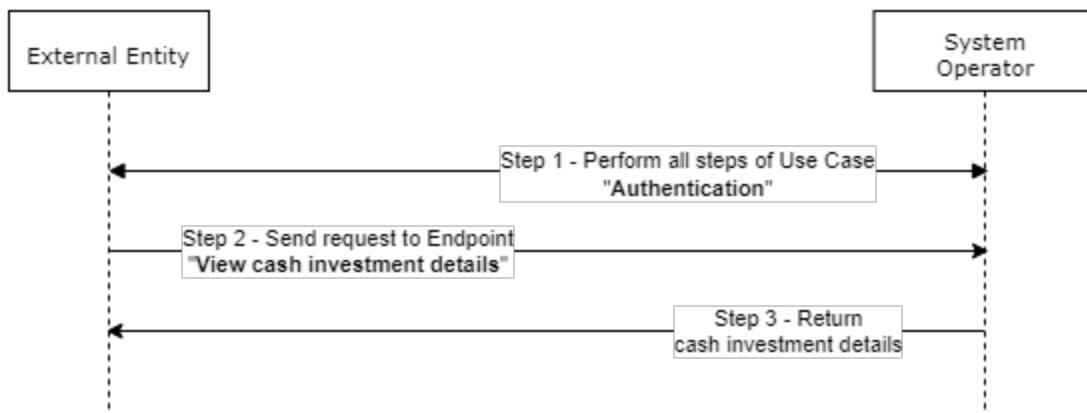
Result example

```
{
  "id": "string",
  "amount": 0,
  "issuerId": "string",
  "cashDeskSerial": "string",
  "fullName": "string",
  "processStatus": "string",
  "type": "string",
  "status": "ok",
  "message": "string"
}
```

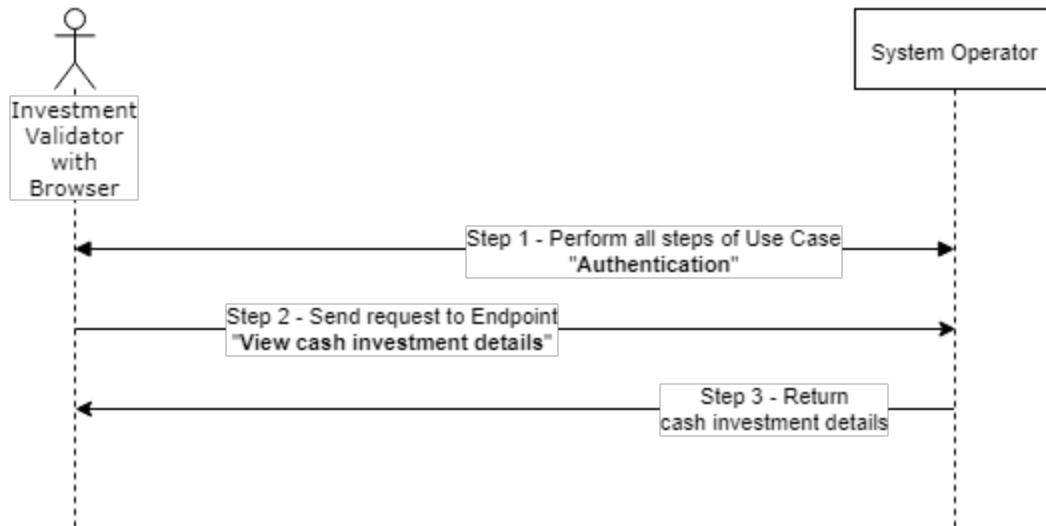
View cash investment details scheme

Use case: View cash investment details

Basic FFlow



Optional Web UI Flow



Gate - interaction with external payment systems

Filter transactions description

Use Case Name

Filter transactions

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Filter transactions".

Note: The filter is optional. When there are no filtering parameter available, the filter could be omitted to get all gate transactions.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Filter transactions".

Endpoint URL: POST /gate/transactions/view

Parameters: {

"pageNumber": 0,

"pageSize": 0,

"filter": {

"ids": [

"string"

],

"externalIds": [

"string"

],

"orderIds": [

0

```

],
"types": [
  "TOPUP"
],
"statuses": [
  "INITIATED"
],
"deviceIds": [
  "string"
],
"processIds": [
  "string"
]
},
"sort": {
  "date": "asc"
}
}
}

```

1. System Operator returns filtered list of transactions to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Filter transactions”.

Endpoint URL: POST /gate/transactions/view

```

Parameters: {
  "pageNumber": 0,
  "pageSize": 0,
  "filter": {
    "ids": [
      "string"
    ],
    "externalIds": [
      "string"
    ],
    "orderIds": [
      0
    ],
    "types": [
      "TOPUP"
    ]
}

```

```

    ],
    "statuses": [
        "INITIATED"
    ],
    "deviceIds": [
        "string"
    ],
    "processIds": [
        "string"
    ]
},
"sort": {
    "date": "asc"
}
}

```

1. System Operator returns filtered list of transactions to User (See Result example below).

Post Conditions

One or more transactions are available. Token is available.

Result example

```
{
    "records": [
        {
            "id": "string",
            "orderId": 0,
            "deviceId": "string",
            "deviceOrderId": "string",
            "type": "TOPUP",
            "status": "INITIATED",
            "errorCode": "UNKNOWN",
            "coin": {
                "serial": "string",
                "name": "string",
                "amount": 0,
                "availableAmount": 0,
                "issuer": {
                    "id": "string",

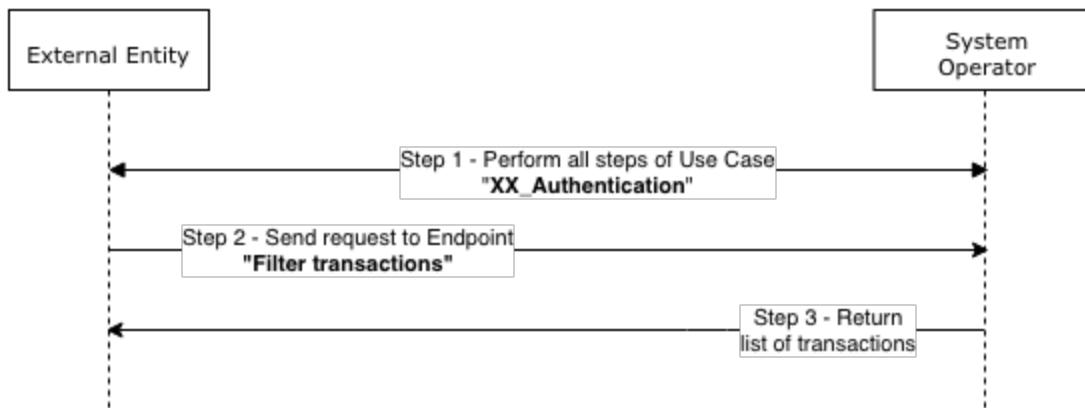
```

```
"sn": "string",
"currency": "string"
},
"active": false,
"type": "regular_commission"
},
"paymentMethod": {
"accountId": "string",
"account": {
"id": "string",
"provider": {
"name": "string"
}
},
"way": "string"
},
"sourceAmount": 0,
"amountToSend": 0,
"finalAmount": 0,
"processId": "string",
"payerData": {}
}
],
"status": "ok",
"message": "string",
"pageNumber": 0,
"pageSize": 0,
"totalRecords": 0,
"totalPages": 0
}
```

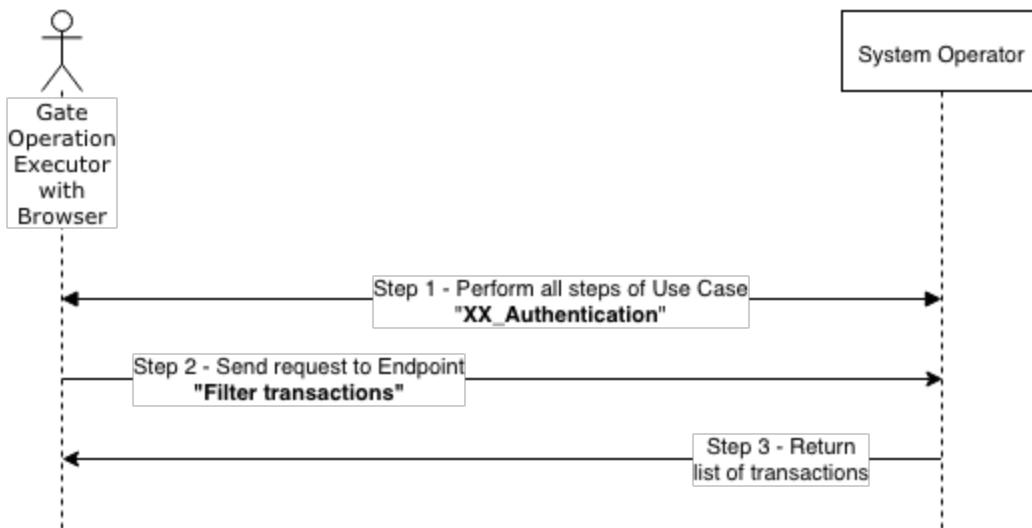
Filter transactions scheme

Use case: Filter transactions

Basic FFlow



Optional Web UI Flow



Step 1: retrieve list of payment providers for transaction with specified parameters description

Use Case Name

Step 1: retrieve list of payment providers for transaction with specified parameters

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Get coins owned by current user" Use Case (to obtain serialId), and then send a request to Endpoint "Step 1: retrieve list of payment providers for transaction with specified parameters".

Note: This request is used to view gate methods for a specified coin.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get coins owned by current user".
2. External Entity sends a request to Endpoint "Step 1: retrieve list of payment providers for transaction with specified parameters".

Endpoint URL: POST /gate/methods/view

Parameters: {

"txType": "TOPUP",

"serial": "string"

}

1. System Operator returns payment provider accounts to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get coins owned by current user".
2. A user sends a request to Endpoint "Step 1: retrieve list of payment providers for transaction with specified parameters".

Endpoint URL: POST /gate/methods/view

Parameters: {

"txType": "TOPUP",

"serial": "string"

```
}
```

1. System Operator returns list of payment provider accounts to User (See Result example below).

Post Conditions

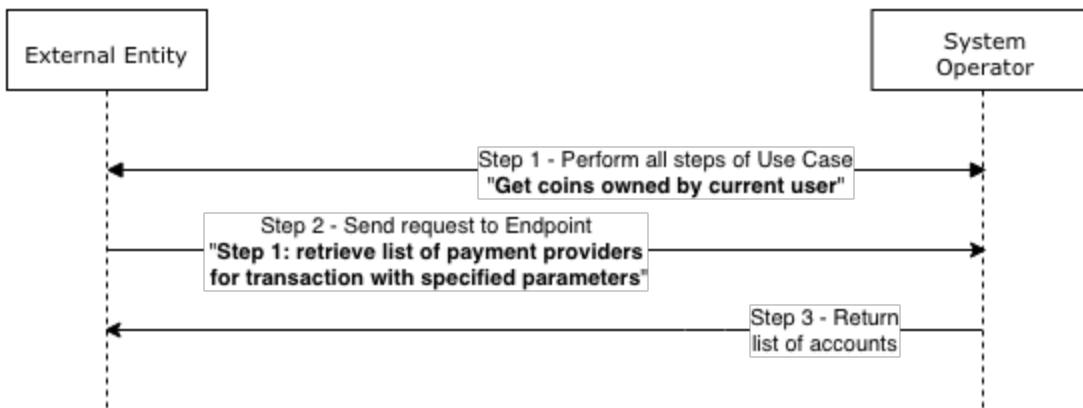
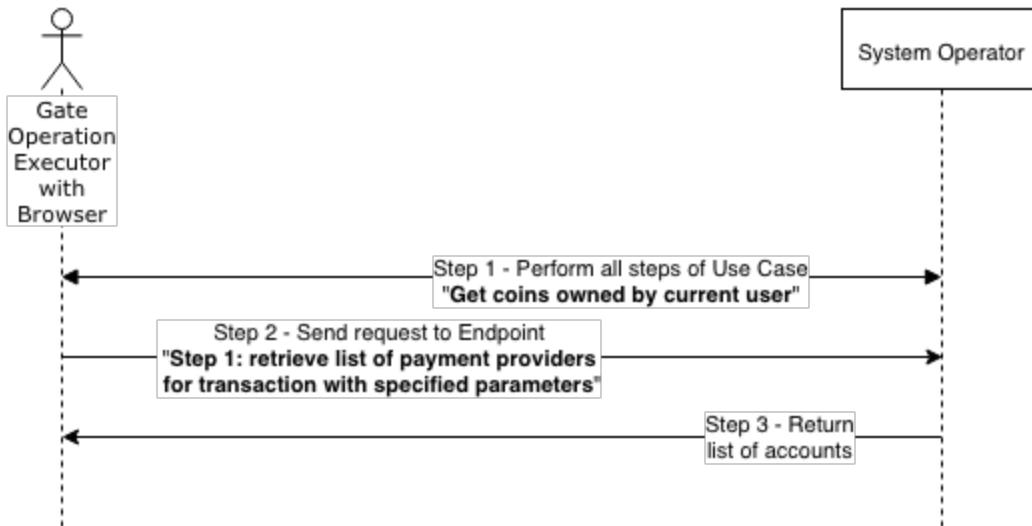
Payment provider is available. Token is available.

Result example

```
{
  "records": [
    {
      "accountId": "string",
      "account": {
        "id": "string",
        "provider": {
          "name": "string"
        }
      },
      "way": "string"
    }
  ],
  "status": "ok",
  "message": "string"
}
```

Step 1: retrieve list of payment providers for transaction with specified parameters scheme

Use case: Step 1: retrieve list of payment providers for transaction with specified parameters

Basic FFlow**Optional Web UI Flow**

Step 2: calculate commission for execution of transaction with specified parameters description

Use Case Name

Step 2: calculate commission for execution of transaction with specified parameters

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 1: retrieve list of payment providers for transaction with specified parameters" (to retrieve accountID and serialId) Use Case, and then send a request to Endpoint "Step 2: calculate commission for execution of transaction with specified parameters".

Note: "Step 1: retrieve list of payment providers for transaction with specified parameters" Use Case includes call to "Get coins owned by current user" Use Case to obtain serialId parameter.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 1: retrieve list of payment providers for transaction with specified parameters".
2. External Entity sends a request to Endpoint "Step 2: calculate commission for execution of transaction with specified parameters".

Endpoint URL: POST /gate/transactions/calculate

```
Parameters: {
  "accountId": "string",
  "serial": "string",
  "amount": 0,
  "txType": "TOPUP"
}
```

1. System Operator returns commission amount and total to be sent to payment provider, plus currency details, to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 1: retrieve list of payment providers for transaction with specified parameters".
2. A user sends a request to Endpoint "Step 2: calculate commission for execution of transaction with specified parameters".

Endpoint URL: POST /gate/transactions/calculate

```
Parameters: {  
    "accountId": "string",  
    "serial": "string",  
    "amount": 0,  
    "txType": "TOPUP"  
}
```

1. System Operator returns commission amount and total to be sent to payment provider, plus currency details, to User (See Result example below).

Post Conditions

One or more payment providers are available. Token is available.

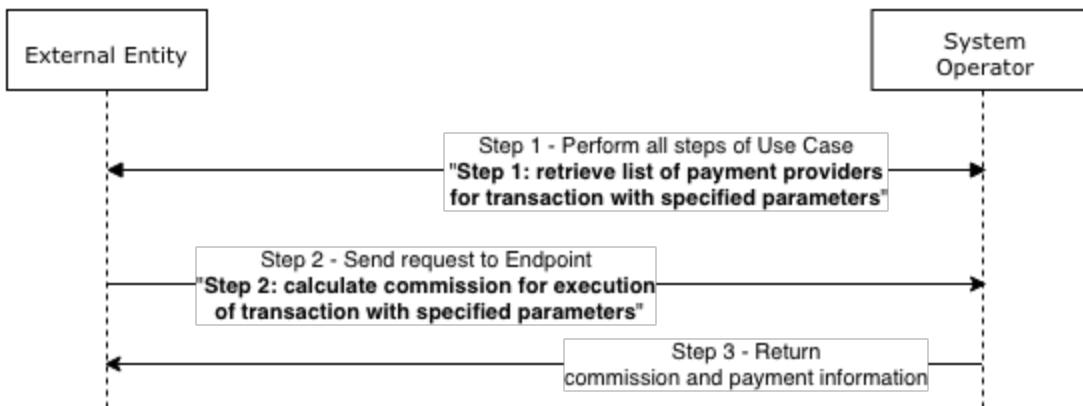
Result example

```
{  
    "sourceAmount": 0,  
    "amountToSend": 0,  
    "commissionAmount": 0,  
    "currency": {  
        "code": "string",  
        "digitalCode": "string",  
        "symbol": "string",  
        "name": "string",  
        "description": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

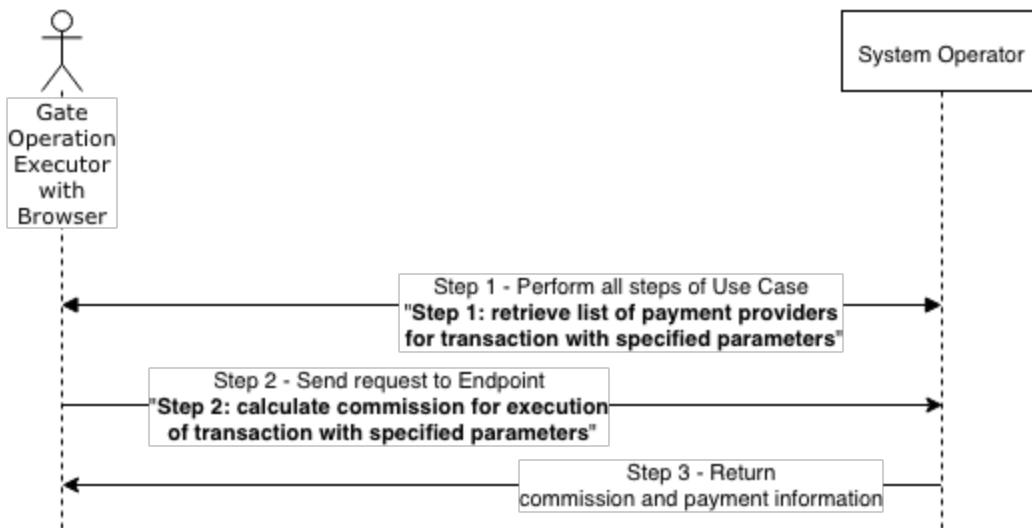
Step 2: calculate commission for execution of transaction with specified parameters scheme

Use case: Step 2: calculate commission for execution of transaction with specified parameters

Basic FLow



Optional Web UI Flow



Step 3: create a transaction description

Use Case Name

Step 3: create a transaction

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 1: retrieve list of payment providers for transaction with specified parameters" Use Case (to obtain method parameter), and then send a request to Endpoint "Step 3: create a transaction".

Note: deviceID is not tracked in the system. It is additional information, e.g. a PoS terminal, a store or a device ID.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 1: retrieve list of payment providers for transaction with specified parameters".
2. External Entity sends a request to Endpoint "Step 3: create a transaction".

Endpoint URL: POST /gate/transactions

Parameters: {

```
"coin": "string",
"amount": 1,
"type": "TOPUP",
"deviceId": "string",
"deviceOrderId": "string",
"method": {
    "accountId": "string",
    "way": "string"
}
```

1. System Operator returns transaction and payment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Step 1: retrieve list of payment providers for transaction with specified parameters”.
2. A user sends a request to Endpoint “Step 3: create a transaction”.

Endpoint URL: POST /gate/transactions

Parameters: {

```
"coin": "string",
"amount": 1,
"type": "TOPUP",
"deviceId": "string",
"deviceOrderId": "string",
"method": {
  "accountId": "string",
  "way": "string"
}
}
```

1. System Operator returns transaction and payment information to User (See Result example below).

Post Conditions

Payment provider is available.

Result example

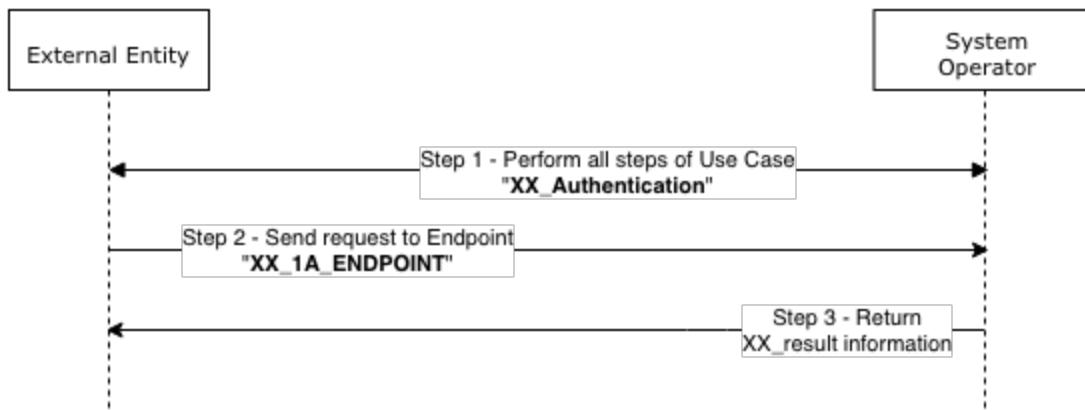
```
{
  "transaction": {
    "id": "string",
    "orderId": 0,
    "deviceId": "string",
    "deviceOrderId": "string",
    "type": "TOPUP",
    "status": "INITIATED",
    "errorCode": "UNKNOWN",
    "coin": {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "availableAmount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
      }
    }
  }
}
```

```
"currency": "string"
},
"active": false,
"type": "regular_commission"
},
"paymentMethod": {
"accountId": "string",
"account": {
"id": "string",
"provider": {
"name": "string"
}
},
"way": "string"
},
"sourceAmount": 0,
"amountToSend": 0,
"finalAmount": 0,
"processId": "string",
"payerData": {}
},
"status": "ok",
"message": "string"
}
```

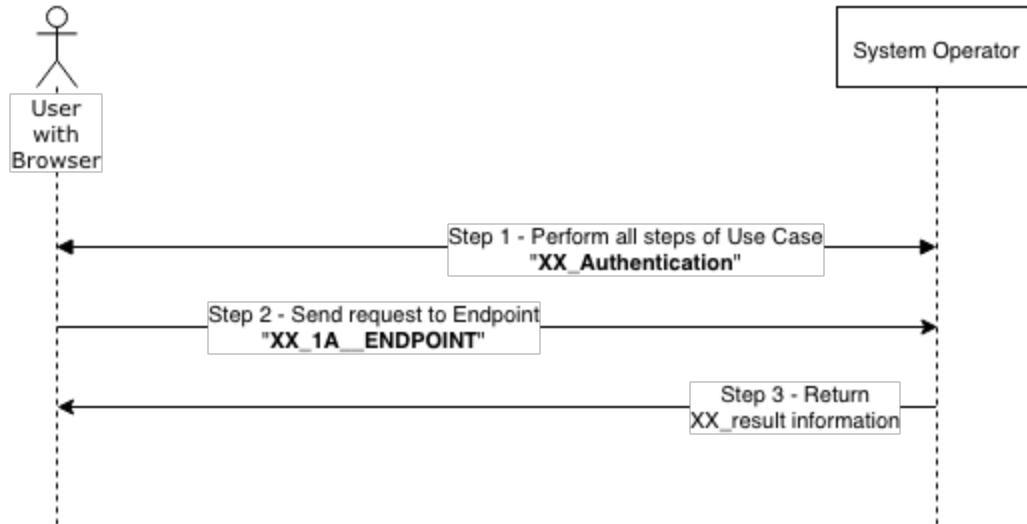
Step 3: create a transaction scheme

Use case: XX_1A_ENDPOINT

Basic FFlow



Optional Web UI Flow



Step 4: retrieve list of payer fields request by payment provider description

Use Case Name

Step 4: retrieve list of payer fields request by payment provider

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 3: create a transaction" (to obtain transaction ID parameter) Use Case, and then send a request to Endpoint "Step 4: retrieve list of payer fields request by payment provider".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 3: create a transaction".
2. External Entity sends a request to Endpoint "Step 4: retrieve list of payer fields request by payment provider".

Endpoint URL: GET /gate/transactions/{tx}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of payer fields to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 3: create a transaction".
2. A user sends a request to Endpoint "Step 4: retrieve list of payer fields request by payment provider".

Endpoint URL: GET /gate/transactions/{tx}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of payer fields to User (See Result example below).

Post Conditions

Transaction is available.

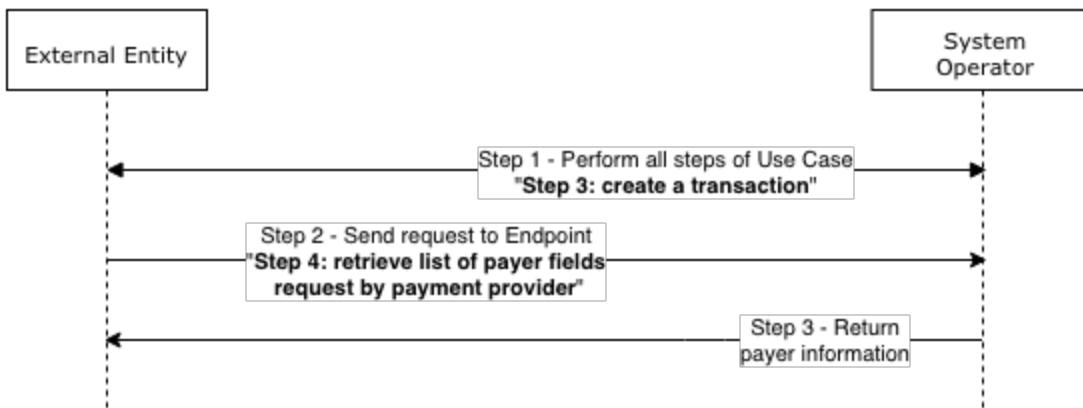
Result example

```
{  
  "options": [  
    {  
      "name": "default",  
      "fields": [  
        {  
          "name": "string",  
          "optional": false,  
          "label": "string",  
          "hint": "string",  
          "constraints": [  
            "string"  
          ]  
        }  
      ]  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

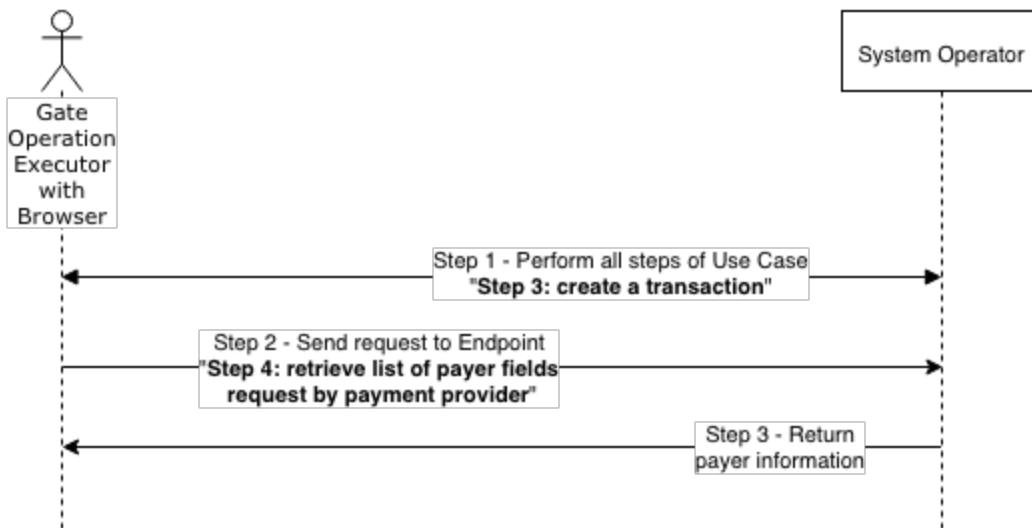
Step 4: retrieve list of payer fields request by payment provider scheme

**Use case: Step 4: retrieve list of payer fields
request by payment provider**

Basic FLow



Optional Web UI Flow



Step 5: submit payer data and send gate transaction to payment provider description

Use Case Name

Step 5: submit payer data and send gate transaction to payment provider

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 4: retrieve list of payer fields request by payment provider" Use Case, and then send a request to Endpoint "Step 5: submit payer data and send gate transaction to payment provider".

Note: Transaction ID is obtained from "Step 3: create a transaction" Use Case. Payer data parameters are obtained from "Step 4: retrieve list of payer fields request by payment provider" Use Case,

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 4: retrieve list of payer fields request by payment provider".
2. External Entity sends a request to Endpoint "Step 5: submit payer data and send gate transaction to payment provider".

Endpoint URL: POST /gate/transactions/{tx}/submit

```
Parameters: {
  "optionName": "default",
  "fields": [
    {
      "name": "string",
      "value": {}
    }
  ]
}
```

1. System Operator returns gate transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Step 4: retrieve list of payer fields request by payment provider”.
2. A user sends a request to Endpoint “Step 5: submit payer data and send gate transaction to payment provider”.

Endpoint URL: POST /gate/transactions/{tx}/submit

```
Parameters: {
  "optionName": "default",
  "fields": [
    {
      "name": "string",
      "value": {}
    }
  ]
}
```

1. System Operator returns gate transaction information to User (See Result example below).

Post Conditions

Transaction is available.

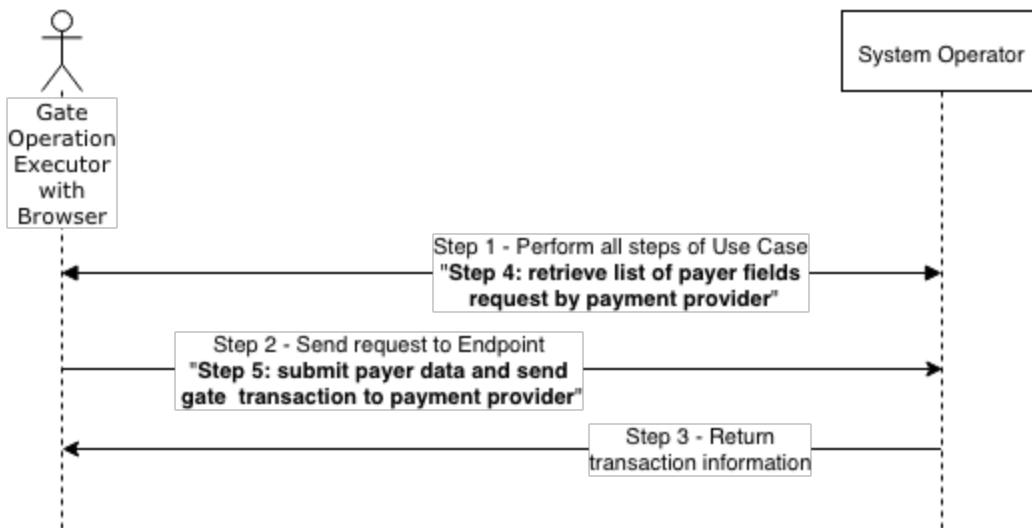
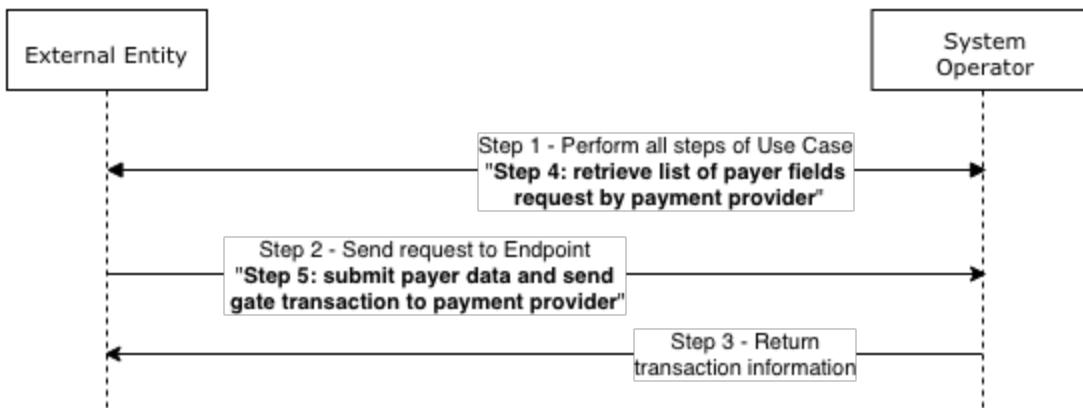
Result example

```
{
  "form": {
    "url": "string",
    "method": "GET",
    "parameters": {}
  },
  "status": "ok",
  "message": "string",
  "transaction": {
    "id": "string",
    "orderId": 0,
    "deviceId": "string",
    "deviceOrderId": "string",
    "type": "TOPUP",
    "status": "INITIATED",
    "errorCode": "UNKNOWN",
    "coin": {
      "serial": "string",
      "name": "string",
      "amount": 0
    }
  }
}
```

```
"availableAmount": 0,  
  "issuer": {  
    "id": "string",  
    "sn": "string",  
    "currency": "string"  
  },  
  "active": false,  
  "type": "regular_commission"  
},  
  "paymentMethod": {  
    "accountId": "string",  
    "account": {  
      "id": "string",  
      "provider": {  
        "name": "string"  
      }  
    },  
    "way": "string"  
},  
  "sourceAmount": 0,  
  "amountToSend": 0,  
  "finalAmount": 0,  
  "processId": "string",  
  "payerData": {}  
}  
}
```

Step 5: submit payer data and send gate transaction to payment provider scheme

Use case: Step 5: submit payer data and send gate transaction to payment provider



Step 6: get transaction state description

Use Case Name

Step 6: get transaction state

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 3: create a transaction" (to obtain transaction ID) Use Case, and then send a request to Endpoint "Step 6: get transaction state".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 3: create a transaction".
2. External Entity sends a request to Endpoint "Step 6: get transaction state".

Endpoint URL: GET /gate/transactions/{tx}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns transaction information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 3: create a transaction".
2. A user sends a request to Endpoint "Step 6: get transaction state".

Endpoint URL: GET /gate/transactions/{tx}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns transaction information to User (See Result example below).

Post Conditions

Transaction is available.

Result example

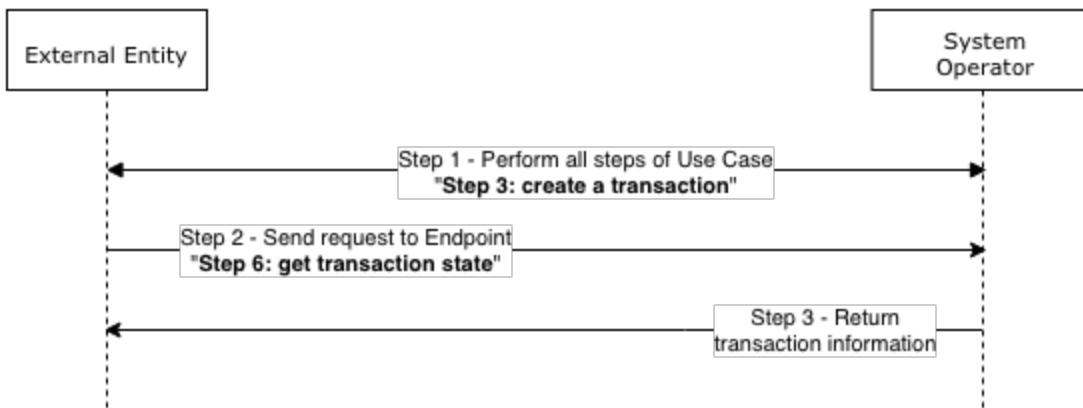
```
{  
  "transaction": {  
    "id": "string",  
    "orderId": 0,  
    "deviceId": "string",  
    "deviceOrderId": "string",  
    "type": "TOPUP",  
    "status": "INITIATED",  
    "errorCode": "UNKNOWN",  
    "coin": {  
      "serial": "string",  
      "name": "string",  
      "amount": 0,  
      "availableAmount": 0,  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "active": false,  
      "type": "regular_commission"  
    },  
    "paymentMethod": {  
      "accountId": "string",  
      "account": {  
        "id": "string",  
        "provider": {  
          "name": "string"  
        }  
      },  
      "way": "string"  
    },  
    "sourceAmount": 0,  
    "amountToSend": 0,  
    "finalAmount": 0,  
    "processId": "string",  
    "payerData": {}  
  },  
  "status": "ok",  
  "message": "string"
```

}

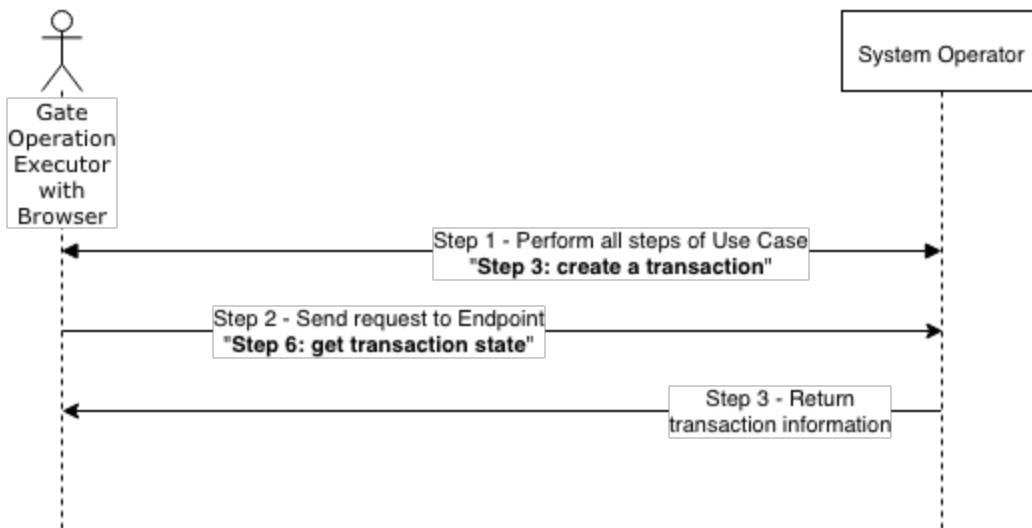
Step 6: get transaction state scheme

Use case: Step 6: get transaction state

Basic FFlow



Optional Web UI Flow



Gate contract management

Create a new exchange settings record description

Use Case Name

Create a new exchange settings record

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain issuers" and "View gate commission profiles for specified contract" Use Cases, and then send a request to Endpoint "Create a new exchange settings record". The first Use Case obtains the issuerID, and the second the contractId and profileId. These parameters are passed to Endpoint "Create a new exchange settings record".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain issuers".
2. Perform all steps of Use Case "View gate commission profiles for specified contract".
3. External Entity sends a request to Endpoint "Create a new exchange settings record".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/exchange-settings-records

Parameters: {

```
"issuerId": "string",
"rate": 1
}
```

1. System Operator returns details of new exchange settings record to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Obtain issuers".
2. Perform all steps of Use Case "View gate commission profiles for specified contract".
3. A User sends a request to Endpoint "Create a new exchange settings record".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/exchange-settings-records

Parameters: {

```
"issuerId": "string",
```

```
"rate": 1  
}
```

1. System Operator returns details of new exchange settings record to User (See Result example below).

Post Conditions

Issuer and gate commission profile for specified contract are available.

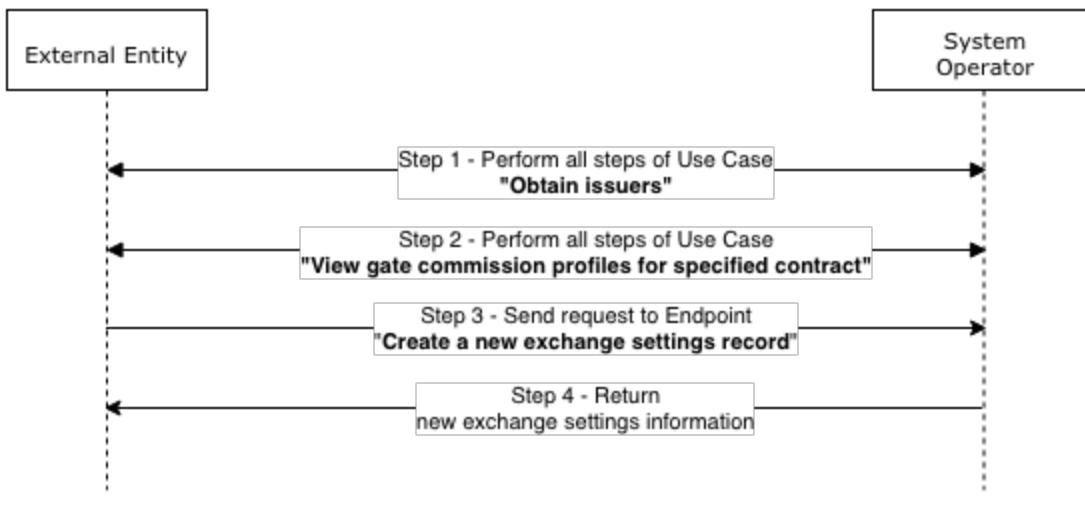
Result example

```
{  
  "record": {  
    "id": "string",  
    "issuer": {  
      "id": "string",  
      "sn": "string",  
      "currency": "string"  
    },  
    "rate": 0  
  },  
  "status": "ok",  
  "message": "string"  
}
```

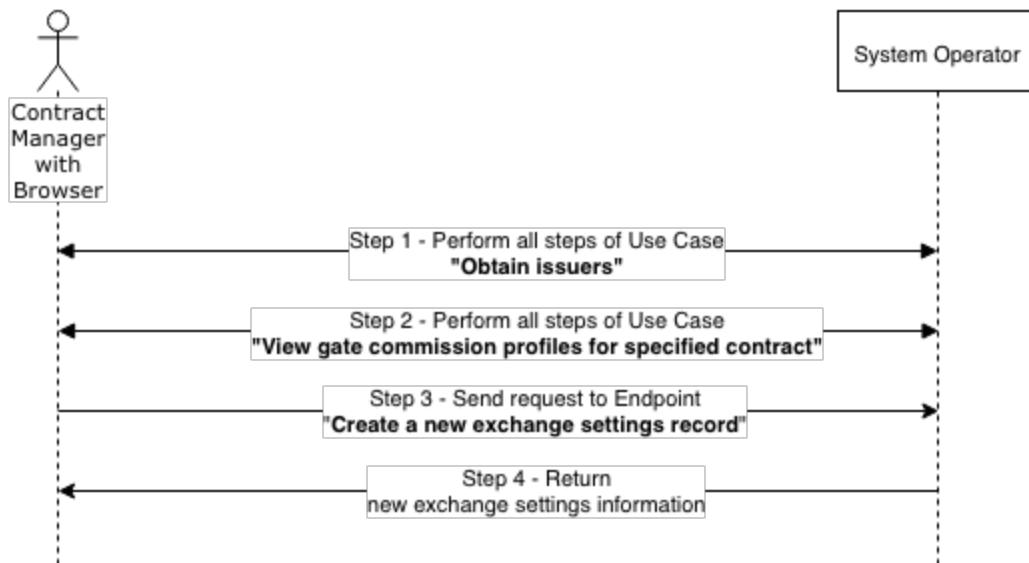
Create a new exchange settings record scheme

Use case: Create a new exchange settings record

Basic FFlow



Optional Web UI Flow



Create a new gate commission profile description

Use Case Name

Create a new gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" Use Case, and then send a request to Endpoint "Create a new gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "Create a new gate commission profile".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles

Parameters: {

```
"providerAccountId": "string",
"currencyCode": "string"
}
```

1. System Operator returns details of new gate commission profile added to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. A user sends a request to Endpoint "Create a new gate commission profile".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles

Parameters: {

```
"providerAccountId": "string",
"currencyCode": "string"
}
```

1. System Operator returns details of new gate commission profile added to User (See Result example below).

Post Conditions

Gate commission profile for specified contract is available.

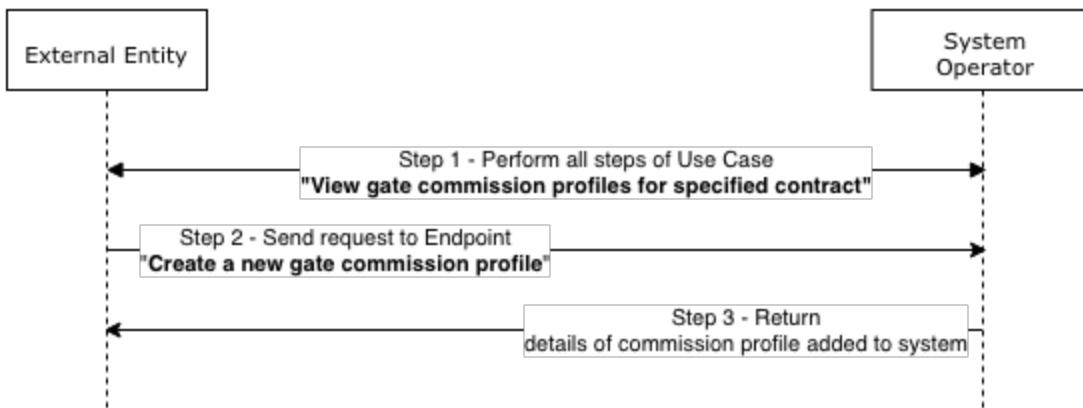
Result example

```
{  
  "profile": {  
    "id": "string",  
    "providerAccountId": "string",  
    "providerCurrency": {  
      "code": "string",  
      "digitalCode": "string",  
      "symbol": "string",  
      "name": "string",  
      "description": "string"  
    },  
    "createdAt": "2018-08-29T10:07:45.892Z",  
    "updatedAt": "2018-08-29T10:07:45.892Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

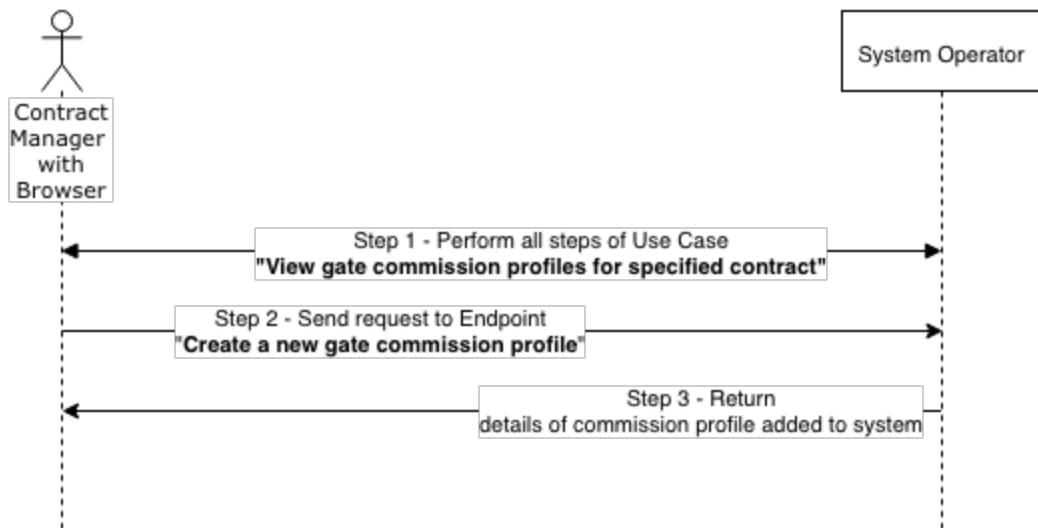
Create a new gate commission profile scheme

Use case: Create a new gate commission profile

Basic FFlow



Optional Web UI Flow



Create a new gate limit profile description

Use Case Name

Create a new gate limit profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain gate limit profiles for specified gate commission profile" Use Case, and then send a request to Endpoint "Create a new gate limit profile". A productId can be obtained from the "Step 1: get list of all available products" Use Case. A User may have 0 or more contracts, to each of which may be linked 0 or more commission profiles, to each of which may be linked to 0 or more limit profiles.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. External Entity sends a request to Endpoint "Create a new gate limit profile".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles

```
Parameters: {
  "txType": "TOPUP",
  "productId": "string",
  "qualifier": "amount",
  "timeUnit": "per_transaction",
  "value": 0,
  "active": false
}
```

1. System Operator returns new gate limit profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. A user sends a request to Endpoint "Create a new gate limit profile".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles

```
Parameters: {
  "txType": "TOPUP",
  "productId": "string",
  "qualifier": "amount",
  "timeUnit": "per_transaction",
  "value": 0,
  "active": false
}
```

1. System Operator returns new gate limit profile information to User (See Result example below).

Post Conditions

User contract is available.

Result example

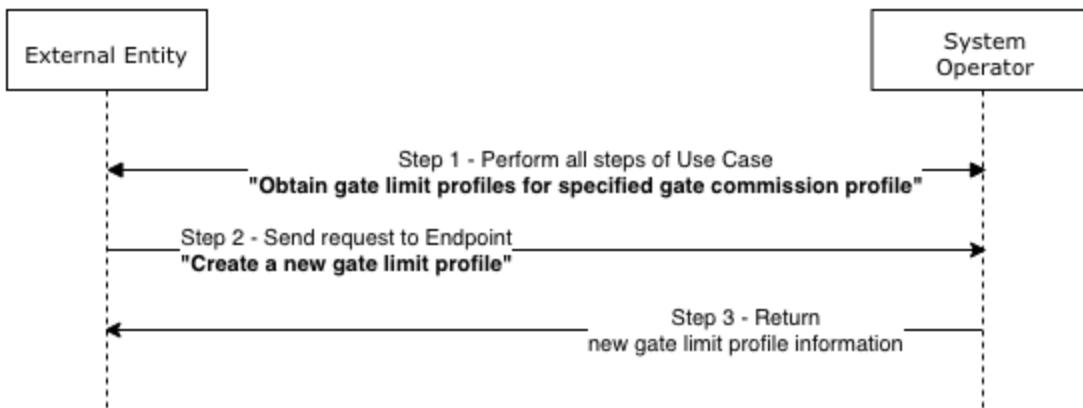
```
{
  "profile": {
    "id": "string",
    "gateProfileId": "string",
    "txType": "string",
    "product": {
      "id": "string",
      "name": "string",
      "description": "string",
      "category": "MOBILE",
      "icon": "string",
      "accountId": "string",
      "currencies": [
        {
          "currency": {
            "code": "string",
            "digitalCode": "string",
            "symbol": "string",
            "name": "string",
            "description": "string"
          },
          "minAmount": 0,
          "maxAmount": 0
        }
      ]
    }
  }
}
```

```
        ],
    },
    "qualifier": "amount",
    "timeUnit": "per_transaction",
    "value": 0,
    "active": false
},
"status": "ok",
"message": "string"
```

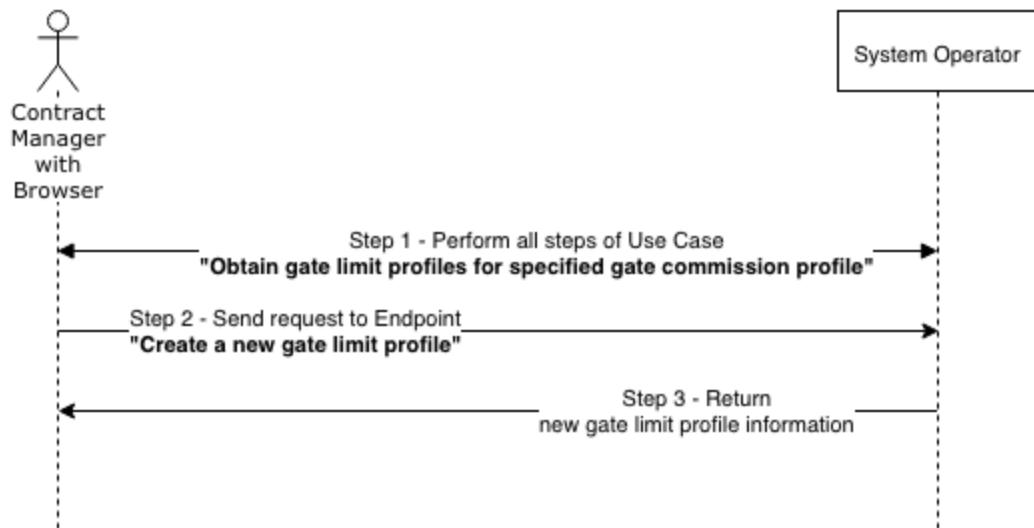
Create a new gate limit profile scheme

Use case: Create a new gate limit profile

Basic FFlow



Optional Web UI Flow



Obtain gate limit profiles for specified gate commission profile description

Use Case Name

Obtain gate limit profiles for specified gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" Use Case, and then send a request to Endpoint "Obtain gate limit profiles for specified gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "Obtain gate limit profiles for specified gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate limit profiles for a commission profile to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. A user sends a request to Endpoint "Obtain gate limit profiles for specified gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate limit profiles for a commission profile to User (See Result example below).

Post Conditions

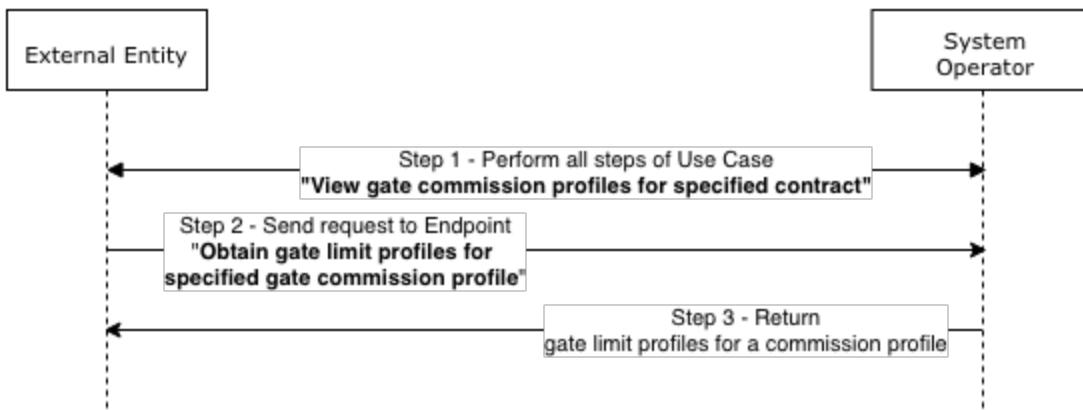
Gate commission profile is available.

Result example

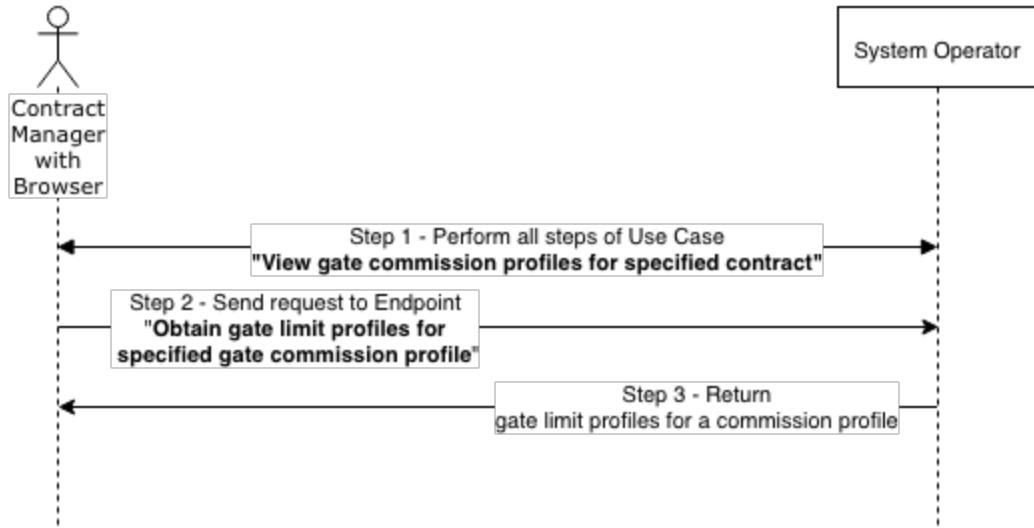
```
{  
  "records": [  
    {  
      "id": "string",  
      "gateProfileId": "string",  
      "txType": "string",  
      "product": {  
        "id": "string",  
        "name": "string",  
        "description": "string",  
        "category": "MOBILE",  
        "icon": "string",  
        "accountId": "string",  
        "currencies": [  
          {  
            "currency": {  
              "code": "string",  
              "digitalCode": "string",  
              "symbol": "string",  
              "name": "string",  
              "description": "string"  
            },  
            "minAmount": 0,  
            "maxAmount": 0  
          }  
        ]  
      },  
      "qualifier": "amount",  
      "timeUnit": "per_transaction",  
      "value": 0,  
      "active": false  
    },  
    {"status": "ok",  
     "message": "string"}  
  ]  
}  
Obtain gate limit profiles for specified gate commission profile scheme
```

Use case: Obtain gate limit profiles for specified gate commission profile

Basic FLow



Optional Web UI Flow



Set up a commission settings description

Use Case Name

Set up a commission settings

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain gate limit profiles for specified gate commission profile" Use Case, and then send a request to Endpoint "Set up a commission settings".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. External Entity sends a request to Endpoint "Set up a commission settings".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/commission-settings-records/set-up-commission-settings

Parameters: {

```
"txType": "TOPUP",
"providerCommission": {
    "type": "zero",
    "valuePercent": 0,
    "valueFixed": 0
},
"totalCommission": {
    "type": "zero",
    "valuePercent": 0,
    "valueFixed": 0
},
"active": false
}
```

1. System Operator returns new commission settings information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Obtain gate limit profiles for specified gate commission profile”.
2. A user sends a request to Endpoint “Set up a commission settings”.

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/commission-settings-records/set-up-commission-settings

Parameters: {

```
"txType": "TOPUP",
"providerCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"totalCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"active": false
}
```

1. System Operator returns new commission settings information to User (See Result example below).

Post Conditions

A gate limit profile for a specified gate commission profile is available.

Result example

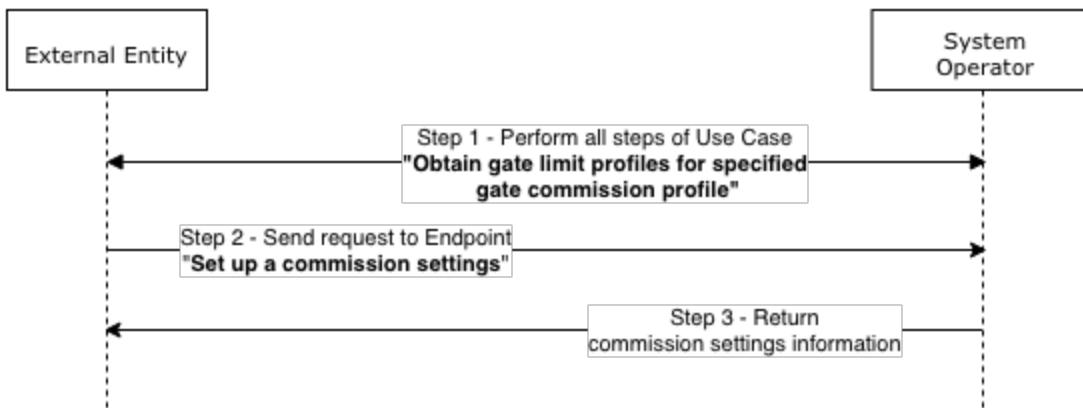
```
{
"records": [
{
  "id": "string",
  "txType": "TOPUP",
  "collector": "BANK",
  "active": false,
  "value": {
    "type": "zero",
    "valuePercent": 0,
    "valueFixed": 0
  }
}]}
```

```
],  
  "status": "ok",  
  "message": "string"  
}
```

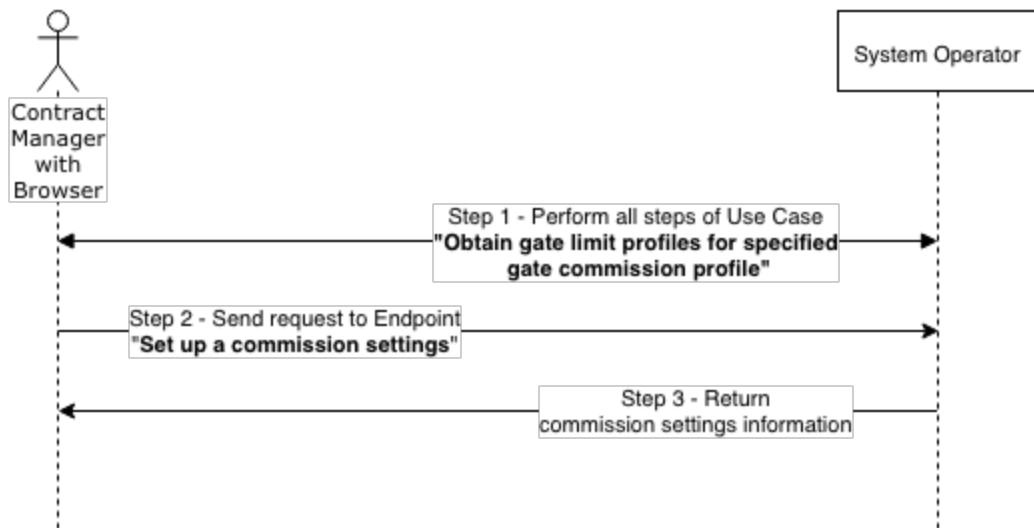
Set up a commission settings scheme

Use case: Set up a commission settings

Basic FLow



Optional Web UI Flow



Set up a product commission settings description

Use Case Name

Set up a product commission settings

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" Use Case, and then send a request to Endpoint "Set up a product commission settings".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "Set up a product commission settings".

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/product-commission-settings-records/set-up-product-commission-settings

Parameters: {

```
"productId": "string",
"providerCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"totalCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"active": false
}
```

1. System Operator returns product commission settings for a profile to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View gate commission profiles for specified contract”.
2. A user sends a request to Endpoint “Set up a product commission settings”.

Endpoint URL: POST /contracts/{contractId}/gate-commission-profiles/{profileId}/product-commission-settings-records/set-up-product-commission-settings

Parameters: {

```
"productId": "string",
"providerCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"totalCommission": {
  "type": "zero",
  "valuePercent": 0,
  "valueFixed": 0
},
"active": false
}
```

1. System Operator returns product commission settings for a profile to User (See Result example below).

Post Conditions

Gate commission profile for a contract is available.

Result example

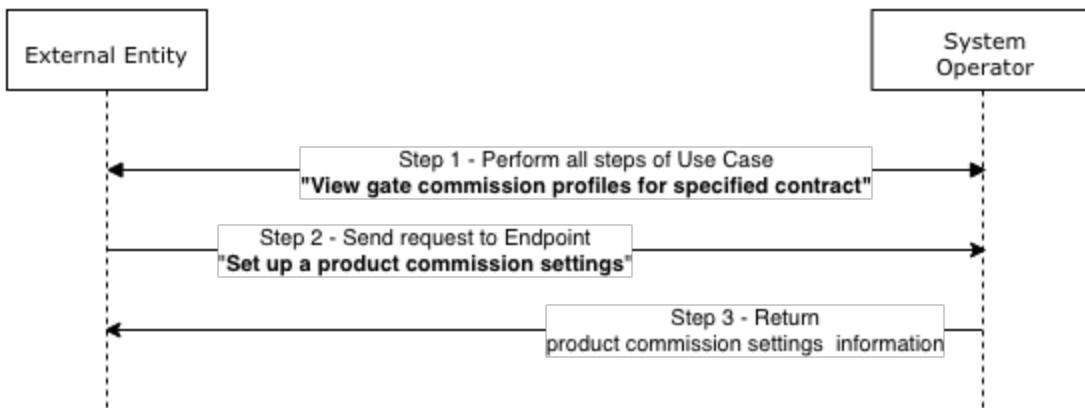
```
{
"records": [
{
  "id": "string",
  "product": {
    "id": "string",
    "name": "string",
    "description": "string",
    "category": "MOBILE",
    "icon": "string",
    "accountId": "string",
    "currencies": [

```

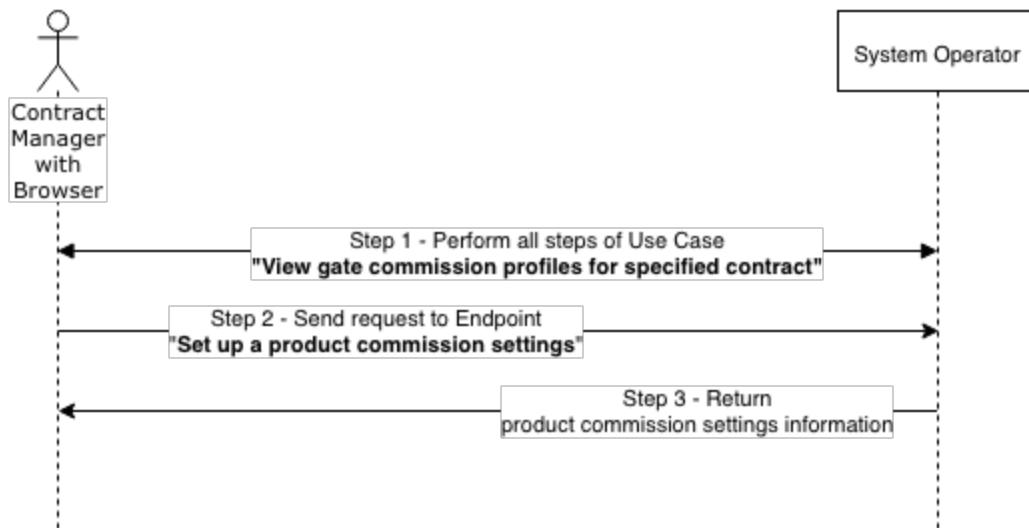
```
{  
    "currency": {  
        "code": "string",  
        "digitalCode": "string",  
        "symbol": "string",  
        "name": "string",  
        "description": "string"  
    },  
    "minAmount": 0,  
    "maxAmount": 0  
},  
]  
},  
"collector": "BANK",  
"active": false,  
"value": {  
    "type": "zero",  
    "valuePercent": 0,  
    "valueFixed": 0  
}  
}  
],  
"status": "ok",  
"message": "string"  
}  
Set up a product commission settings scheme
```

Use case: Set up a product commission settings

Basic FLow



Optional Web UI Flow



Update an existing limit profile 1 description

Use Case Name

Update an existing limit profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain gate limit profiles for specified gate commission profile" Use Case, and then send a request to Endpoint "Update an existing limit profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. External Entity sends a request to Endpoint "Update an existing limit profile".

Endpoint URL: PATCH /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles/{limitProfileId}

```
Parameters: {
  "value": 0,
  "active": false
}
```

1. System Operator returns updated limit profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. A user sends a request to Endpoint "Update an existing limit profile".

Endpoint URL: PATCH /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles/{limitProfileId}

```
Parameters: {
  "value": 0,
  "active": false
}
```

1. System Operator returns updated limit profile information to User (See Result example below).

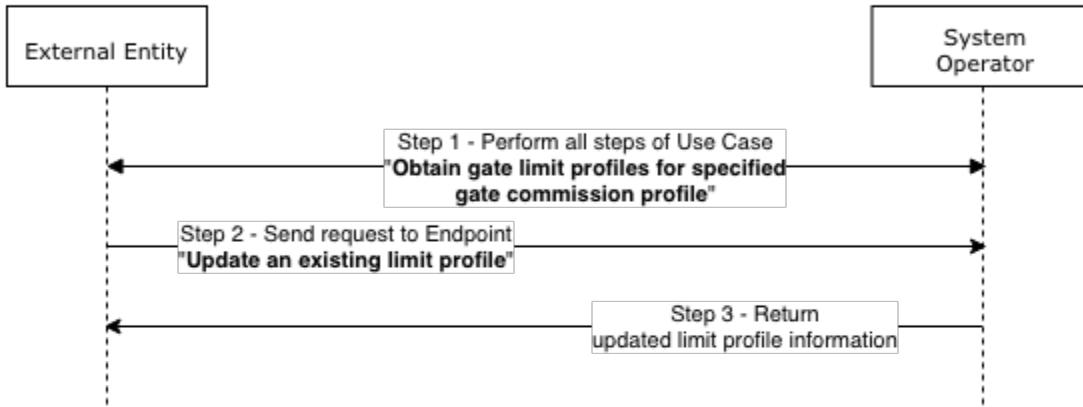
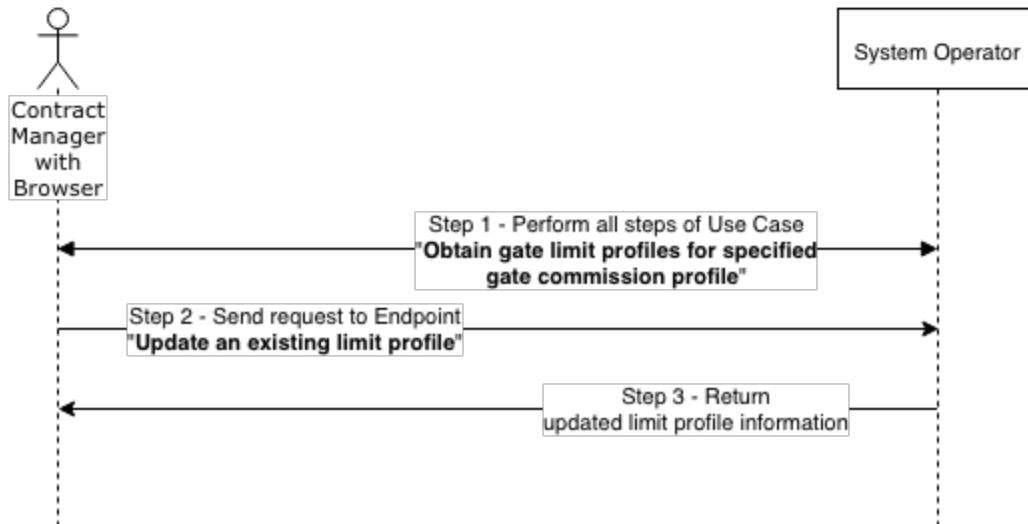
Post Conditions

Limit profile is available.

Result example

```
{  
  "profile": {  
    "id": "string",  
    "gateProfileId": "string",  
    "txType": "string",  
    "product": {  
      "id": "string",  
      "name": "string",  
      "description": "string",  
      "category": "MOBILE",  
      "icon": "string",  
      "accountId": "string",  
      "currencies": [  
        {  
          "currency": {  
            "code": "string",  
            "digitalCode": "string",  
            "symbol": "string",  
            "name": "string",  
            "description": "string"  
          },  
          "minAmount": 0,  
          "maxAmount": 0  
        }  
      ],  
      "qualifier": "amount",  
      "timeUnit": "per_transaction",  
      "value": 0,  
      "active": false  
    },  
    "status": "ok",  
    "message": "string"  
  }  
}
```

Update an existing limit profile 1 scheme

Use case: Update an existing limit profile**Basic FLow****Optional Web UI Flow**

View commission settings of gate commission profile description

Use Case Name

View commission settings of gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" Use Case, and then send a request to Endpoint "View commission settings of gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "View commission settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/commission-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns commission settings of gate commission profile result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. A user sends a request to Endpoint "View commission settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/commission-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns commission settings of gate commission profile to User (See Result example below).

Post Conditions

Gate commission profile is available.

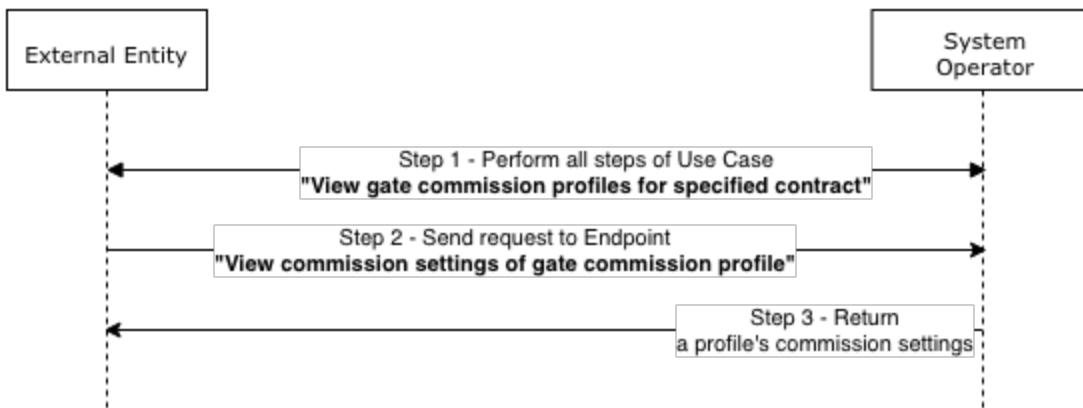
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "txType": "TOPUP",  
      "collector": "BANK",  
      "active": false,  
      "value": {  
        "type": "zero",  
        "valuePercent": 0,  
        "valueFixed": 0  
      }  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

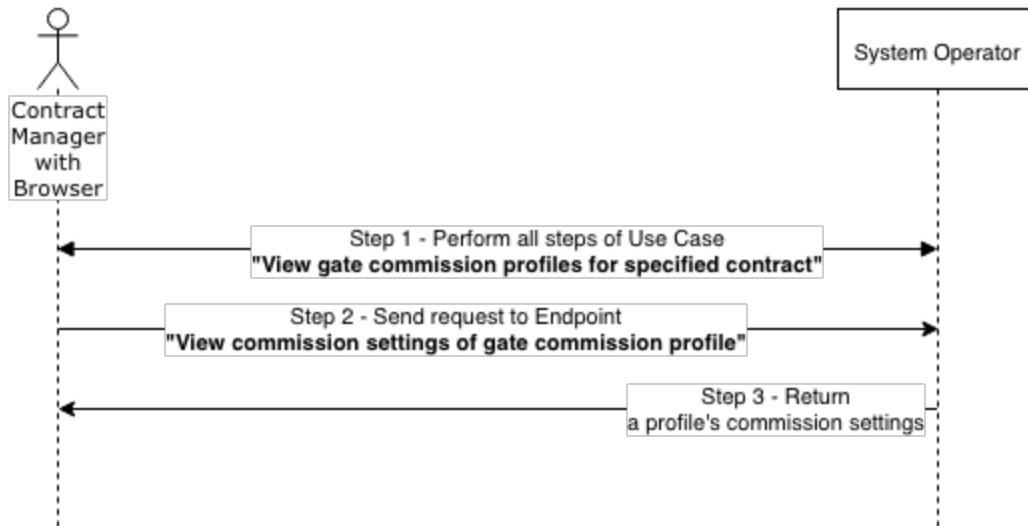
View commission settings of gate commission profile scheme

Use case: View commission settings of gate commission profile

Basic FLow



Optional Web UI Flow



View exchange settings of gate commission profile description

Use Case Name

View exchange settings of gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profile" (to obtain profileID) Use Case, and then send a request to Endpoint "View exchange settings of gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profile".
2. External Entity sends a request to Endpoint "View exchange settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/exchange-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns commission profile's exchange settings to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profile".
2. A user sends a request to Endpoint "View exchange settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/exchange-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns commission profile's exchange settings to User (See Result example below).

Post Conditions

Gate commission profile is available.

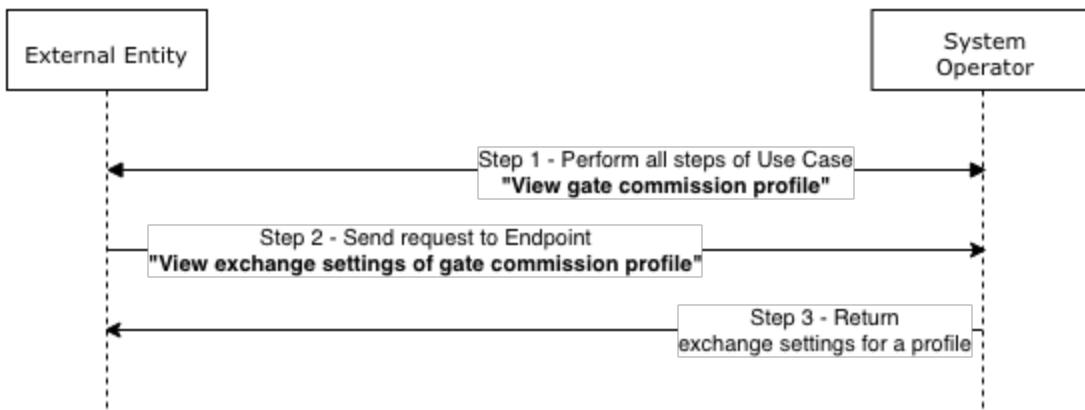
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "rate": 0  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

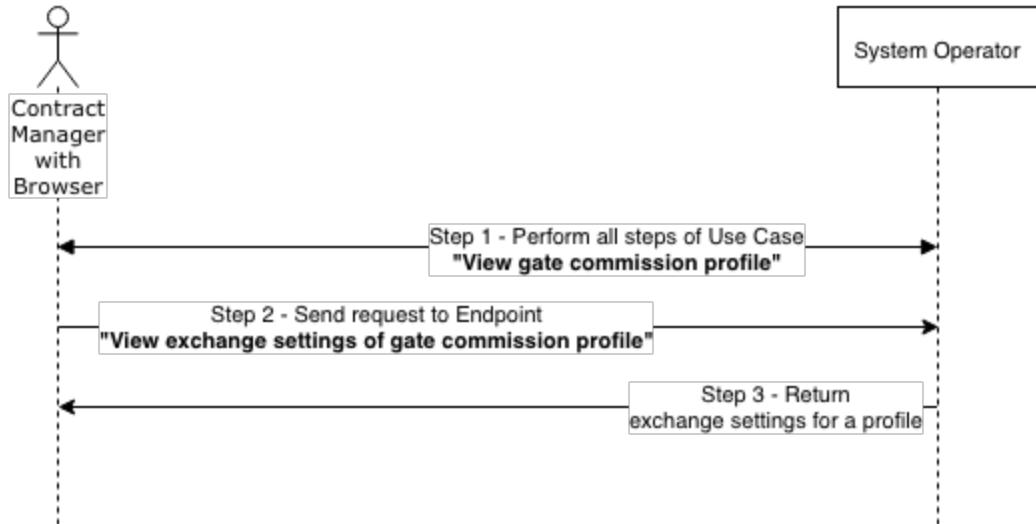
View exchange settings of gate commission profile scheme

Use case: View exchange settings of gate commission profile

Basic FLow



Optional Web UI Flow



View gate commission profile description

Use Case Name

View gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" (to obtain profileID and contractID) Use Case, and then send a request to Endpoint "View gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "View gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate commission profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. A user sends a request to Endpoint "View gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate commission profile information to User (See Result example below).

Post Conditions

Gate commission profile is available.

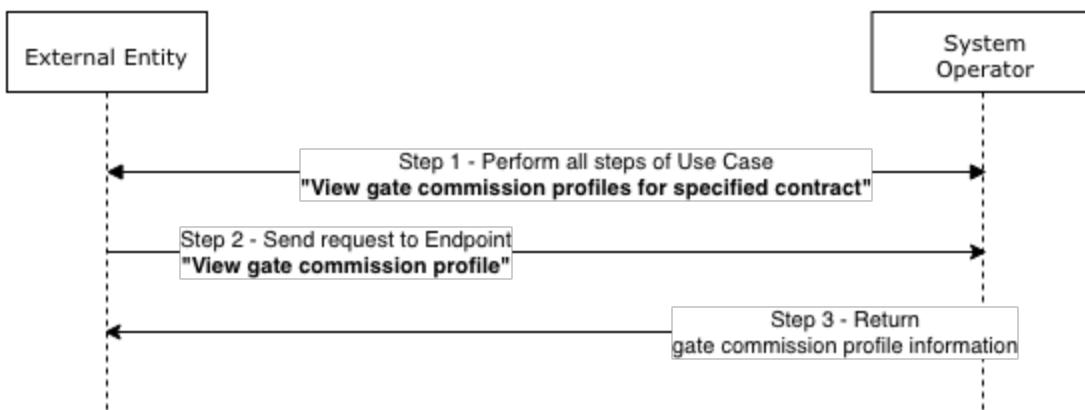
Result example

```
{  
  "profile": {  
    "id": "string",  
    "providerAccountId": "string",  
    "providerCurrency": {  
      "code": "string",  
      "digitalCode": "string",  
      "symbol": "string",  
      "name": "string",  
      "description": "string"  
    },  
    "createdAt": "2018-08-29T10:07:45.896Z",  
    "updatedAt": "2018-08-29T10:07:45.896Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

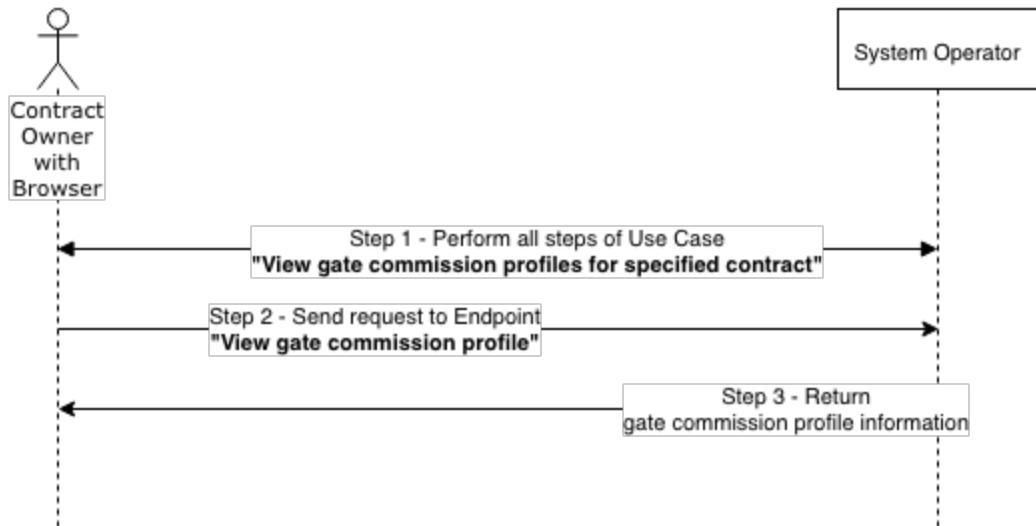
View gate commission profile scheme

Use case: View gate commission profile

Basic FLow



Optional Web UI Flow



View gate commission profiles for specified contract description

Use Case Name

View gate commission profiles for specified contract

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain contracts matching the specified criteria" (to obtain contractID) Use Case, and then send a request to Endpoint "View gate commission profiles for specified contract".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain contracts matching the specified criteria".
2. External Entity sends a request to Endpoint "View gate commission profiles for specified contract".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of gate commission profiles for a specified contract to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Obtain contracts matching the specified criteria".
2. A user sends a request to Endpoint "View gate commission profiles for specified contract".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of gate commission profiles for a specified contract to User (See Result example below).

Post Conditions

Gate commission profiles are available.

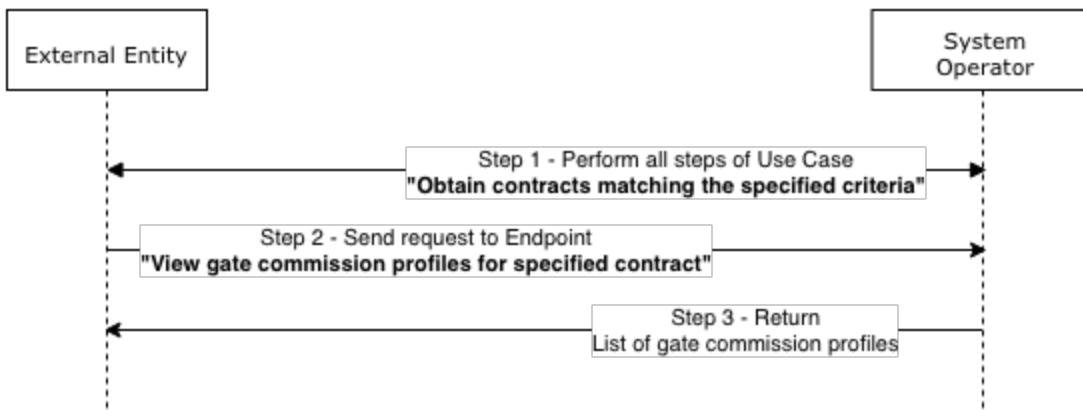
Result example

```
{  
  "profile": {  
    "id": "string",  
    "providerAccountId": "string",  
    "providerCurrency": {  
      "code": "string",  
      "digitalCode": "string",  
      "symbol": "string",  
      "name": "string",  
      "description": "string"  
    },  
    "createdAt": "2018-08-29T10:07:45.890Z",  
    "updatedAt": "2018-08-29T10:07:45.890Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

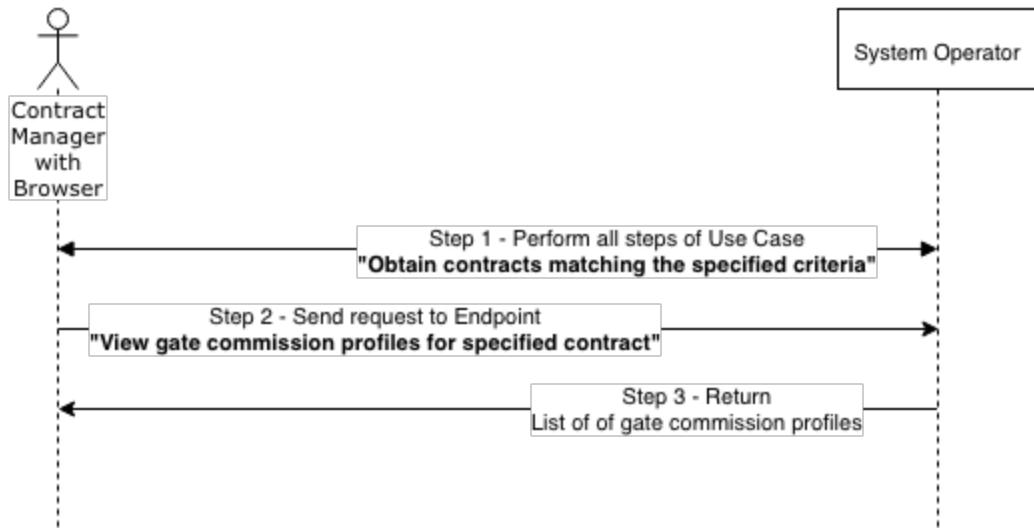
View gate commission profiles for specified contract scheme

Use case: View gate commission profiles for specified contract

Basic FFlow



Optional Web UI Flow



View gate limit profile description

Use Case Name

View gate limit profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "Obtain gate limit profiles for specified gate commission profile" (to obtain limit profile ID and profile ID) Use Case, and then send a request to Endpoint "View gate limit profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. External Entity sends a request to Endpoint "View gate limit profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles/{limitProfileId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate limit profile for a gate commission profile to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Obtain gate limit profiles for specified gate commission profile".
2. A user sends a request to Endpoint "View gate limit profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/limit-profiles/{limitProfileId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns gate limit profile for a gate commission profile to User (See Result example below).

Post Conditions

Gate limit profile is available.

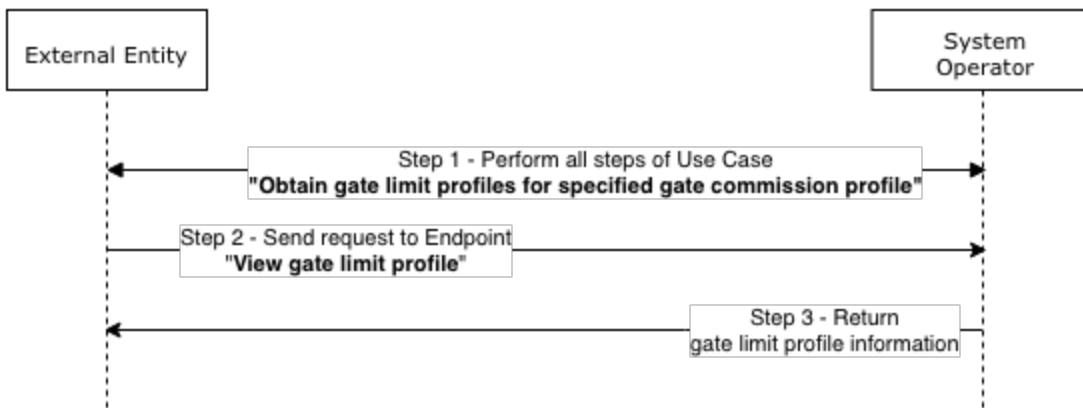
Result example

```
{  
  "profile": {  
    "id": "string",  
    "gateProfileId": "string",  
    "txType": "string",  
    "product": {  
      "id": "string",  
      "name": "string",  
      "description": "string",  
      "category": "MOBILE",  
      "icon": "string",  
      "accountId": "string",  
      "currencies": [  
        {  
          "currency": {  
            "code": "string",  
            "digitalCode": "string",  
            "symbol": "string",  
            "name": "string",  
            "description": "string"  
          },  
          "minAmount": 0,  
          "maxAmount": 0  
        },  
        ...  
      ],  
      "qualifier": "amount",  
      "timeUnit": "per_transaction",  
      "value": 0,  
      "active": false  
    },  
    "status": "ok",  
    "message": "string"  
  }  
}
```

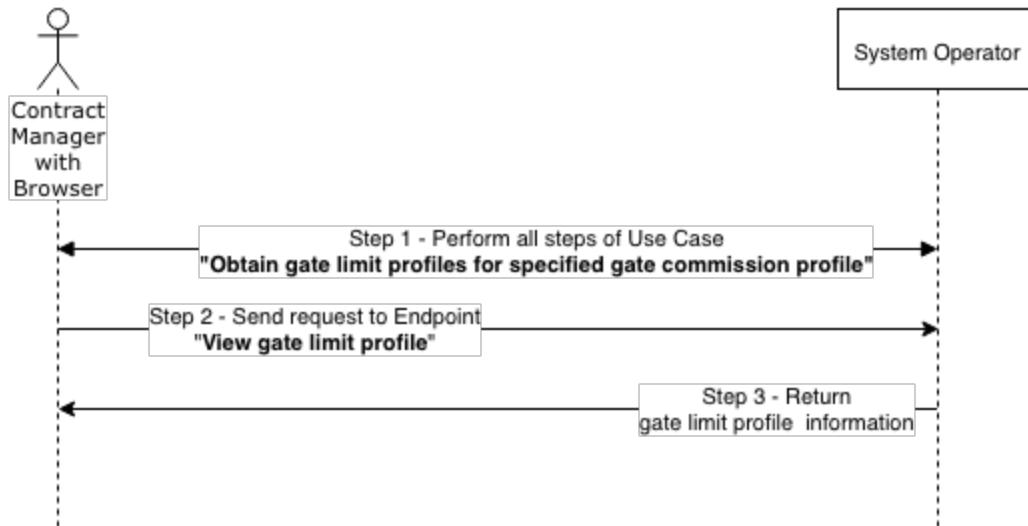
View gate limit profile scheme

Use case: View gate limit profile

Basic FLow



Optional Web UI Flow



View product commission settings of gate commission profile description

Use Case Name

View product commission settings of gate commission profile

Brief Description

A User or External Entity on behalf of a User with role permission "CONTRACT_MANAGER" will go through all steps of "View gate commission profiles for specified contract" (to obtain profile ID) Use Case, and then send a request to Endpoint "View product commission settings of gate commission profile".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CONTRACT_MANAGER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. External Entity sends a request to Endpoint "View product commission settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/product-commission-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns product commission settings for a gate commission profile to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View gate commission profiles for specified contract".
2. A user sends a request to Endpoint "View product commission settings of gate commission profile".

Endpoint URL: GET /contracts/{contractId}/gate-commission-profiles/{profileId}/product-commission-settings-records

Parameters: TOKEN - identifies authenticated user

1. System Operator returns product commission settings for a gate commission profile to User (See Result example below).

Post Conditions

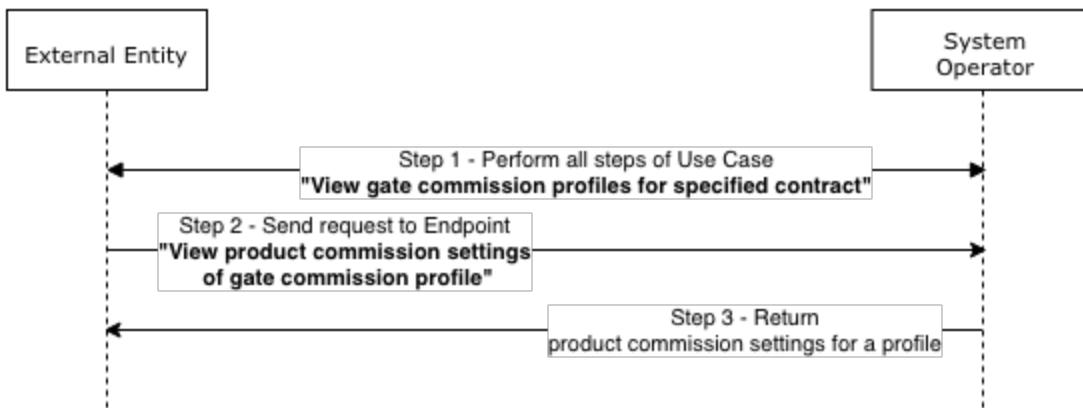
Gate commission profile is available.

Result example

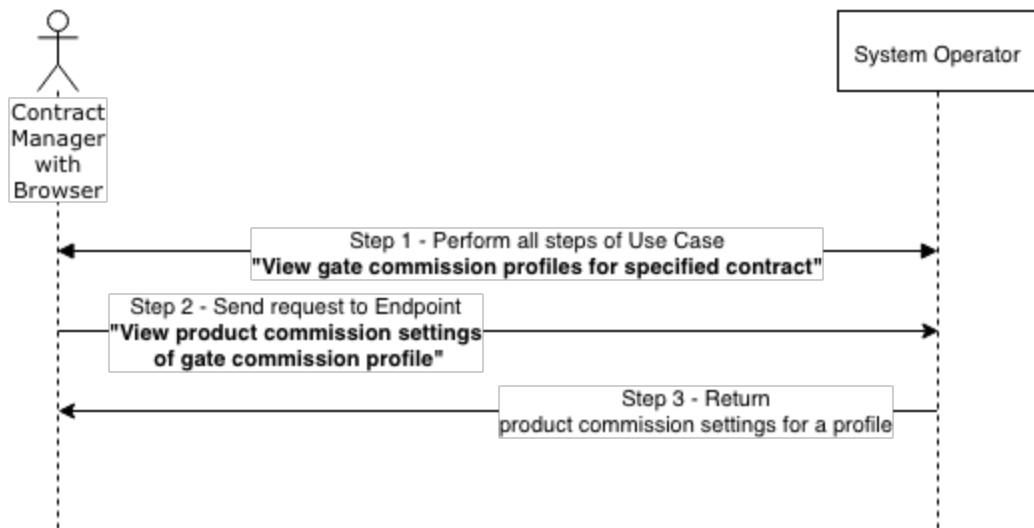
```
{  
  "records": [  
    {  
      "id": "string",  
      "product": {  
        "id": "string",  
        "name": "string",  
        "description": "string",  
        "category": "MOBILE",  
        "icon": "string",  
        "accountId": "string",  
        "currencies": [  
          {  
            "currency": {  
              "code": "string",  
              "digitalCode": "string",  
              "symbol": "string",  
              "name": "string",  
              "description": "string"  
            },  
            "minAmount": 0,  
            "maxAmount": 0  
          }  
        ],  
        "collector": "BANK",  
        "active": false,  
        "value": {  
          "type": "zero",  
          "valuePercent": 0,  
          "valueFixed": 0  
        }  
      },  
      "status": "ok",  
      "message": "string"  
    }  
  ]  
}  
View product commission settings of gate commission profile scheme
```

Use case: View product commission settings of gate commission profile

Basic FLow



Optional Web UI Flow



Gate purchases - pay for external services with your wallet

Filter products available for purchase description

Use Case Name

Filter products available for purchase

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Filter products available for purchase".

Note: The filter is optional. When there are no filtering parameter available, the filter could be omitted to get all available products.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Filter products available for purchase".

Endpoint URL: POST /gate/products/view

Parameters: {

```
"filter": {
    "id": "string",
    "category": "MOBILE",
    "accountId": "string",
    "name": "string"
},
"sort": {
    "category": "asc"
},
"pageNumber": 0,
"pageSize": 0
}
```

1. System Operator returns a list of products available for purchase to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Filter products available for purchase”.

Endpoint URL: POST /gate/products/view

Parameters: {

```
"filter": {
  "id": "string",
  "category": "MOBILE",
  "accountId": "string",
  "name": "string"
},
"sort": {
  "category": "asc"
},
"pageNumber": 0,
"pageSize": 0
}
```

1. System Operator returns a list of products available for purchase to User (See Result example below).

Post Conditions

One or more more products are available. Token is available.

Result example

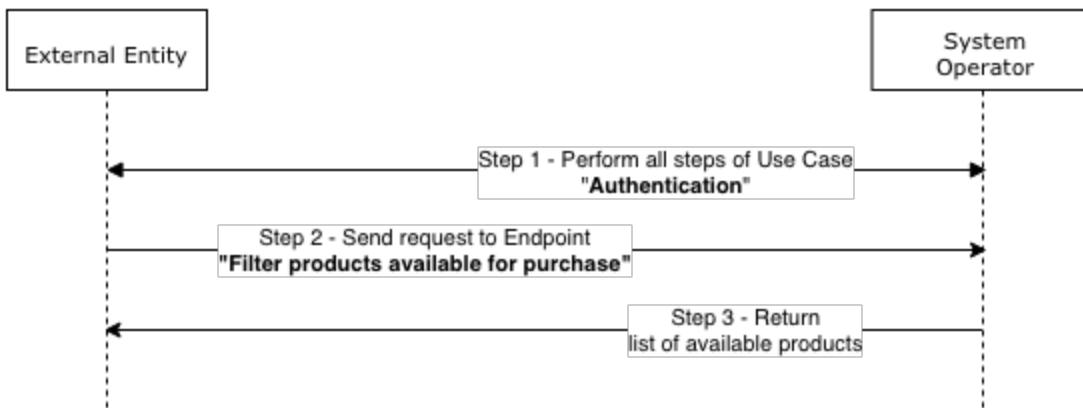
```
{
  "records": [
    {
      "id": "string",
      "name": "string",
      "description": "string",
      "category": "MOBILE",
      "icon": "string",
      "accountId": "string",
      "currencies": [
        {
          "id": "string",
          "name": "string",
          "symbol": "string",
          "rate": 1.0
        }
      ]
    }
  ]
}
```

```
"currency": {  
    "code": "string",  
    "digitalCode": "string",  
    "symbol": "string",  
    "name": "string",  
    "description": "string"  
},  
"minAmount": 0,  
"maxAmount": 0  
}  
]  
}  
],  
"status": "ok",  
"message": "string",  
"pageNumber": 0,  
"pageSize": 0,  
"totalRecords": 0,  
"totalPages": 0  
}
```

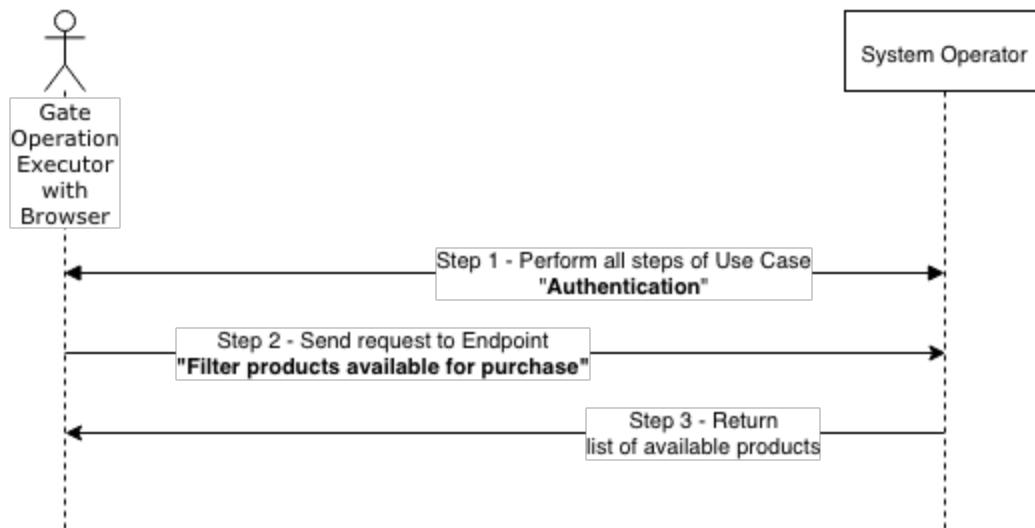
Filter products available for purchase scheme

Use case: Filter products available for purchase

Basic FFlow



Optional Web UI Flow



Get all products for gate provider account description

Use Case Name

Get all products for gate provider account

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "View provider accounts" (to obtain provider account ID) Use Case, and then send a request to Endpoint "Get all products for gate provider account".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View provider accounts".
2. External Entity sends a request to Endpoint "Get all products for gate provider account".

Endpoint URL: GET /gate/products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of products for a provider account to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View provider accounts".
2. A user sends a request to Endpoint "Get all products for gate provider account".

Endpoint URL: GET /gate/products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of products for a provider account to User (See Result example below).

Post Conditions

Provider account is available.

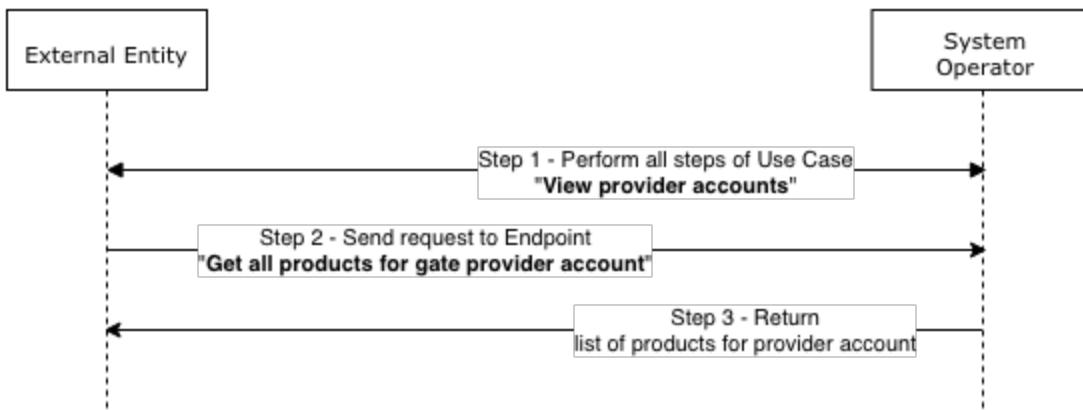
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "name": "string",  
      "description": "string",  
      "category": "MOBILE",  
      "icon": "string",  
      "accountId": "string",  
      "currencies": [  
        {  
          "currency": {  
            "code": "string",  
            "digitalCode": "string",  
            "symbol": "string",  
            "name": "string",  
            "description": "string"  
          },  
          "minAmount": 0,  
          "maxAmount": 0  
        },  
      ],  
      "status": "ok",  
      "message": "string"  
    }  
  ]  
},  
"status": "ok",  
"message": "string"
```

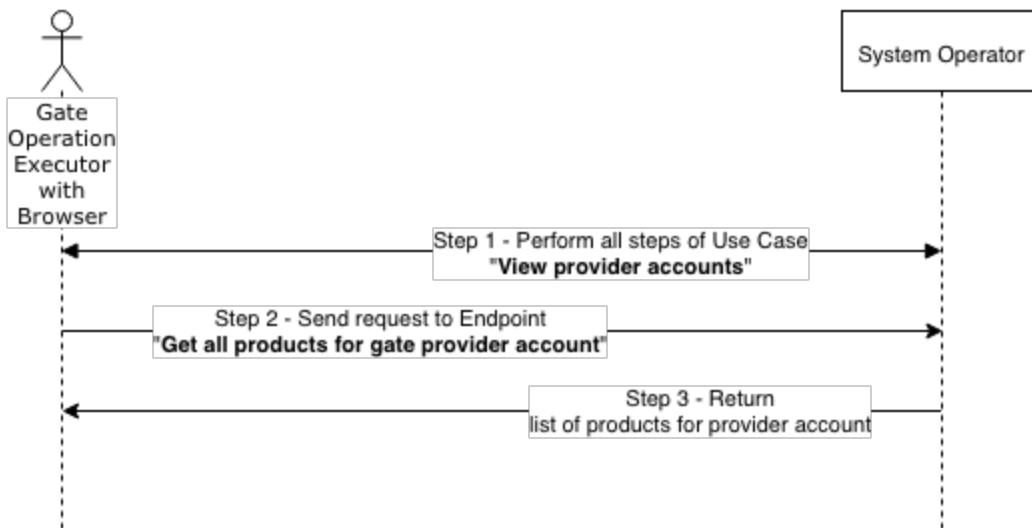
Get all products for gate provider account scheme

Use case: Get all products for gate provider account

Basic FFlow



Optional Web UI Flow



Get payer fields for specific product description

Use Case Name

Get payer fields for specific product

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 1: get list of all available products" (to obtain productId) Use Case, and then send a request to Endpoint "Get payer fields for specific product".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. External Entity sends a request to Endpoint "Get payer fields for specific product".

Endpoint URL: GET /gate/product/{id}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns payer fields for specific product to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. A user sends a request to Endpoint "Get payer fields for specific product".

Endpoint URL: GET /gate/product/{id}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns payer fields for specific product to User (See Result example below).

Post Conditions

Product is available and payer fields exist. Token is available..

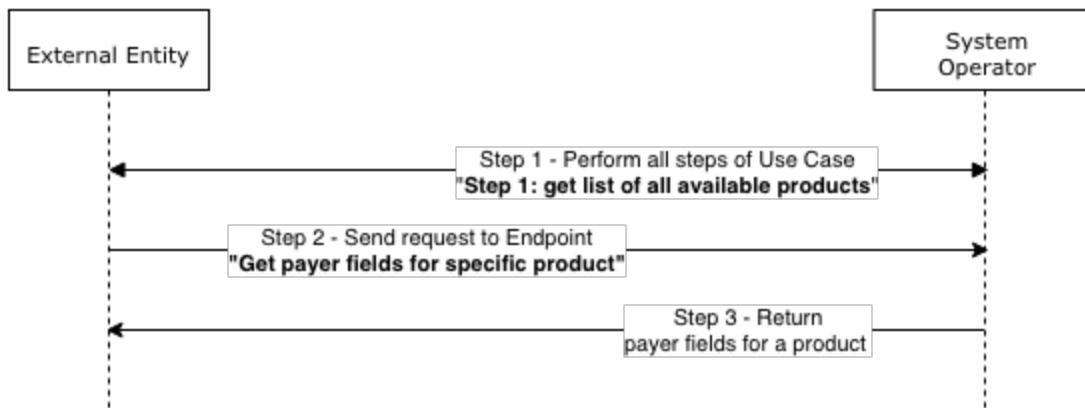
Result example

```
{  
  "options": [  
    {  
      "name": "default",  
      "fields": [  
        {  
          "name": "string",  
          "optional": false,  
          "label": "string",  
          "hint": "string",  
          "constraints": [  
            "string"  
          ]  
        }  
      ]  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

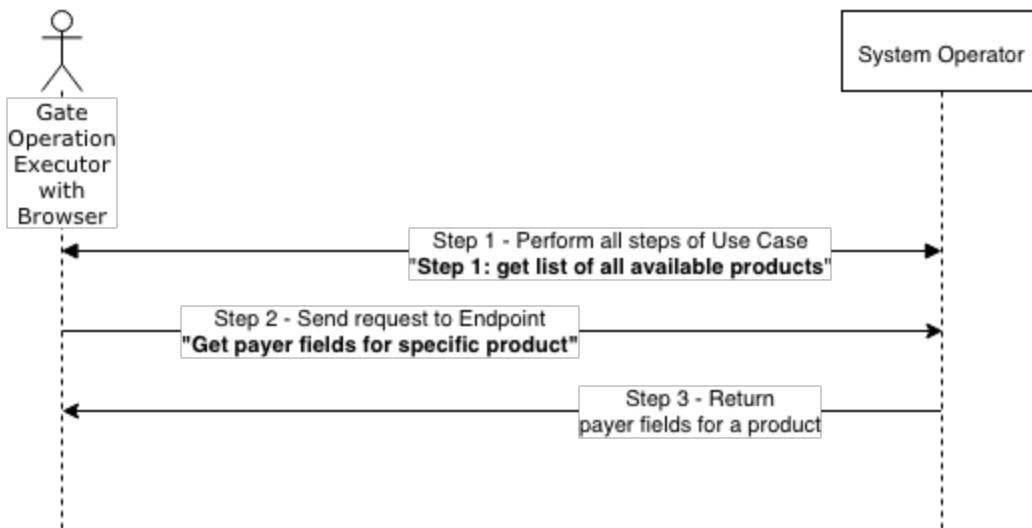
Get payer fields for specific product scheme

Use case: Get payer fields for specific product

Basic FLow



Optional Web UI Flow



Step 1: get list of all available products description

Use Case Name

Get payer fields for specific product

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 1: get list of all available products" (to obtain productId) Use Case, and then send a request to Endpoint "Get payer fields for specific product".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. External Entity sends a request to Endpoint "Get payer fields for specific product".

Endpoint URL: GET /gate/product/{id}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns payer fields for specific product to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. A user sends a request to Endpoint "Get payer fields for specific product".

Endpoint URL: GET /gate/product/{id}/payer-fields

Parameters: TOKEN - identifies authenticated user

1. System Operator returns payer fields for specific product to User (See Result example below).

Post Conditions

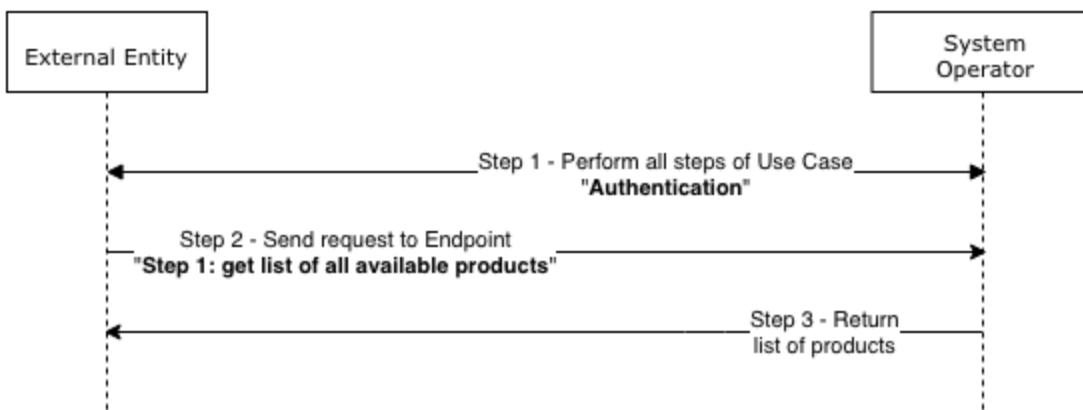
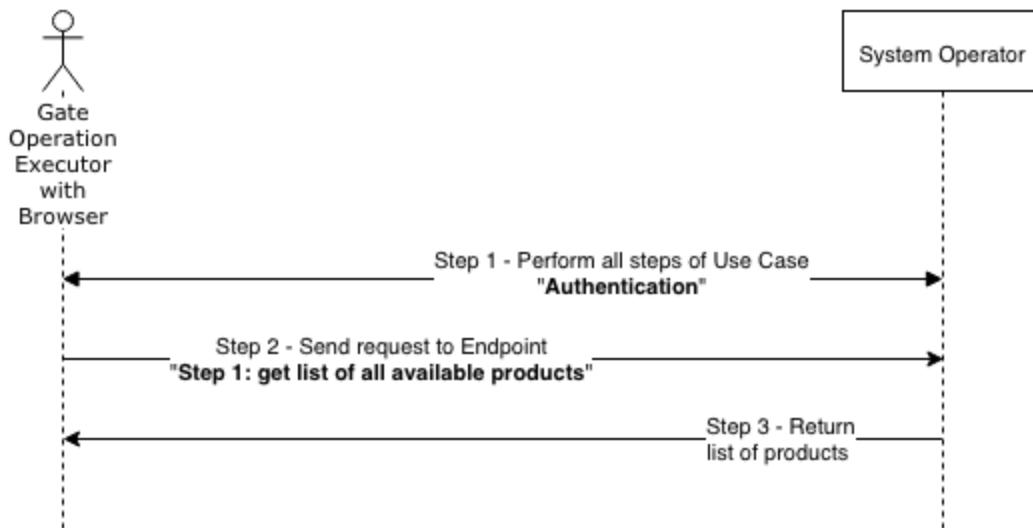
Product is available and payer fields exist. Token is available..

Result example

```
{  
  "options": [  
    {  
      "name": "default",  
      "fields": [  
        {  
          "name": "string",  
          "optional": false,  
          "label": "string",  
          "hint": "string",  
          "constraints": [  
            "string"  
          ]  
        }  
      ]  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

Step 1: get list of all available products scheme

Use case: Step 1: get list of all available products

Basic FFlow**Optional Web UI Flow**

Step 2: calculate commission amount description

Use Case Name

Step 2: calculate commission amount

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 1: get list of all available products" (in order to obtain a productID) and "Get coins owned by current user" (in order to obtain a serialId) Use Cases, and then send a request to Endpoint "Step 2: calculate commission amount".

Note: When performing "Step 3: create purchase for selected product" Use Case, it is not necessary to perform "Step 1: get list of all available products" Use Case, as this will have been done in the "Get coins owned by current user" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. Perform all steps of Use Case "Get coins owned by current user".
3. External Entity sends a request to Endpoint "Step 2: calculate commission amount".

Endpoint URL: POST /gate/purchases/calculate

```
Parameters: {
  "productId": "string",
  "serial": "string",
  "amount": 0
}
```

1. System Operator returns calculated commission amount to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 1: get list of all available products".
2. Perform all steps of Use Case "Get coins owned by current user".
3. A user sends a request to Endpoint "Step 2: calculate commission amount".

Endpoint URL: POST /gate/purchases/calculate

```
Parameters: {
```

```
"productId": "string",
"serial": "string",
"amount": 0
}
```

1. System Operator returns calculated commission amount to User (See Result example below).

Post Conditions

Product and coins are available.

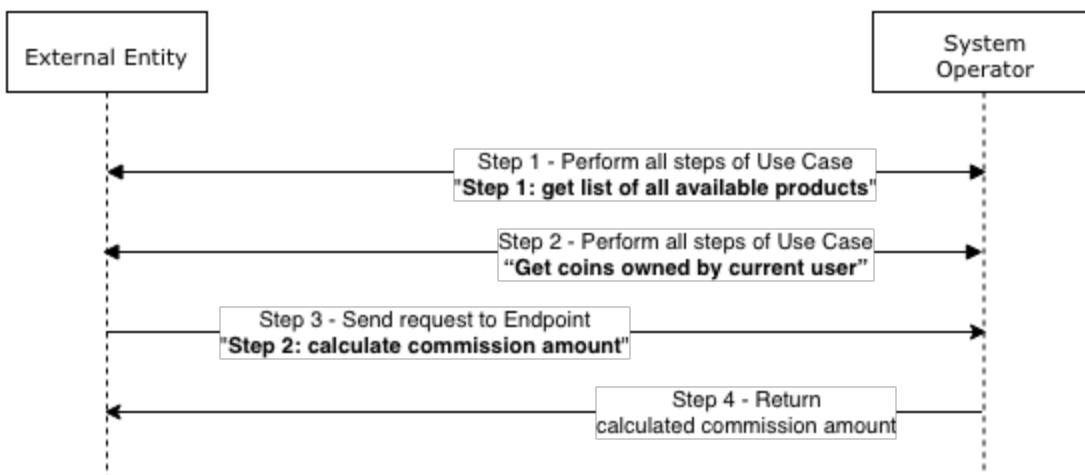
Result example

```
{
  "sourceAmount": 0,
  "amountToSend": 0,
  "commissionAmount": 0,
  "currency": {
    "code": "string",
    "digitalCode": "string",
    "symbol": "string",
    "name": "string",
    "description": "string"
  },
  "status": "ok",
  "message": "string"
}
```

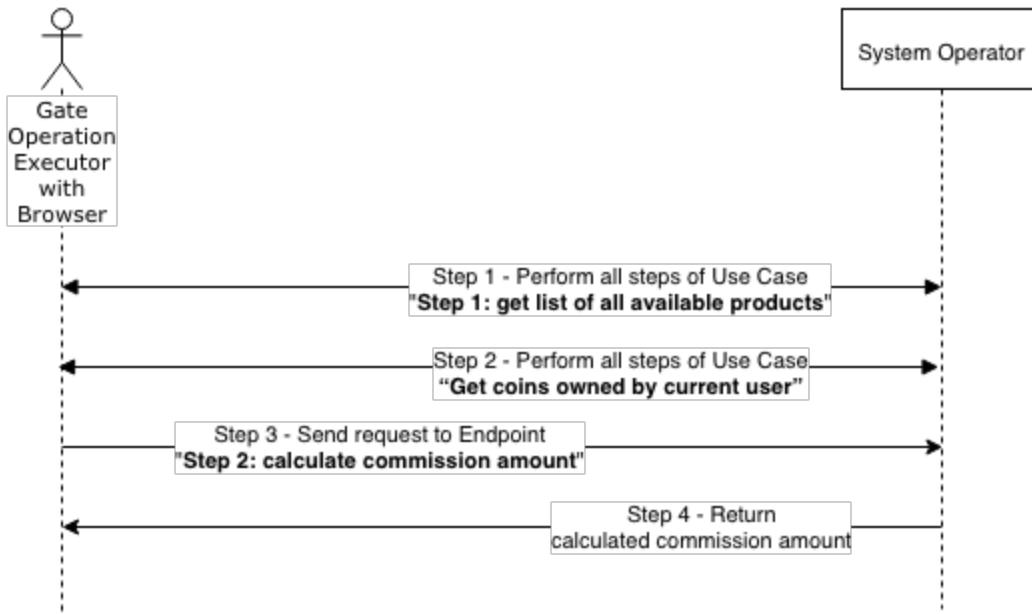
Step 2: calculate commission amount scheme

Use case: Step 2: calculate commission amount

Basic FFlow



Optional Web UI Flow



Step 3: create purchase for selected product description

Use Case Name

Step 3: create purchase for selected product

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Step 2: calculate commission amount" Use Case, and then send a request to Endpoint "Step 3: create purchase for selected product",

Note: "Step 2: calculate commission amount" Use Case calls "Step 1: get list of all available products" (in order to obtain a productID) and "Get coins owned by current user" (in order to obtain a serialId) Use Cases. "Step 1: get list of all available products" Use Case can be used as a standalone module to obtain a list of available products.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Step 2: calculate commission amount".
2. External Entity sends a request to Endpoint "Step 3: create purchase for selected product".

Endpoint URL: POST /gate/purchases

```
Parameters: {
  "productId": "string",
  "serial": "string",
  "amount": 0
}
```

1. System Operator returns payment information for a product purchase to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Step 2: calculate commission amount".
2. A user sends a request to Endpoint "Step 3: create purchase for selected product".

Endpoint URL: POST /gate/purchases

Parameters: {
 "productId": "string",
 "serial": "string",
 "amount": 0
 }

1. System Operator returns payment information for a product purchase to User (See Result example below).

Post Conditions

Product and coin information is available.

Result example

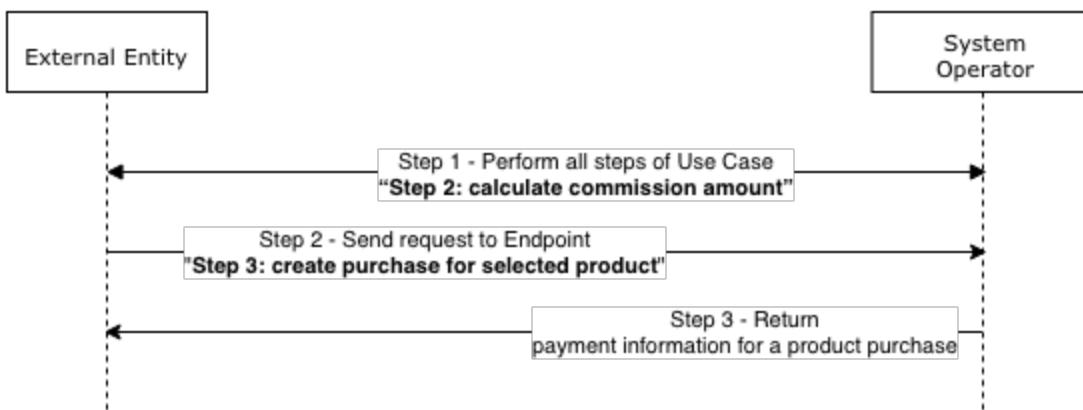
```
{
  "transaction": {
    "id": "string",
    "orderId": 0,
    "deviceId": "string",
    "deviceOrderId": "string",
    "type": "TOPUP",
    "status": "INITIATED",
    "errorCode": "UNKNOWN",
    "coin": {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "availableAmount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "active": false,
      "type": "regular_commission"
    },
    "paymentMethod": {
      "accountId": "string",
      "account": {
        "id": "string"
      }
    }
  }
}
```

```
"id": "string",
"provider": {
    "name": "string"
},
"way": "string"
},
"sourceAmount": 0,
"amountToSend": 0,
"finalAmount": 0,
"processId": "string",
"payerData": {}
},
"status": "ok",
"message": "string"
}
```

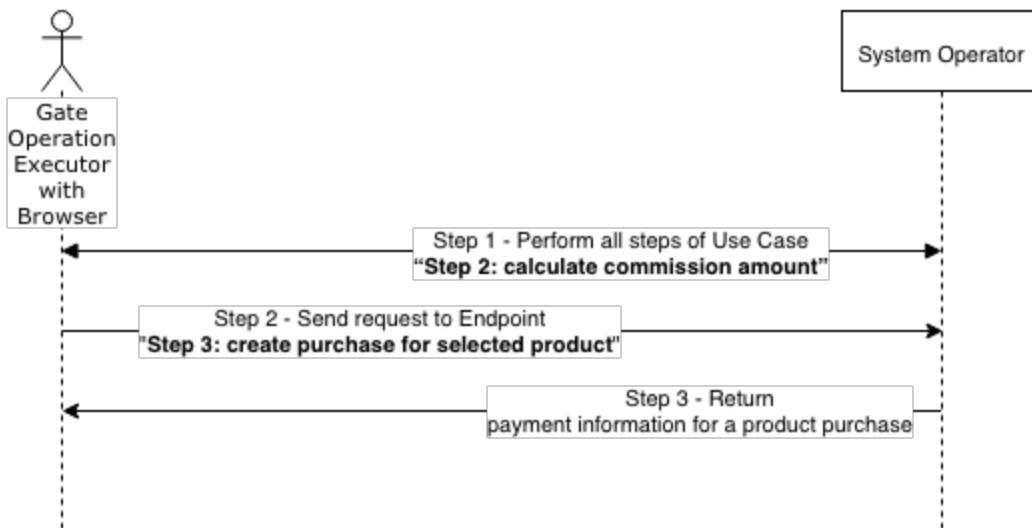
Step 3: create purchase for selected product scheme

Use case: Step 3: create purchase for selected product

Basic FFlow



Optional Web UI Flow



Prepaid

Activate prepaid coin description

Use Case Name

Activate prepaid coin

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_REDEEM_EXECUTOR will go through all steps of “View prepaid vouchers” Use Case, and then send a request to Endpoint “Activate prepaid coin”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PREPAID_REDEEM_EXECUTOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View prepaid vouchers”.
2. External Entity sends a request to Endpoint “Activate prepaid coin”.

Endpoint URL: POST /prepaid-coins/{serial}/activate

Parameters:

```
{
  "prepaidPin": "string",
  "destSerial": "string"
}
```

“prepaidPin”: “string”, - passed to receiver by any channel.

“destSerial”: “string” - Serial of destination coin where funds will be transferred to.

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View prepaid vouchers”.
2. A user sends a request to Endpoint “Activate prepaid coin”.

Endpoint URL: POST /prepaid-coins/{serial}/activate

Parameters:

```
{
  "prepaidPin": "string",
  "destSerial": "string"
}
```

“prepaidPin”: “string”, - passed to receiver by any channel.

“destSerial”: “string” - Serial of destination coin where funds will be transferred to.

3. System Operator returns result information to User (See Result example below).

Post Conditions

Prepaid coin is activated.

Result example

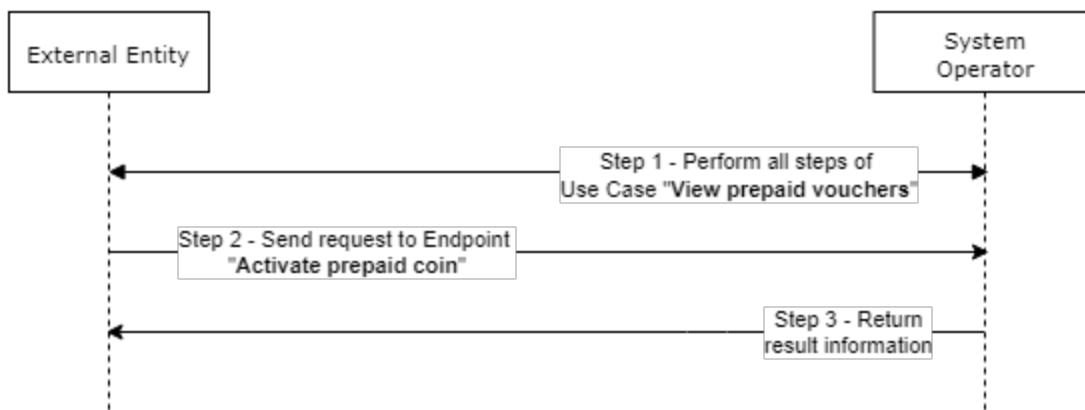
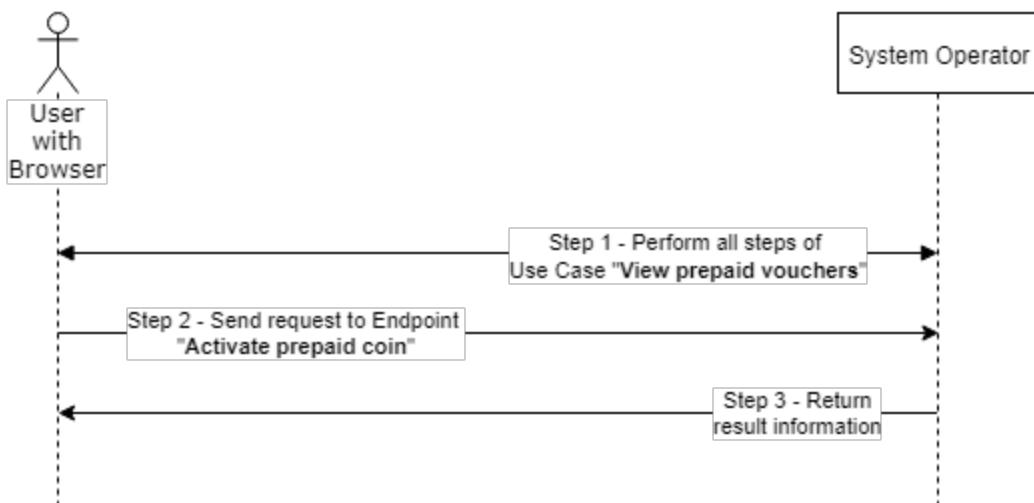
```
{
  "process": {
    "id": "string",
    "createdAt": "2018-08-10T11:35:33.852Z",
    "updatedAt": "2018-08-10T11:35:33.852Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "to": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "amount": 0,
        "performedAt": "2018-08-10T11:35:33.852Z",
        "issuer": {
          "id": "string",
          "sn": "string",
          "currency": "string"
        }
      }
    ]
  }
}
```

```
        }
    }
],
"children": [
{
}
],
"errorMessage": "string"
},

```

```
"status": "ok",  
"message": "string"  
}
```

Activate prepaid coin scheme

Use case: Activate prepaid coin**Basic FFlow****Optional Web UI Flow**

Calculate commission for prepaid creation description

Use Case Name

Calculate commission for prepaid creation

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_CREATION_EXECUTOR will go through all steps of “Get coins owned by current user” Use Case and then send a request to Endpoint “Calculate commission for prepaid creation” after selecting a source coin.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PREPAID_CREATION_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. External Entity sends a request to Endpoint “Calculate commission for prepaid creation”.

Endpoint URL: POST /prepaid-coins/calculate

Parameter:

```
{
  "srcSerial": "string",
  "prepaidAmount": 0
}
```

“srcSerial”: “string”, - Serial number of the coin where funds will be taken from

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. A user sends a request to Endpoint “Calculate commission for prepaid creation”.

Endpoint URL: POST /prepaid-coins/calculate

Parameter:

```
{
  "srcSerial": "string",
  "prepaidAmount": 0
}
```

“srcSerial”: “string”, - Serial number of the coin where funds will be taken from

3. System Operator returns result information to User (See Result example below).

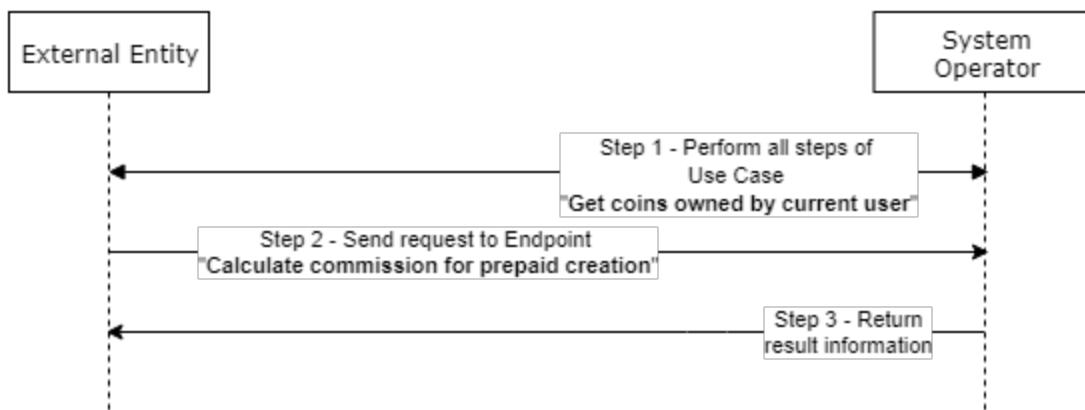
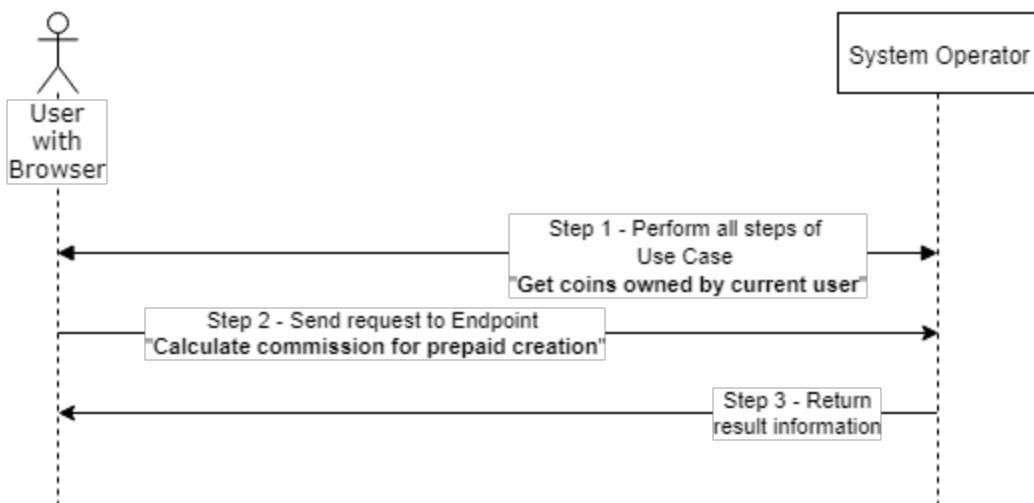
Post Conditions

Commission amount is properly calculated.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "Change of the sender coin balance": 0,  
    "Change of the recipient coin balance": 0,  
    "Change of the commission coin balance": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for prepaid creation scheme

Use case: Calculate commission for prepaid creation**Basic FFlow****Optional Web UI Flow**

Calculate commission for redeem of the prepaid description

Use Case Name

Calculate commission for redeeming of the prepaid

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_REDEEM_EXECUTOR will go through all steps of “View prepaid vouchers” Use Case and then send a request to Endpoint “Calculate commission for redeeming of the prepaid”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
PREPAID_REDEEM_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View prepaid vouchers”.
2. External Entity sends a request to Endpoint “Calculate commission for redeeming of the prepaid”.

Endpoint URL: POST /prepaid-coins/{serial}/calculate

Parameter:

```
{
  "prepaidPin": "string",
  "destSerial": "string"
}
```

“prepaidPin”: “string”, - PIN of the prepaid coin

“destSerial”: “string” - Serial of destination coin where funds will be transferred to

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View prepaid vouchers”.
2. A user sends a request to Endpoint “Calculate commission for redeem of the prepaid”.

Endpoint URL: POST /prepaid-coins/{serial}/calculate

Parameter:

```
{
  "prepaidPin": "string",
  "destSerial": "string"
}
```

"prepaidPin": "string", - PIN of the prepaid coin
"destSerial": "string" - Serial of destination coin where funds will be transferred to
3. System Operator returns result information to User (See Result example below).

Post Conditions

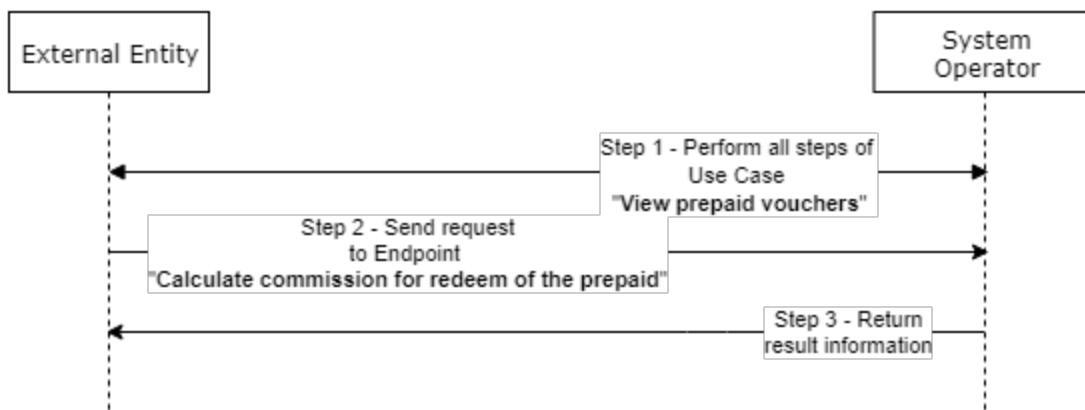
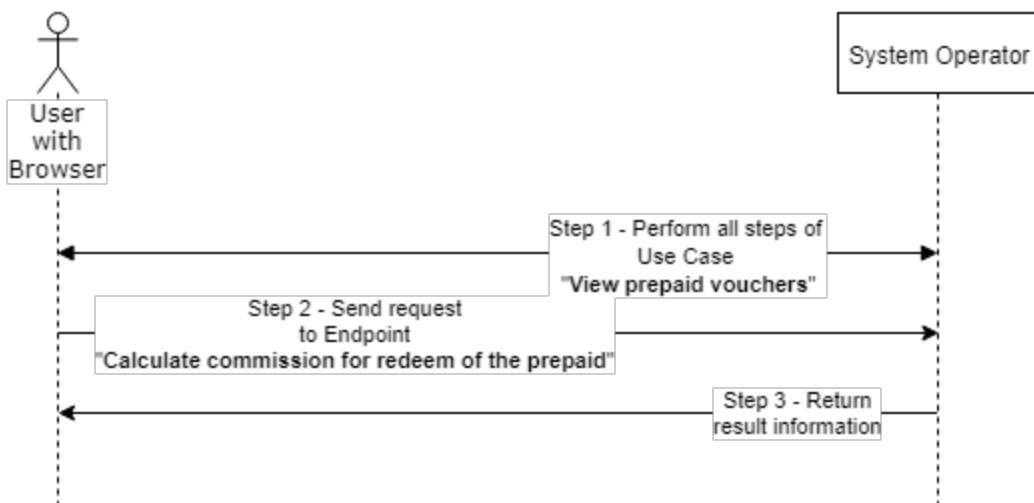
Right commission amount is calculated.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

"senderAmountPush": 0, - Increase of the sender coin balance
"recipientAmountPush": 0, - numberIncrease of the recipient coin balance
"commissionAmountPush": 0, - numberIncrease of the commission coin balance

Calculate commission for redeem of the prepaid scheme

Use case: Calculate commission for redeem of the prepaid**Basic FFlow****Optional Web UI Flow**

Change pin for prepaid description

Use Case Name

Change pin for prepaid

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_PIN_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Change pin for prepaid”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PREPAID_PIN_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
3. Communication channel provider - mobile phone company or email server.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Change pin for prepaid”.

Endpoint URL: POST /prepaid-coins/{serial}/change-pin

Parameters: TOKEN

3. System Operator generates, saves, and sends the PIN to the user via SMS or Email.
4. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Change pin for prepaid”.

Endpoint URL: POST /prepaid-coins/{serial}/change-pin

Parameters: TOKEN

3. System Operator generates, saves, and sends the PIN to the user via SMS or Email.
4. System Operator returns result information to User (See Result example below).

Post Conditions

The new PIN is available.

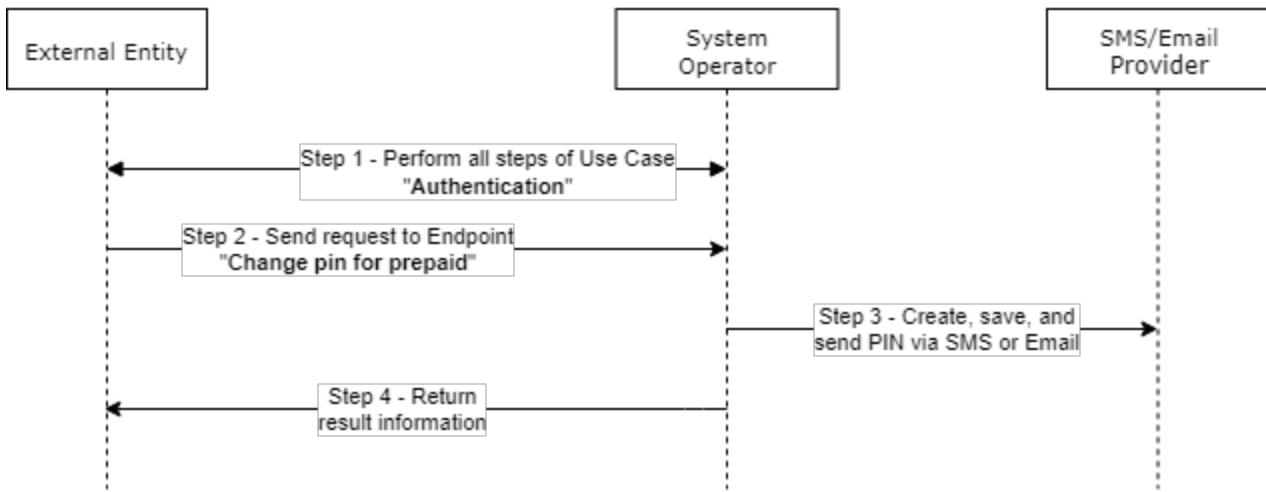
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

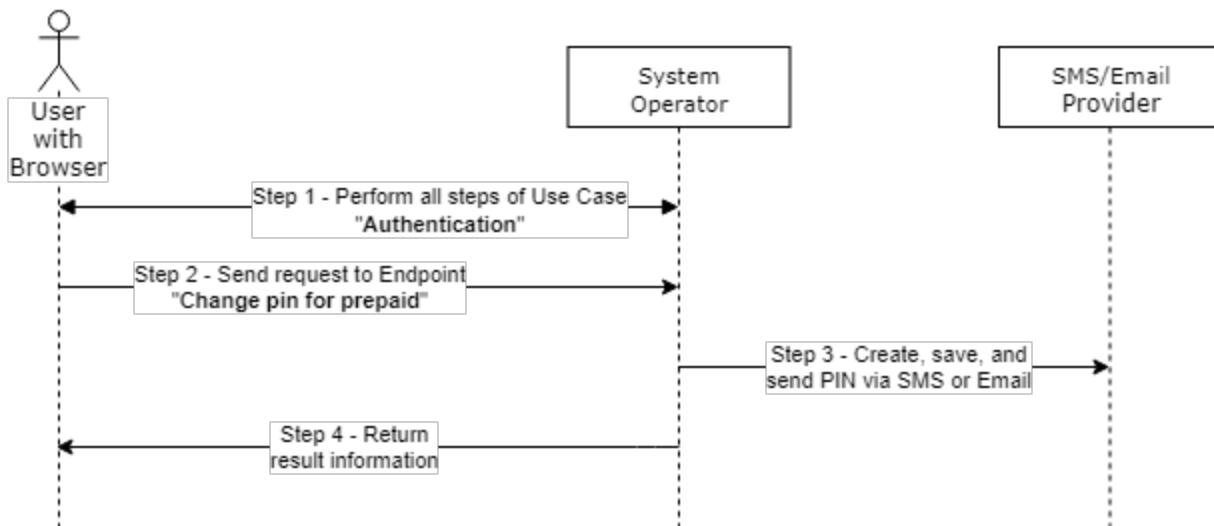
Change pin for prepaid scheme

Use case: XX_TEMPLATE

Basic FLow



Optional Web UI Flow



Create a prepaid description

Use Case Name

Create a prepaid

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_CREATION_EXECUTOR will go through all steps of “Get coins owned by current user” Use Case, and then send a request to Endpoint “Create a prepaid”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PREPAID_CREATION_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. External Entity sends a request to Endpoint “Create a prepaid”.

Endpoint URL: POST /prepaid-coins

Parameter:

```
{
  "srcSerial": "string",
  "prepaidAmount": 0
}
```

“srcSerial”: “string” - Serial number of the coin where funds will be taken from

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. A user sends a request to Endpoint “Create a prepaid”.

Endpoint URL: POST /prepaid-coins

Parameter:

```
{
  "srcSerial": "string",
  "prepaidAmount": 0
}
```

“srcSerial”: “string” - Serial number of the coin where funds will be taken from

3. System Operator returns result information to User (See Result example below).

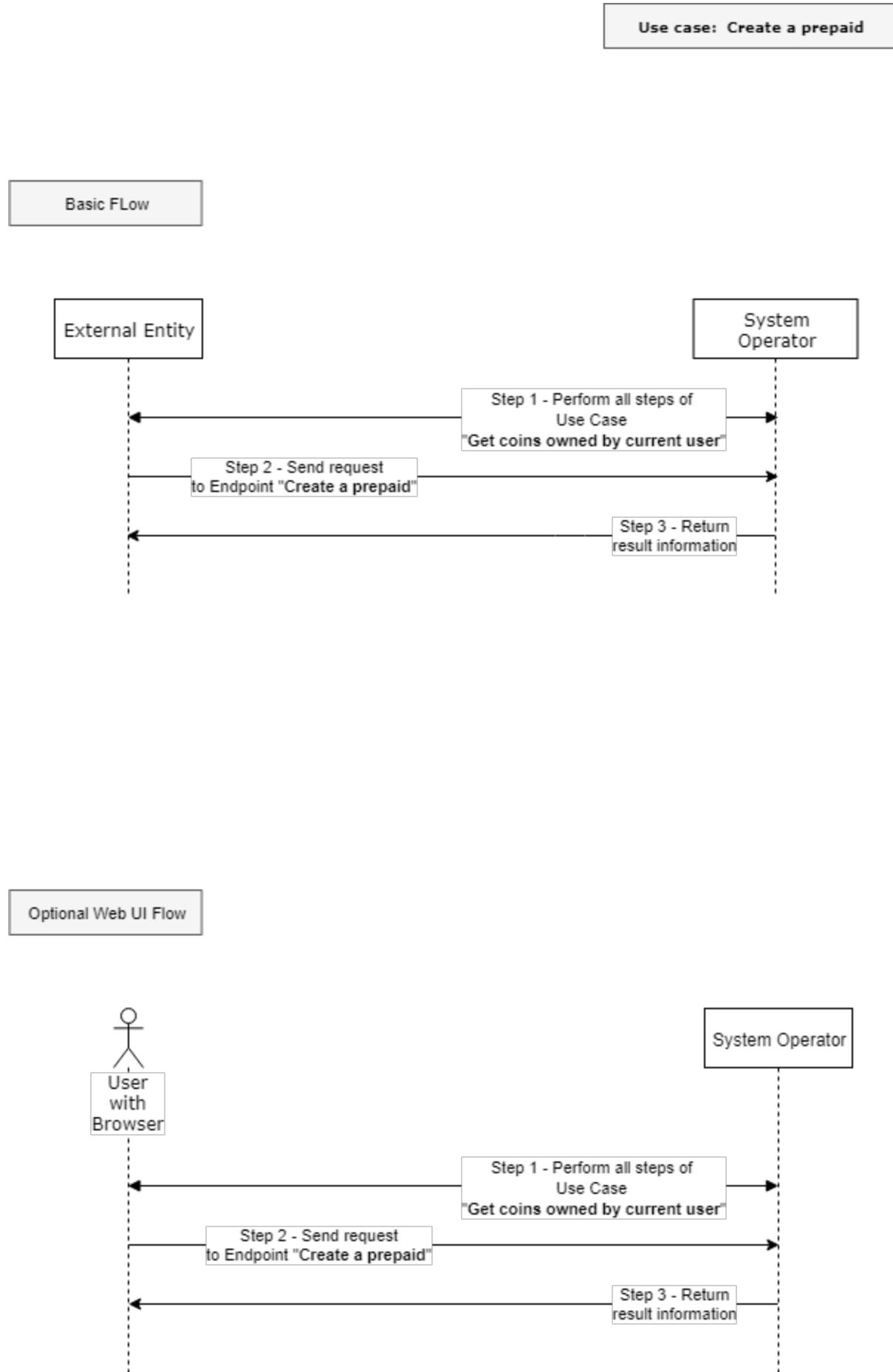
Post Conditions

New voucher is available

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "coin": {  
        "Unique serial number": "string",  
        "name": "string",  
        "amount": 0,  
        "availableAmount": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "Coin's PIN": "string"  
}
```

Create a prepaid scheme



View prepaid vouchers description

Use Case Name

View prepaid vouchers

Brief Description

A User or External Entity on behalf of a User with role permission PREPAID_CREATION_EXECUTOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View prepaid vouchers”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with the following role permissions:
PREPAID_CREATION_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View prepaid vouchers”.

Endpoint URL: POST /prepaid-coins/view

Parameter:

```
{
  "filter": {
    "activated": false,
    "createdFrom": "2018-07-25T12:12:32.341Z",
    "createdTo": "2018-07-25T12:12:32.341Z",
    "activatedFrom": "2018-07-25T12:12:32.341Z",
    "activatedTo": "2018-07-25T12:12:32.341Z"
  },
  "sort": {
    "createDate": "asc",
    "redeemDate": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns requested information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View prepaid vouchers”.

Endpoint URL: POST /prepaid-coins/view

Parameter:

```
{
  "filter": {
    "activated": false,
    "createdFrom": "2018-07-25T12:12:32.341Z",
    "createdTo": "2018-07-25T12:12:32.341Z",
    "activatedFrom": "2018-07-25T12:12:32.341Z",
    "activatedTo": "2018-07-25T12:12:32.341Z"
  },
  "sort": {
    "createDate": "asc",
    "redeemDate": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns requested information to User (See Result example below).

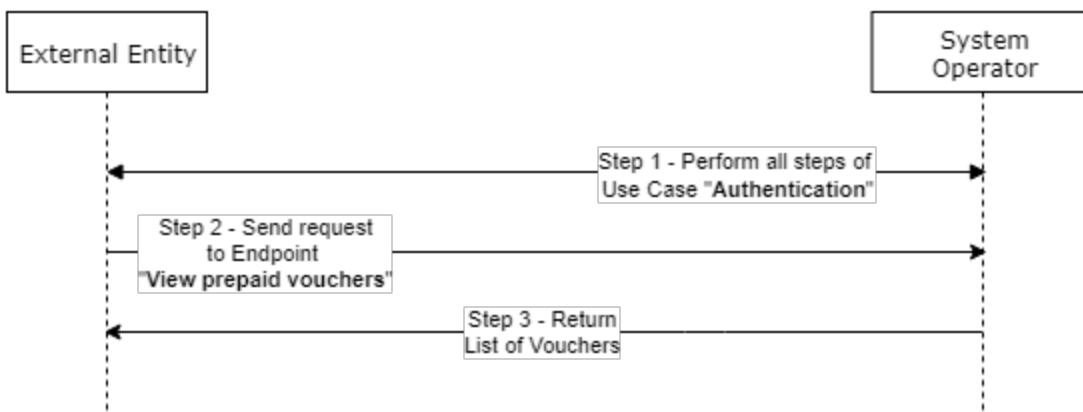
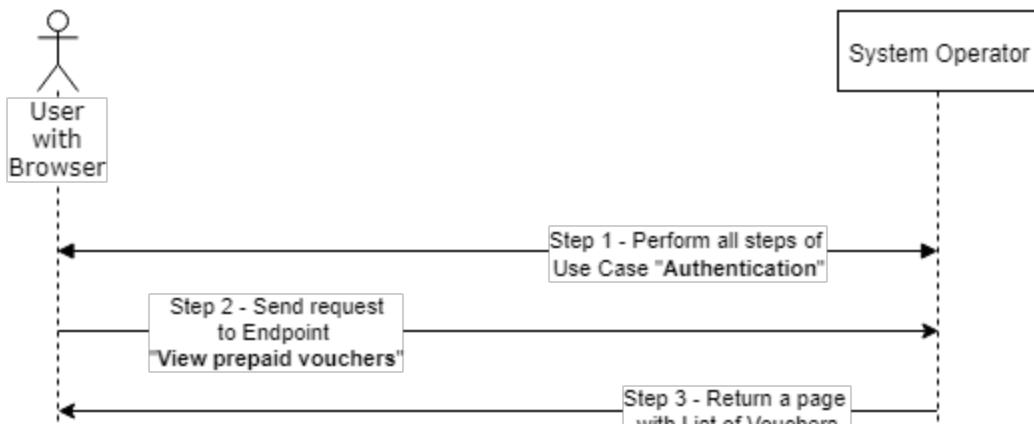
Post Conditions

If the voucher is available, they will be provided.

Result example

```
{
  "status": "ok",
  "message": "string",
  "prepaidVouchers": [
    {
      "serial": "995782708163",
      "createdAt": "2018-07-25T12:12:33.870Z",
      "activatedAt": "2018-07-25T12:12:33.870Z",
      "amount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "status": "string"
    }
  ]
}
```

[View prepaid vouchers scheme](#)

Use case: View prepaid vouchers**Basic FLow****Optional Web UI Flow**

VIEW PROFILE'S BANK ACCOUNTS
CREATE A NEW BANK ACCOUNT

Profile - bank accounts

Create a new bank account description

Use Case Name

Create a new bank account

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “View profile's bank accounts” Use Case, and then send a request to Endpoint “Create a new bank account”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View profile's bank accounts”.
2. External Entity sends a request to Endpoint “Create a new bank account”.

Endpoint URL: POST /profiles/my/bank-accounts

Parameter:

```
{
  "accountDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View profile's bank accounts”.
2. A user sends a request to Endpoint “Create a new bank account”.

Endpoint URL: POST /profiles/my/bank-accounts

Parameter:

```
{
  "accountDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator returns result information to User (See Result example below).

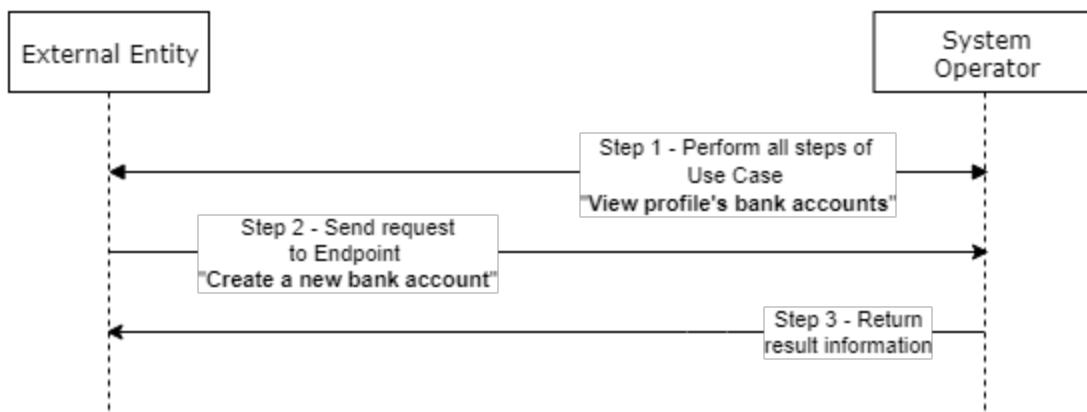
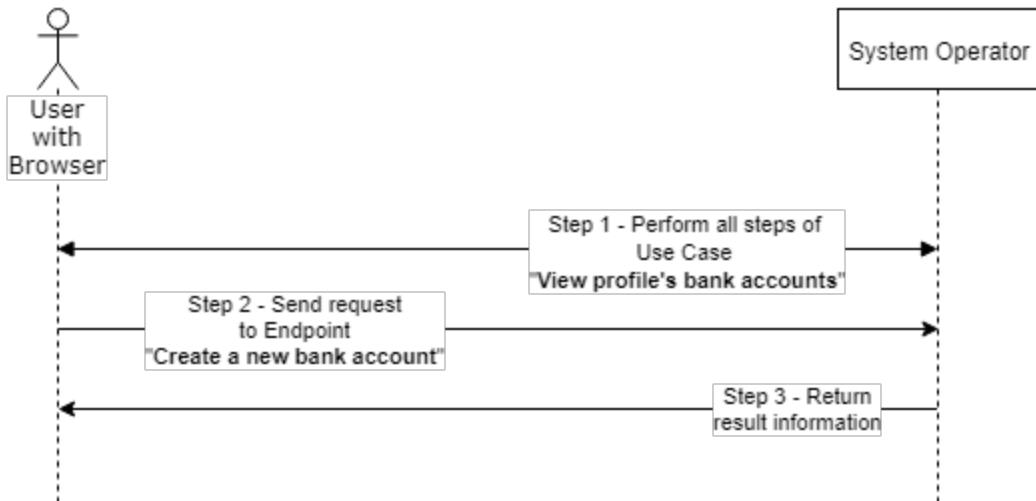
Post Conditions

A new bank account is available.

Result example

```
{
  "account": {
    "id": "string",
    "createdAt": "2018-08-07T14:16:19.137Z",
    "updatedAt": "2018-08-07T14:16:19.137Z",
    "status": "PENDING",
    "details": {
      "fullName": "string",
      "account": "string",
      "iban": "string",
      "bic": "string",
      "swift": "string",
      "name": "string",
      "address": "string"
    }
  },
  "status": "ok",
  "message": "string"
}
```

Create a new bank account scheme

Use case: Create a new bank account**Basic FFlow****Optional Web UI Flow**

Delete a bank account description

Use Case Name

Delete a bank account

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “View profile's bank accounts” Use Case, and then send a request to Endpoint “Delete a bank account”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View profile's bank accounts”.
2. External Entity sends a request to Endpoint “Delete a bank account”.

Endpoint URL: DELETE /profiles/my/bank-accounts/{id}

Parameter: TOKEN only

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View profile's bank accounts”.
2. A user sends a request to Endpoint “Delete a bank account”.

Endpoint URL: DELETE /profiles/my/bank-accounts/{id}

Parameter: TOKEN only

3. System Operator returns result information to User (See Result example below).

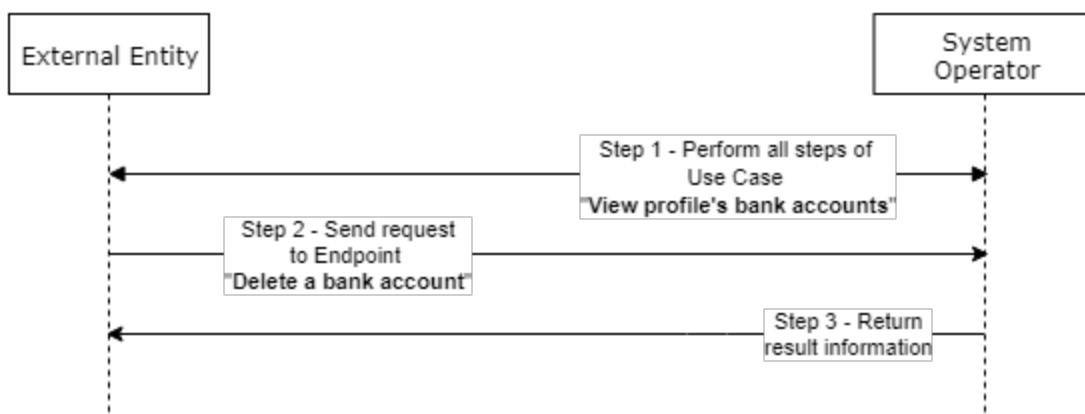
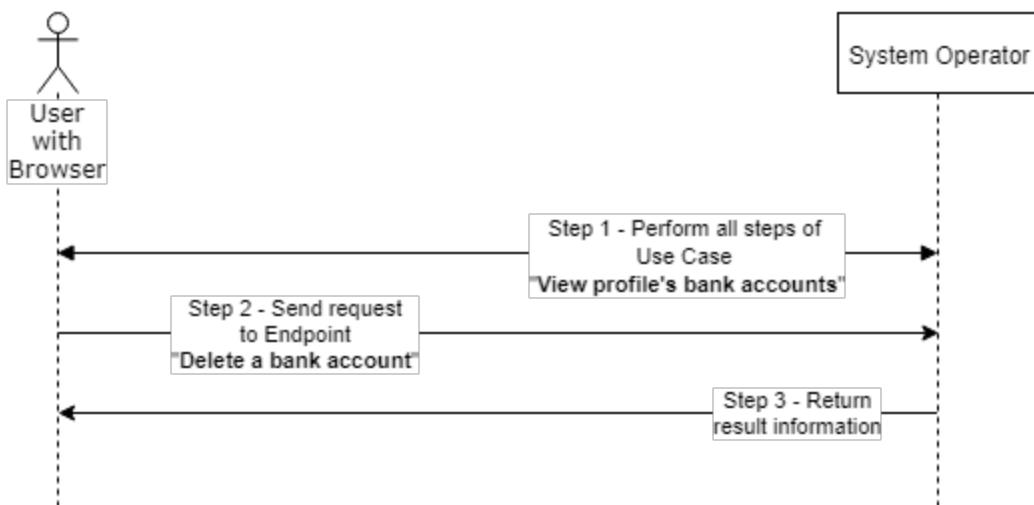
Post Conditions

Bank account is deleted..

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete a bank account scheme

Use case: Delete a bank account**Basic FFlow****Optional Web UI Flow**

Update a bank account description

Use Case Name

Update a bank account

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “View profile's bank accounts” Use Case, and then send a request to Endpoint “Update a bank account”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View profile's bank accounts”.
2. External Entity sends a request to Endpoint “Update a bank account”.

Endpoint URL: PATCH /profiles/my/bank-accounts/{id}

Parameters:

```
{
  "accountDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View profile's bank accounts”.
2. A user sends a request to Endpoint “Update a bank account”.

Endpoint URL: PATCH /profiles/my/bank-accounts/{id}

Parameters:

```
{
  "accountDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator returns result information to User (See Result example below).

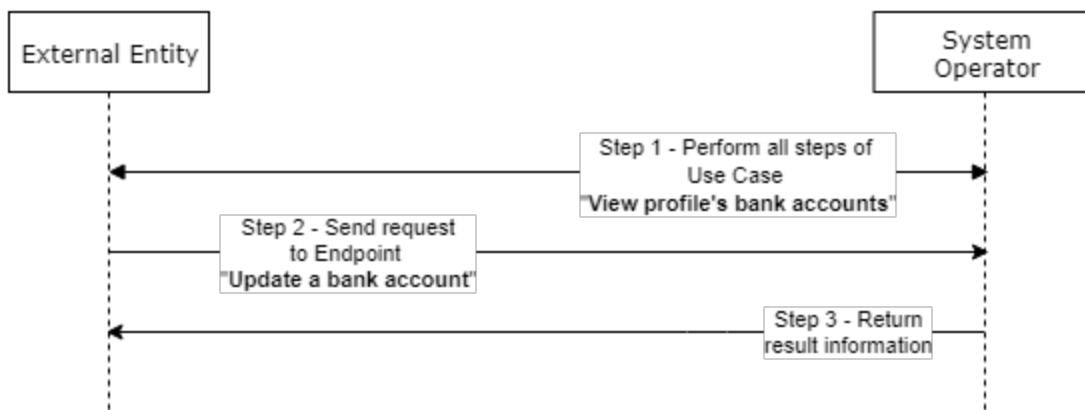
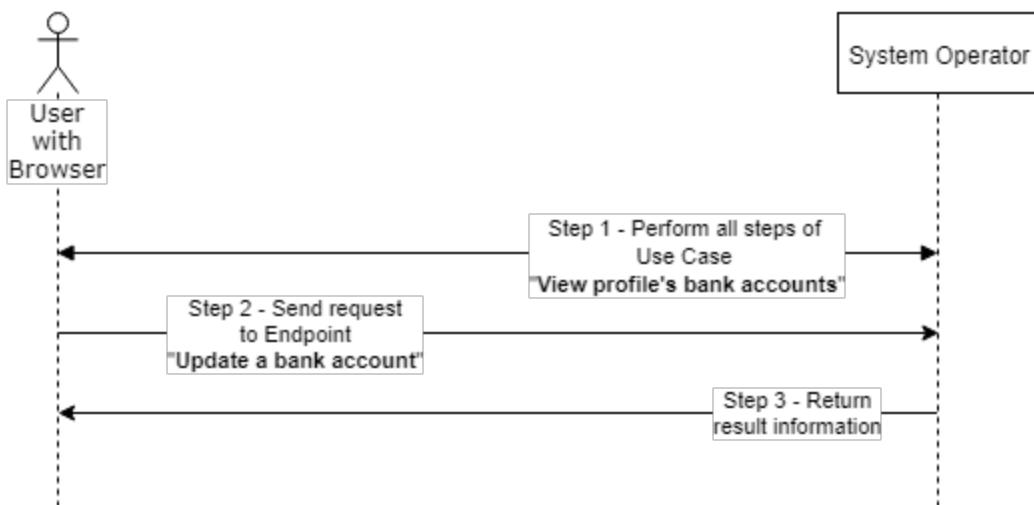
Post Conditions

The bank account is updated.

Result example

```
{
  "account": {
    "id": "string",
    "createdAt": "2018-08-07T14:16:19.147Z",
    "updatedAt": "2018-08-07T14:16:19.147Z",
    "status": "PENDING",
    "details": {
      "fullName": "string",
      "account": "string",
      "iban": "string",
      "bic": "string",
      "swift": "string",
      "name": "string",
      "address": "string"
    }
  },
  "status": "ok",
  "message": "string"
}
```

Update a bank account scheme

Use case: Update a bank account**Basic FFlow****Optional Web UI Flow**

View profile's bank accounts description

Use Case Name

View profile's bank accounts

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View profile's bank accounts”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View profile's bank accounts”.

Endpoint URL: GET /profiles/my/bank-accounts

Parameter: Security TOKEN only

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View profile's bank accounts”.

Endpoint URL: GET /profiles/my/bank-accounts

Parameter: Security TOKEN only

3. System Operator returns xx result information to User (See Result example below).

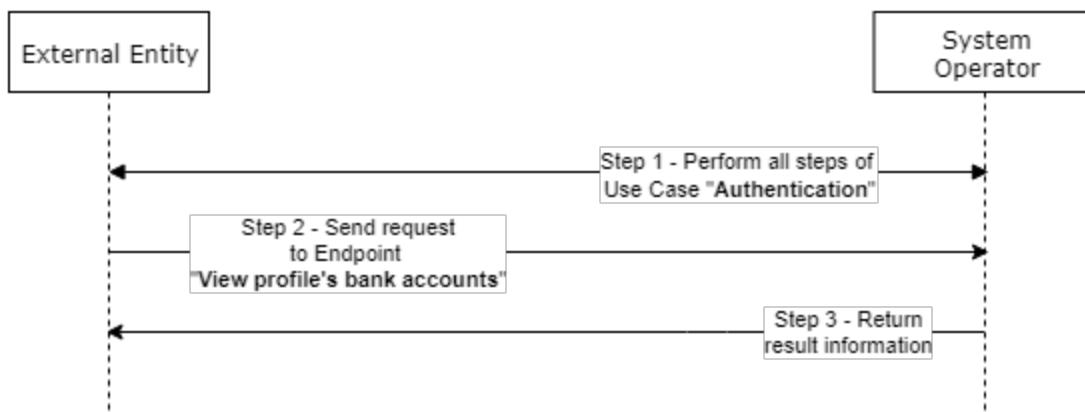
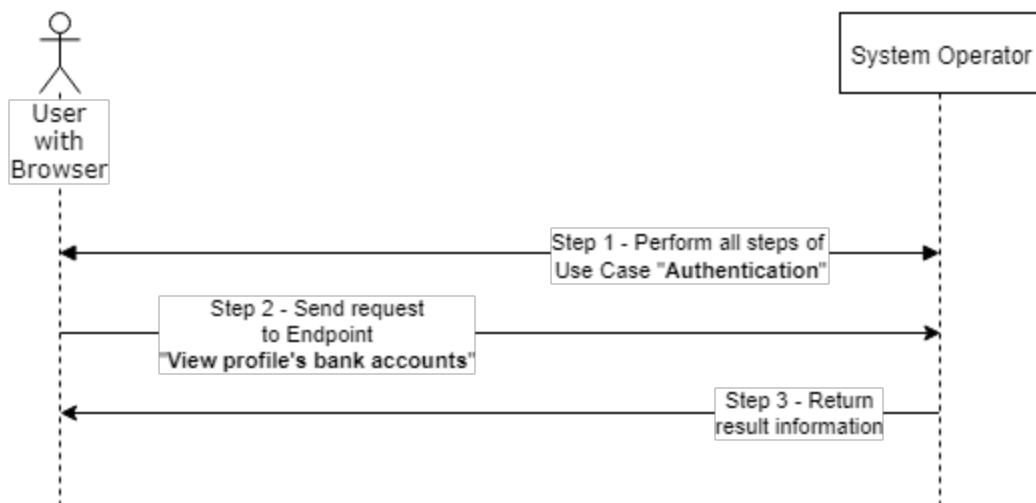
Post Conditions

List of bank accounts is available

Result example

```
{  
    "List of bank accounts": [  
        {  
            "ID of bank account": "string",  
            "createdAt": "2018-08-07T13:12:40.950Z",  
            "updatedAt": "2018-08-07T13:12:40.950Z",  
            "status": "PENDING",  
            "details": {  
                "fullName": "string",  
                "account": "string",  
                "iban": "string",  
                "bic": "string",  
                "swift": "string",  
                "name": "string",  
                "address": "string"  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

View profile's bank accounts scheme

Use case: View profile's bank accounts**Basic FFlow****Optional Web UI Flow**

Profile - documents approving

Approve the profile document description

Use Case Name

Approve the profile document

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Approve the profile document”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View all profile documents”.
2. External Entity sends a request to Endpoint “Approve the profile document”.

Endpoint URL: POST /profile-documents/{id}/approve

Parameter: Security TOKEN

System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View all profile documents”.
 2. A user sends a request to Endpoint “Approve the profile document”.
- Endpoint URL: POST /profile-documents/{id}/approve
- Parameter: Security TOKEN
3. System Operator returns result information to External Entity. (See Result example below)

Post Conditions

Document status has changed from PENDING to APPROVED

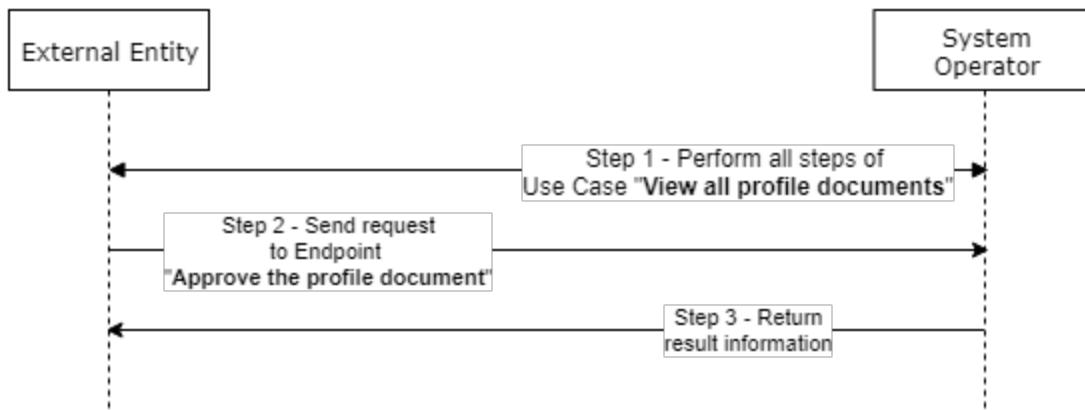
Result example

```
{  
  "document": {  
    "id": "string",  
    "file": {  
      "id": "string",  
      "ownerId": "string",  
      "mediaType": "string",  
      "name": "string",  
      "url": "string",  
      "md5": "string",  
      "sha1": "string",  
      "size": 0,  
      "used": false,  
      "createdAt": "2018-07-25T12:12:33.919Z",  
      "expiresAt": "2018-07-25T12:12:33.919Z",  
      "tag": "string"  
    },  
    "type": "string",  
    "label": "string",  
    "status": "APPROVED",  
    "updatedAt": "2018-07-25T12:12:33.919Z"  
  },  
}  
}
```

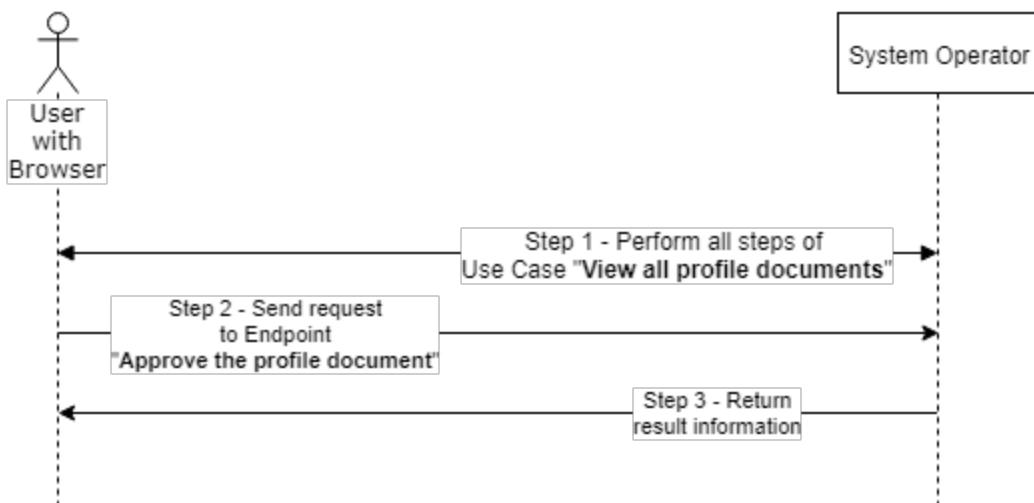
Approve the profile document scheme

Use case: Approve the profile document

Basic FFlow



Optional Web UI Flow



Decline the profile document description

Use Case Name

Decline the profile document

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Decline the profile document”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View all profile documents”.
2. External Entity sends a request to Endpoint “Decline the profile document”.

Endpoint URL: POST /profile-documents/{id}/decline

Parameter: Security TOKEN

System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View all profile documents”.
2. A user sends a request to Endpoint “Decline the profile document”.

Endpoint URL: POST /profile-documents/{id}/decline

Parameter: Security TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Post Conditions

Document status has changed from PENDING to DECLINED

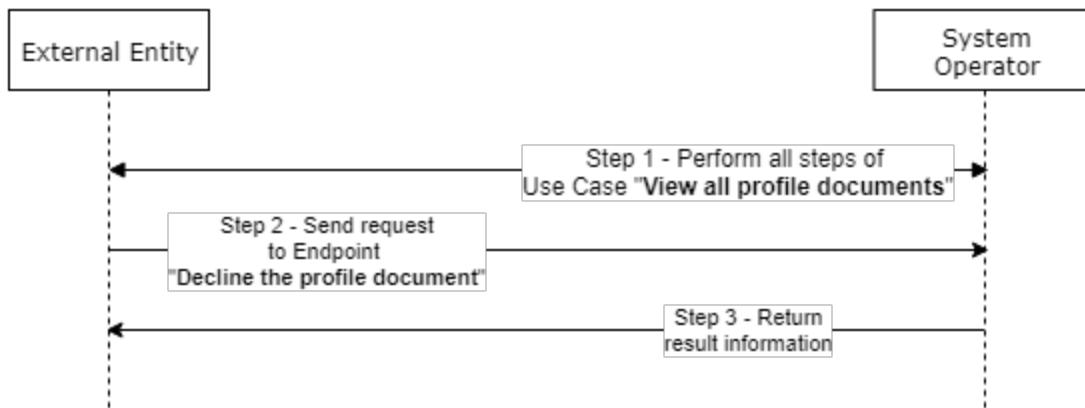
Result example

```
{  
  "document": {  
    "id": "string",  
    "file": {  
      "id": "string",  
      "ownerId": "string",  
      "mediaType": "string",  
      "name": "string",  
      "url": "string",  
      "md5": "string",  
      "sha1": "string",  
      "size": 0,  
      "used": false,  
      "createdAt": "2018-07-25T12:12:33.922Z",  
      "expiresAt": "2018-07-25T12:12:33.922Z",  
      "tag": "string"  
    },  
    "type": "string",  
    "label": "string",  
    "status": "PENDING",  
    "updatedAt": "2018-07-25T12:12:33.922Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

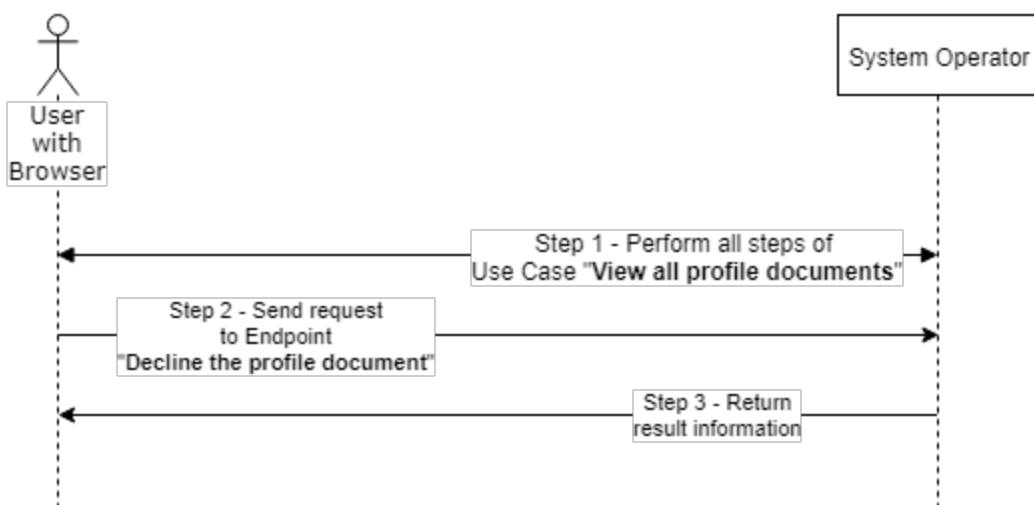
Decline the profile document scheme

Use case: Decline the profile document

Basic FFlow



Optional Web UI Flow



View all profile documents description

Use Case Name

View all profile documents

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View all profile documents”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View all profile documents”.

Endpoint URL: POST /profile-documents/view

Parameter:

```
{
  "filter": {
    "orgIds": [
      "string"
    ],
    "statuses": [
      "PENDING"
    ],
    "types": [
      "string"
    ]
  },
  "sort": {
    "status": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns xx to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “View all profile documents”.

Endpoint URL: xx

Parameter: xx Security TOKEN.

3. System Operator returns xx to User (See Result example below).

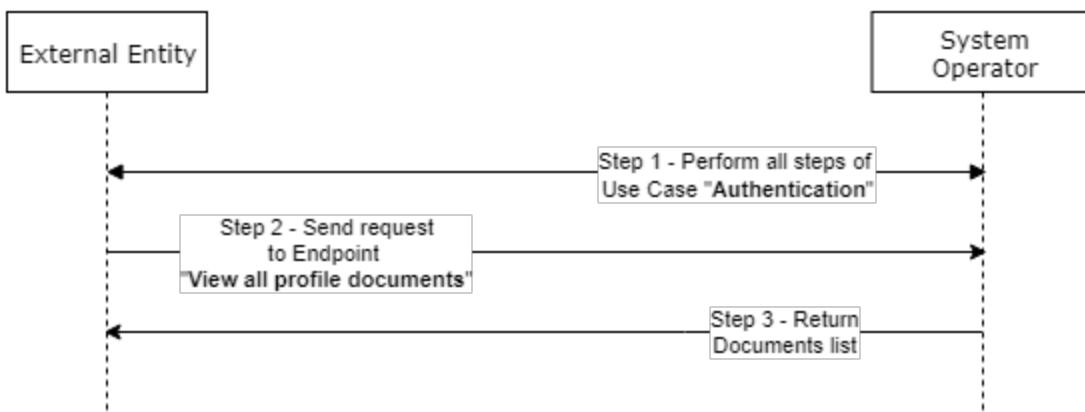
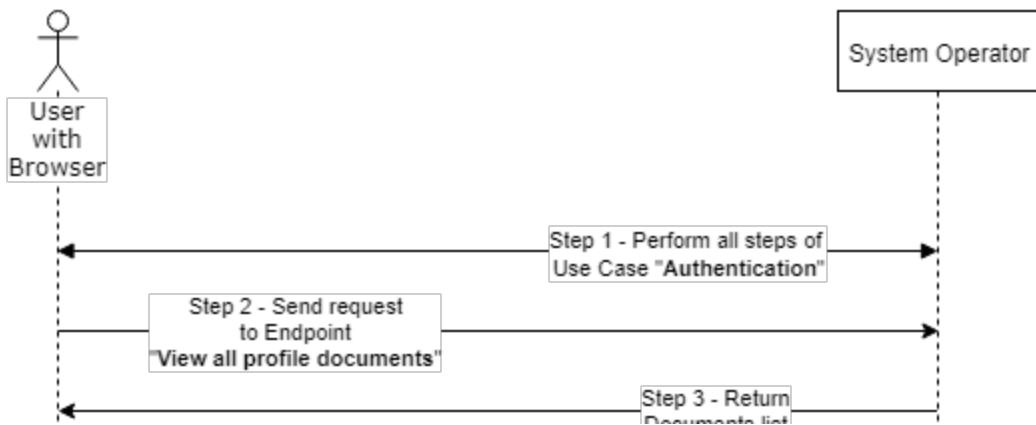
Post Conditions

All profile documents are available

Result example

```
{
  "records": [
    {
      "id": "string",
      "file": {
        "id": "string",
        "ownerId": "string",
        "mediaType": "string",
        "name": "string",
        "url": "string",
        "md5": "string",
        "sha1": "string",
        "size": 0,
        "used": false,
        "createdAt": "2018-07-25T12:12:33.912Z",
        "expiresAt": "2018-07-25T12:12:33.912Z",
        "tag": "string"
      },
      "type": "string",
      "label": "string",
      "status": "PENDING",
      "updatedAt": "2018-07-25T12:12:33.912Z"
    }
  ],
  "status": "ok",
  "message": "string"
}
```

View all profile documents scheme

Use case: View all profile documents**Basic FLow****Optional Web UI Flow**



Profile - documents uploading

Get all document types required for approval description

Use Case Name

Get all document types required for approval

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get all document types required for approval”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get all document types required for approval”.

Endpoint URL: POST /profile-documents/view-document-types

Parameter: Security TOKEN.

3. System Operator returns List of required types to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get all document types required for approval”.

Endpoint URL: POST /profile-documents/view-document-types

Parameter: Security TOKEN.

3. System Operator returns List of required types to External Entity. (See Result example below)

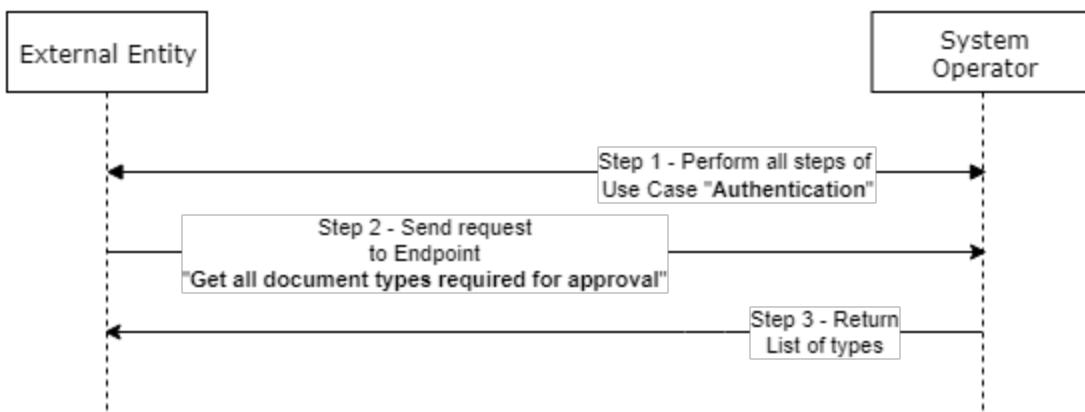
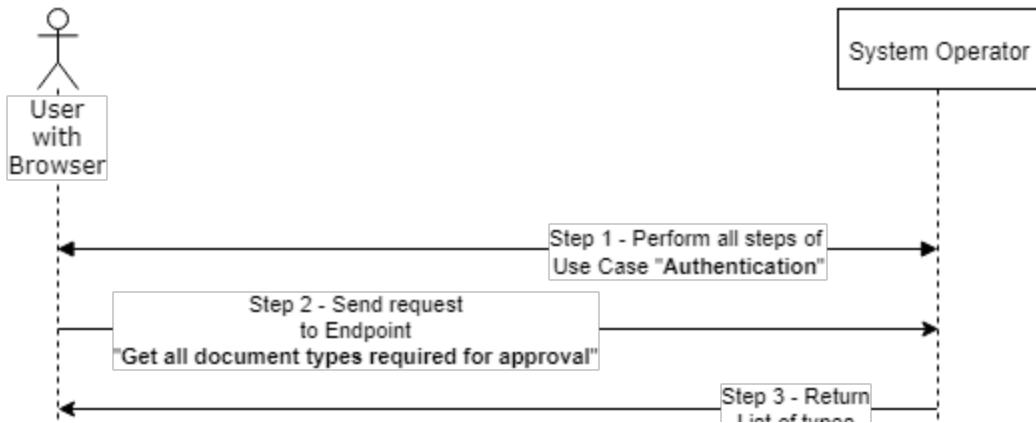
Post Conditions

Required list of documents is available.

Result example

```
{  
  "documentTypes": [  
    {  
      "type": "string",  
      "label": "string",  
      "optional": false  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

Get all document types required for approval scheme

Use case: Get all document types required for approval**Basic FLow****Optional Web UI Flow**



LIST OF TYPES

Get profile documents, last of each type description

Use Case Name

Get profile documents, last of each type

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get profile documents, last of each type”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_OWNER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get profile documents, last of each type”.

Endpoint URL: GET /profile-documents

Parameter: Security TOKEN.

3. System Operator returns List of requested documents to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get profile documents, last of each type”.

Endpoint URL: GET /profile-documents

Parameter: Security TOKEN.

3. System Operator returns List of requested documents to External Entity. (See Result example below)

Post Conditions

Proper set of documents is available.

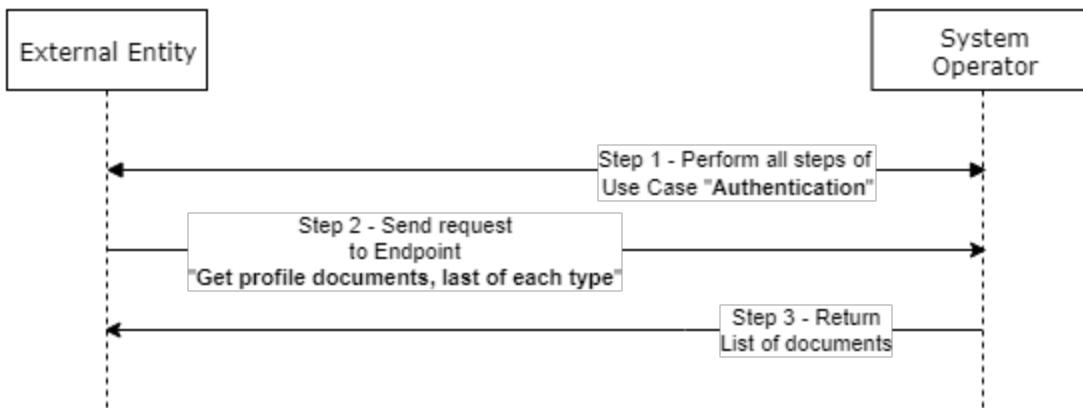
Result example

```
{  
    "documents": [  
        {  
            "id": "string",  
            "file": {  
                "id": "string",  
                "ownerId": "string",  
                "mediaType": "string",  
                "name": "string",  
                "url": "string",  
                "md5": "string",  
                "sha1": "string",  
                "size": 0,  
                "used": false,  
                "createdAt": "2018-07-25T12:12:33.900Z",  
                "expiresAt": "2018-07-25T12:12:33.900Z",  
                "tag": "string"  
            },  
            "type": "string",  
            "label": "string",  
            "status": "PENDING",  
            "updatedAt": "2018-07-25T12:12:33.900Z"  
        },  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

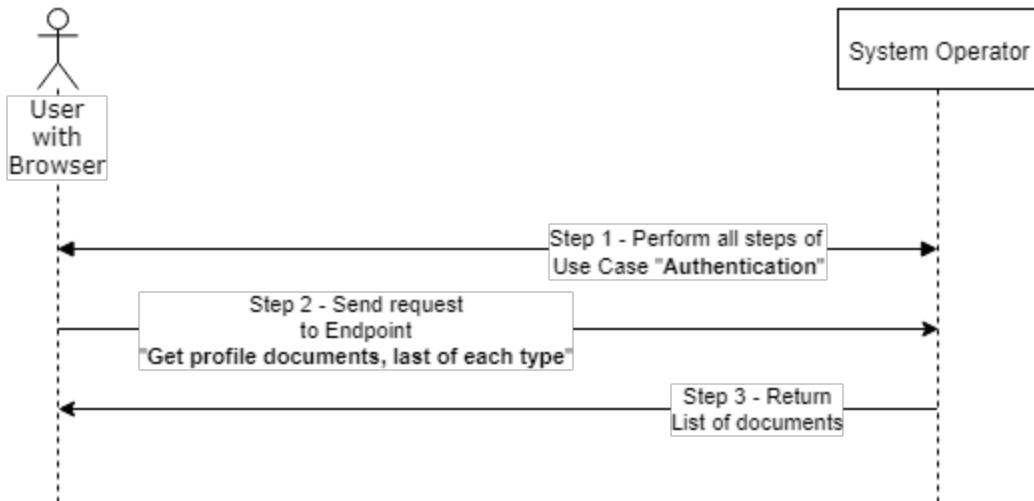
Get profile documents, last of each type scheme

Use case: Get profile documents, last of each type

Basic FFlow



Optional Web UI Flow



Submit file for approval as profile document description

Use Case Name

Submit file for approval as the profile document

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_DOCUMENTS_OWNER will go through all steps of “Get profile documents, last of each type” Use Case, and then send a request to Endpoint “Submit file for approval as profile document”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_DOCUMENTS_OWNER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get profile documents, last of each type”.
2. External Entity sends a request to Endpoint “Submit file for approval as a profile document”.

Endpoint URL: POST /profile-documents

Parameter:

```
{
  "fileId": "string",
  "type": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get profile documents, last of each type”.
2. A user sends a request to Endpoint “Submit file for approval as profile document”.

Endpoint URL: POST /profile-documents

Parameter:

```
{
  "fileId": "string",
  "type": "string"
}
```

3. System Operator returns result information to User (See Result example below).

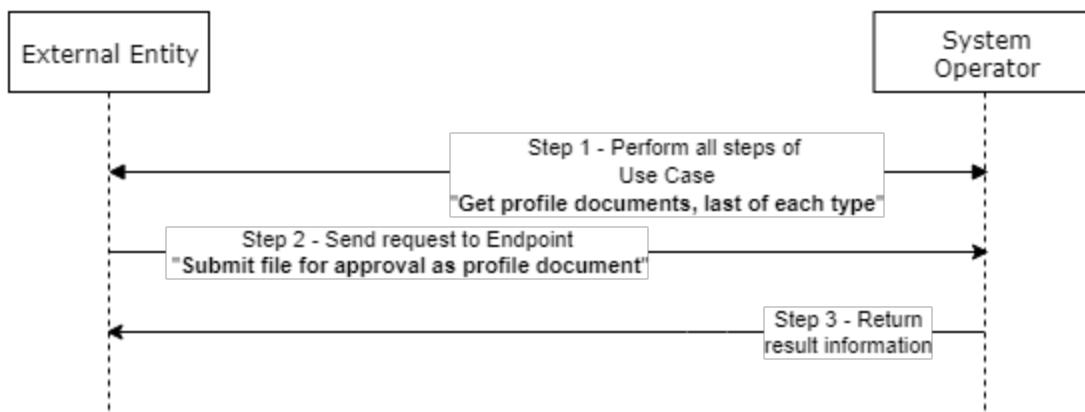
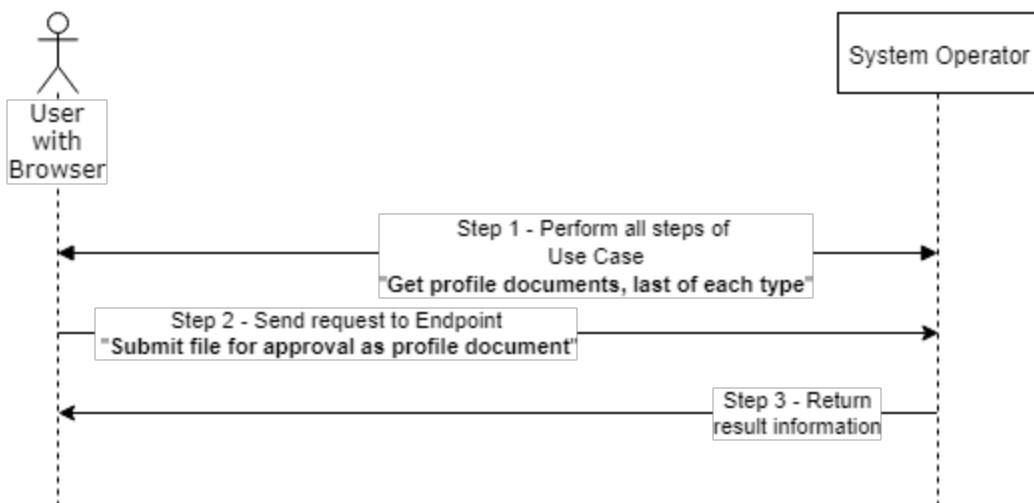
Post Conditions

The file is submitted.

Result example

```
{  
  "document": {  
    "id": "string",  
    "file": {  
      "id": "string",  
      "ownerId": "string",  
      "mediaType": "string",  
      "name": "string",  
      "url": "string",  
      "md5": "string",  
      "sha1": "string",  
      "size": 0,  
      "used": false,  
      "createdAt": "2018-07-25T12:12:33.903Z",  
      "expiresAt": "2018-07-25T12:12:33.903Z",  
      "tag": "string"  
    },  
    "type": "string",  
    "label": "string",  
    "status": "PENDING",  
    "updatedAt": "2018-07-25T12:12:33.903Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

Submit file for approval as profile document scheme

Use case: Submit file for approval as profile document**Basic FFlow****Optional Web UI Flow**

Profile - documents verification

Verify a profile document description

Use Case Name

Verify a profile document

Brief Description

A User or External Entity on behalf of a User with role permission DOCUMENT_VALIDATOR will go through all steps of “View not verified documents” Use Case, and then send a request to Endpoint “Verify a profile document”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: DOCUMENT_VALIDATOR R
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View not verified documents”.
2. External Entity sends a request to Endpoint “Verify a profile document”.

Endpoint URL: GET /profile-documents/{id}/verify

Parameter: Security TOKEN.

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View not verified documents”.
2. A user sends a request to Endpoint “Verify a profile document”.

Endpoint URL: GET /profile-documents/{id}/verify

Parameter: Security TOKEN.

3. System Operator returns result information to User (See Result example below).

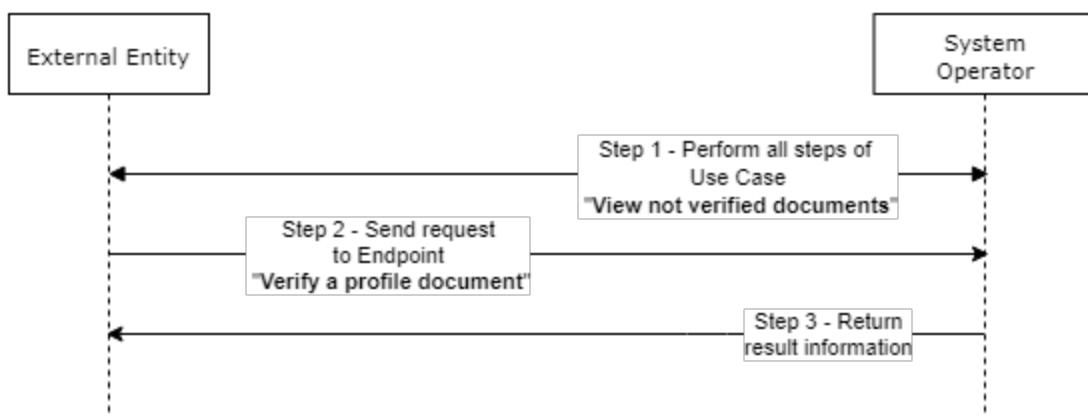
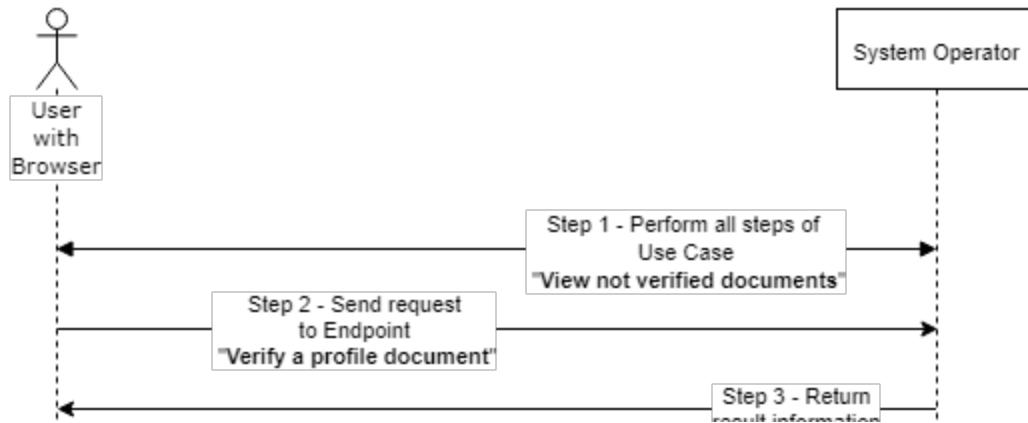
Post Conditions

List of test is available

Result example

```
{  
    "verifiedDocument": {  
        "id": "string",  
        "profileDocumentId": "string",  
        "verified": false,  
        "message": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Verify a profile document scheme

Use case: Verify a profile document**Basic FLow****Optional Web UI Flow**



View forgery tests of verified document description

Use Case Name

View forgery tests of verified document

Brief Description

A User or External Entity on behalf of a User with role permission DOCUMENT_VALIDATOR will go through all steps of “View last verified documents” Use Case, and then send a request to Endpoint “View forgery tests of the verified document”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: DOCUMENT_VALIDATOR R
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View last verified documents”.
2. External Entity sends a request to Endpoint “View forgery tests of the verified document”.

Endpoint URL: GET/profile-documents/{verifiedProfileDocumentId}/forgery-tests

Parameter: Security TOKEN.

3. System Operator returns List of tests to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View last verified documents”.
2. A user sends a request to Endpoint “View forgery tests of the verified document”.

Endpoint URL: GET/profile-documents/{verifiedProfileDocumentId}/forgery-tests

Parameter: Security TOKEN.

3. System Operator returns List of tests to User (See Result example below).

Post Conditions

List of test is available

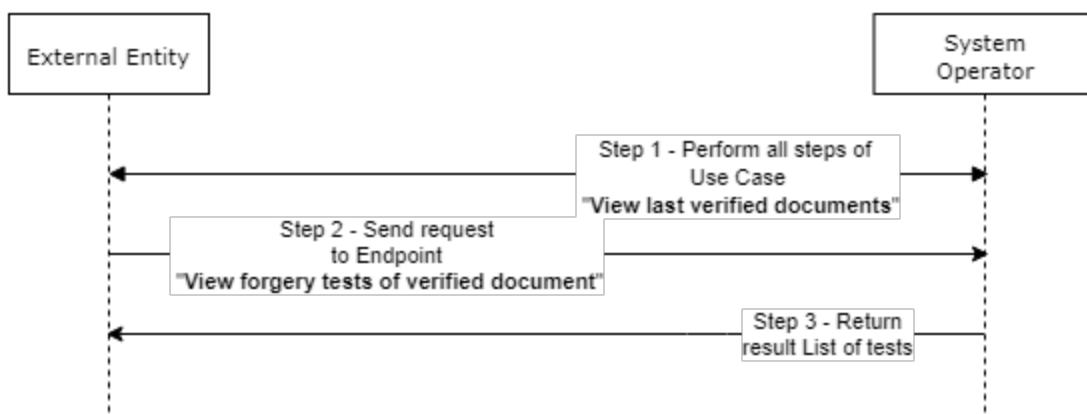
Result example

```
{  
    "records": [  
        {  
            "type": "string",  
            "result": "string"  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

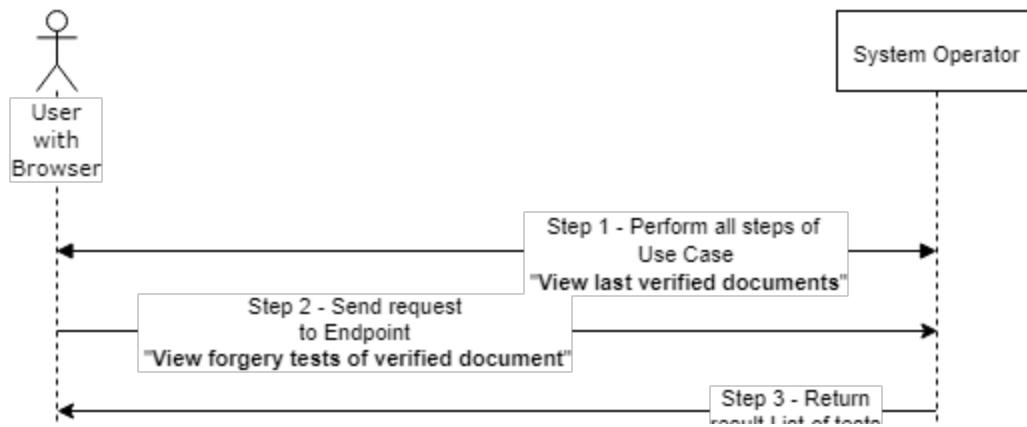
[View forgery tests of verified document scheme](#)

Use case: View forgery tests of verified document

Basic FLow



Optional Web UI Flow



View last verified documents description

Use Case Name

View last verified documents

Brief Description

A User or External Entity on behalf of a User with role permission DOCUMENT_VALIDATOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View last verified documents”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: DOCUMENT_VALIDATOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View last verified documents”.

Endpoint URL: GET /profile-documents/{userId}/view-verified

Parameter: Security TOKEN.

3. System Operator returns requested list to User (See Result example below).

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View last verified documents”.

Endpoint URL: GET /profile-documents/{userId}/view-verified

Parameter: Security TOKEN.

3. System Operator returns requested list to User (See Result example below).

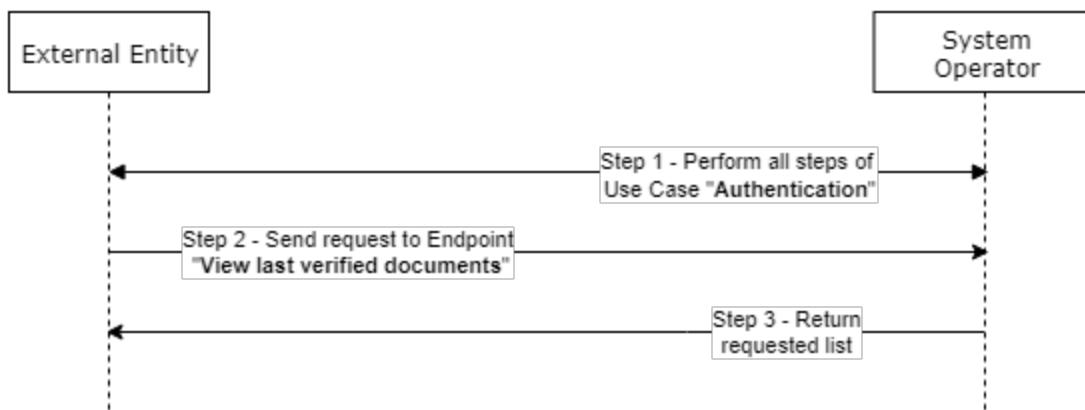
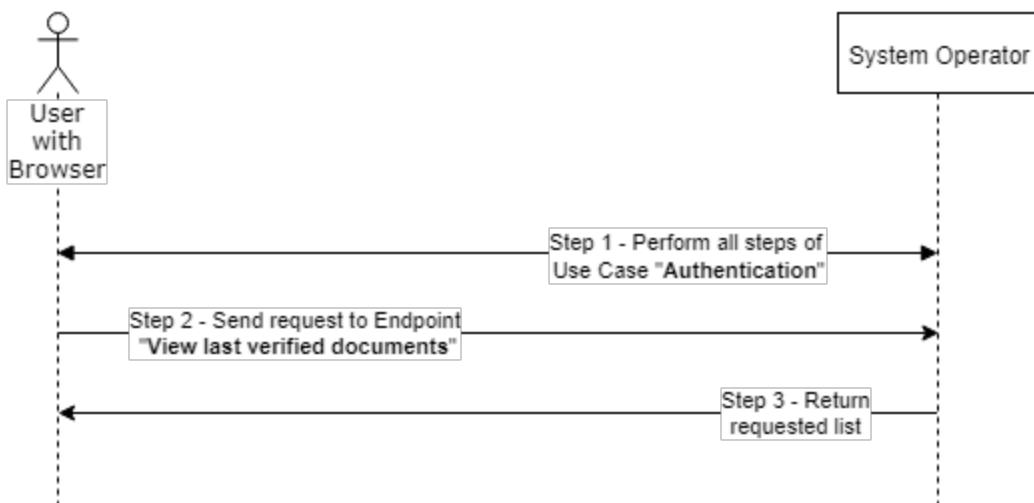
Post Conditions

Required list is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "profileDocumentId": "string",  
            "verified": false,  
            "message": "string"  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

[View last verified documents scheme](#)

Use case: View last verified documents**Basic FFlow****Optional Web UI Flow**

View not verified documents description

Use Case Name

View not verified documents

Brief Description

A User or External Entity on behalf of a User with role permission DOCUMENT_VALIDATOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View not verified documents”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: DOCUMENT_VALIDATOR R.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View not verified documents”.

Endpoint URL: GET /profile-documents/{userId}/view-not-verified

Parameter: Security TOKEN.

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View not verified documents”.

Endpoint URL: GET /profile-documents/{userId}/view-not-verified

Parameter: Security TOKEN.

3. System Operator returns xx result information to User (See Result example below).

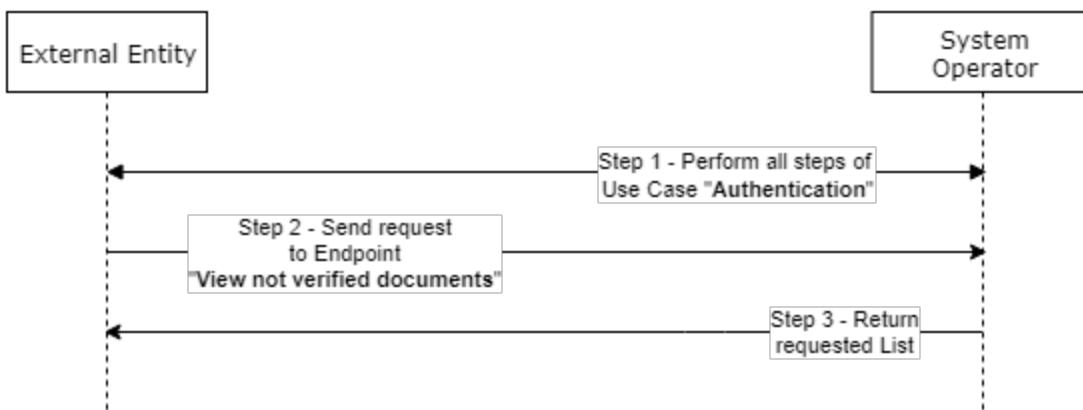
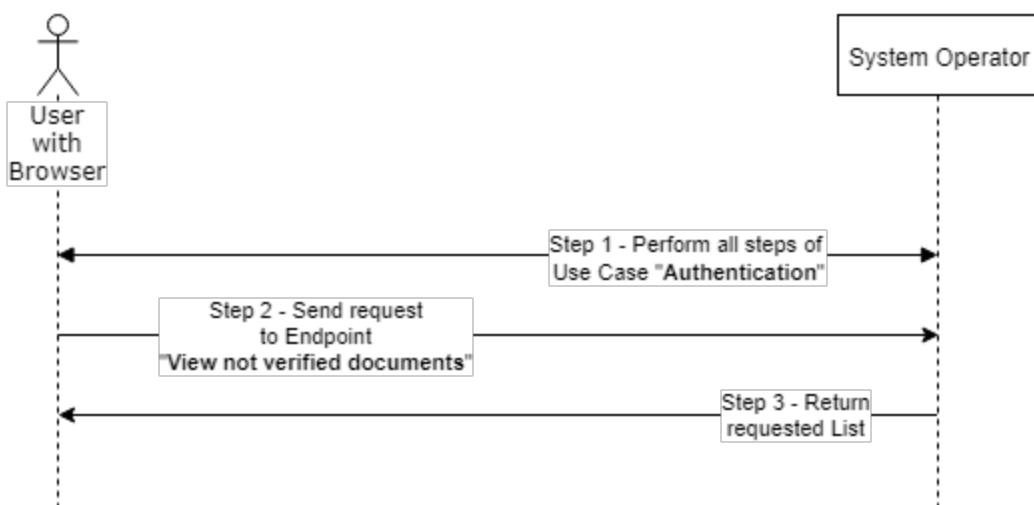
Post Conditions

The list is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "file": {  
                "id": "string",  
                "ownerId": "string",  
                "mediaType": "string",  
                "name": "string",  
                "url": "string",  
                "md5": "string",  
                "sha1": "string",  
                "size": 0,  
                "used": false,  
                "createdAt": "2018-07-25T12:12:33.933Z",  
                "expiresAt": "2018-07-25T12:12:33.933Z",  
                "tag": "string"  
            },  
            "type": "string",  
            "label": "string",  
            "status": "PENDING",  
            "updatedAt": "2018-07-25T12:12:33.933Z"  
        },  
        ],  
        "status": "ok",  
        "message": "string"  
    }  
}
```

[View not verified documents scheme](#)

Use case: View not verified documents**Basic FFlow****Optional Web UI Flow**

Media files management

Delete an unused file description

Use Case Name

Delete an unused file

Brief Description

A User or External Entity on behalf of a User with role permission MEDIA_FILE_OWNER will go through all steps of “Get media files by specified filter” Use Case, and then send a request to Endpoint “Delete an unused file”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MEDIA_FILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get media files by specified filter”.
2. External Entity sends a request to Endpoint “Delete an unused file”.

Endpoint URL: DELETE /media-files/{id}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get media files by specified filter”.

2. A user sends a request to Endpoint “Delete an unused file”.

Endpoint URL: DELETE /media-files/{id}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

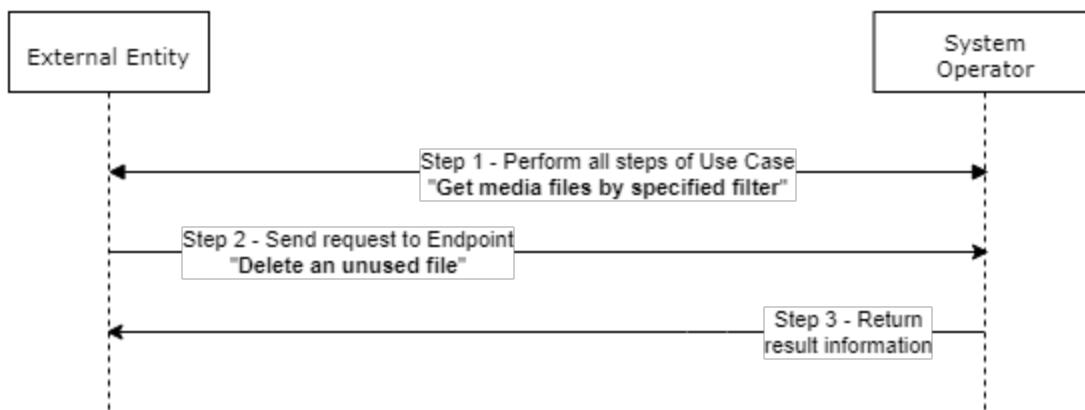
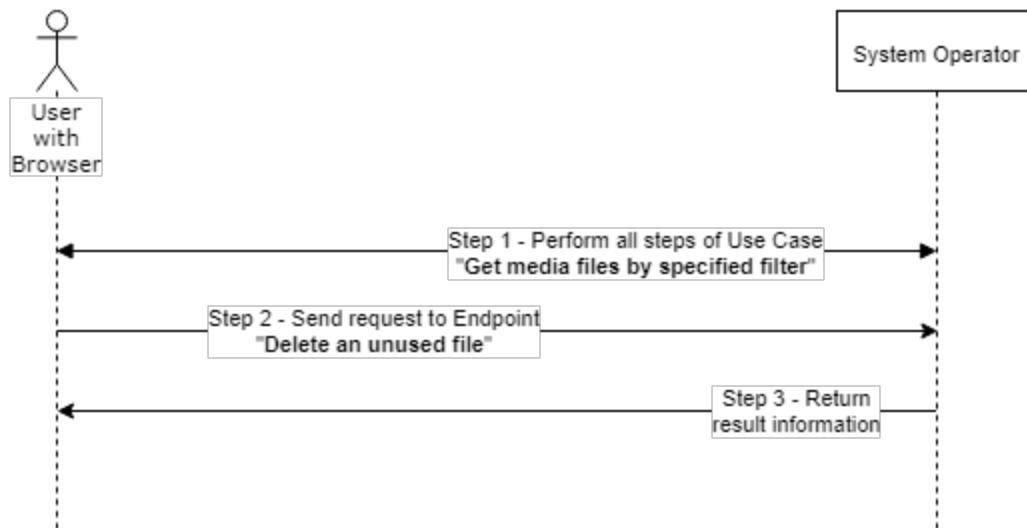
Post Conditions

File deleted.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete an unused file scheme

Use case: Delete an unused file**Basic FFlow****Optional Web UI Flow**

Get media files by specified filter description

Use Case Name

Get media files by the specified filter

Brief Description

A User or External Entity on behalf of a User with role permissions MEDIA_FILE_VIEWER, MEDIA_FILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get media files by specified filter”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MEDIA_FILE_VIEWER, MEDIA_FILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get media files by specified filter”.

Endpoint URL: POST /media-files/view

Parameters:

```
{
  "filter": {
    "id": "string",
    "name": "string",
    "types": [
      "string"
    ],
    "used": false,
    "fromExpiresAt": "2018-08-16T14:36:19.587Z",
    "toExpiresAt": "2018-08-16T14:36:19.589Z",
    "creationDateFrom": "2018-08-16T14:36:19.589Z",
    "creationDateTo": "2018-08-16T14:36:19.589Z"
  },
  "sort": {
    "creationDate": "asc",
    "expirationDate": "asc",
    "name": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns List of files (could be empty) to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get media files by specified filter”.

Endpoint URL: POST /media-files/view

Parameters:

```
{  
    "filter": {  
        "id": "string",  
        "name": "string",  
        "types": [  
            "string"  
        ],  
        "used": false,  
        "fromExpiresAt": "2018-08-16T14:36:19.587Z",  
        "toExpiresAt": "2018-08-16T14:36:19.589Z",  
        "creationDateFrom": "2018-08-16T14:36:19.589Z",  
        "creationDateTo": "2018-08-16T14:36:19.589Z"  
    },  
    "sort": {  
        "creationDate": "asc",  
        "expirationDate": "asc",  
        "name": "asc"  
    },  
    "pageNumber": 0,  
    "pageSize": 0  
}
```

3. System Operator returns List of files (could be empty) to User (See Result example below).

Post Conditions

List of files (could be empty)is available.

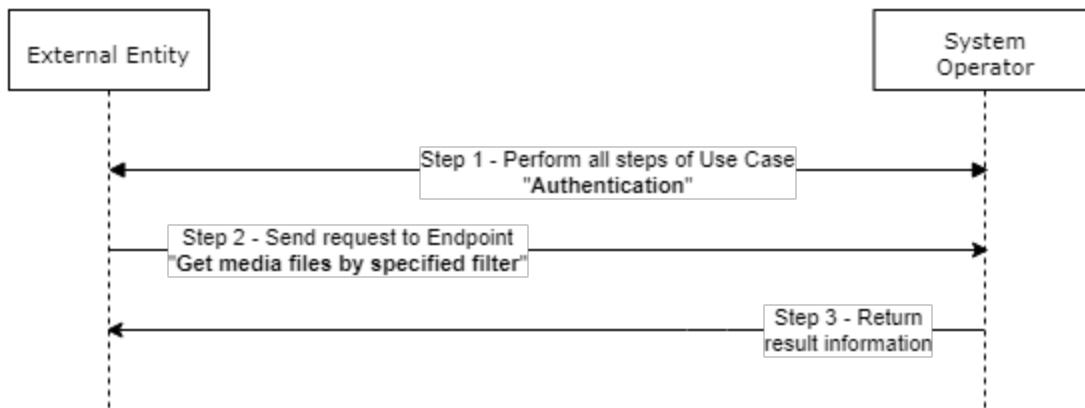
Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "ownerId": "string",  
            "mediaType": "string",  
            "name": "string",  
            "url": "string",  
            "md5": "string",  
            "sha1": "string",  
            "size": 0,  
            "used": false,  
            "createdAt": "2018-08-16T14:36:20.972Z",  
            "expiresAt": "2018-08-16T14:36:20.972Z",  
            "tag": "string"  
        }  
    ],  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0  
}
```

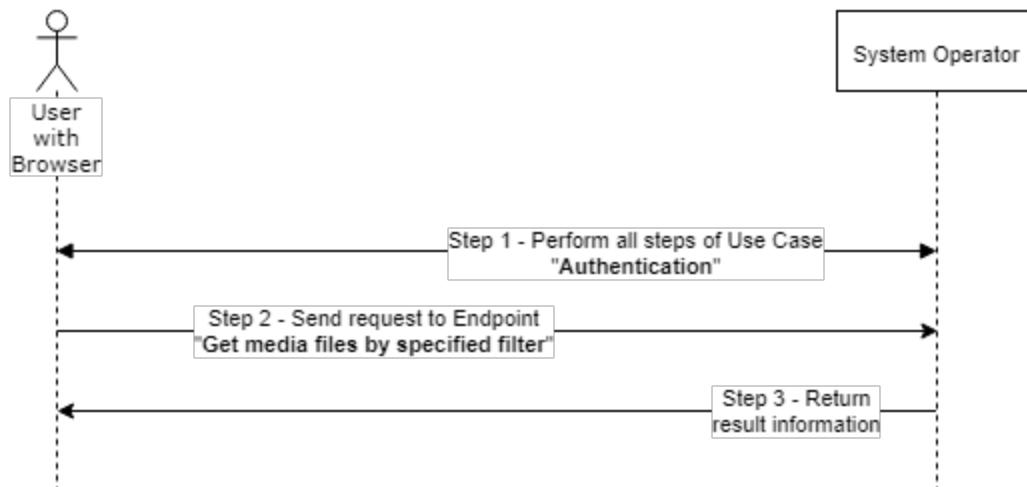
Get media files by specified filter scheme

Use case: Get media files by specified filter

Basic FFlow



Optional Web UI Flow



Update a file description

Use Case Name

Update a file

Brief Description

A User or External Entity on behalf of a User with role permission MEDIA_FILE_OWNER will go through all steps of “Get media files by specified filter” Use Case and then send a request to Endpoint “Update a file”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MEDIA_FILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get media files by specified filter”.
2. External Entity sends a request to Endpoint “Update a file”.

Endpoint URL: PATCH /media-files/{id}

Parameters:

```
{
  "tag": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get media files by specified filter”.
2. A user sends a request to Endpoint “Update a file”.

Endpoint URL: PATCH /media-files/{id}

Parameters:

```
{
  "tag": "string"
}
```

3. System Operator returns result information to User (See Result example below).

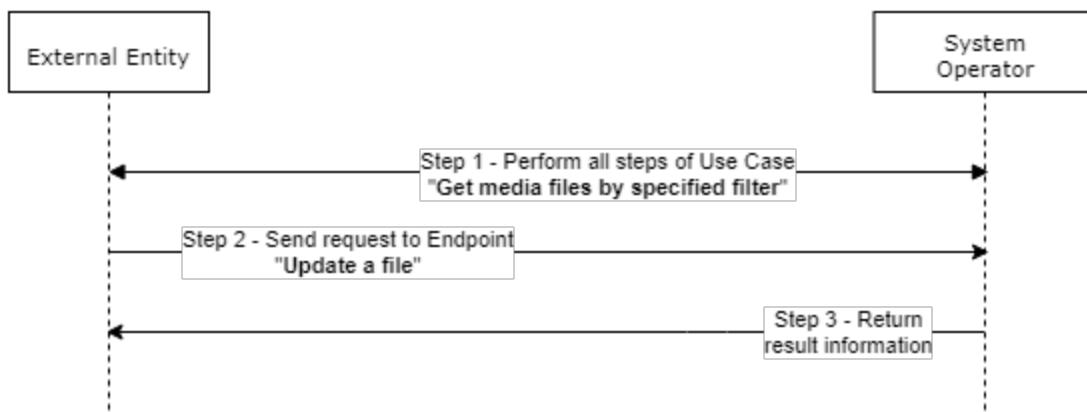
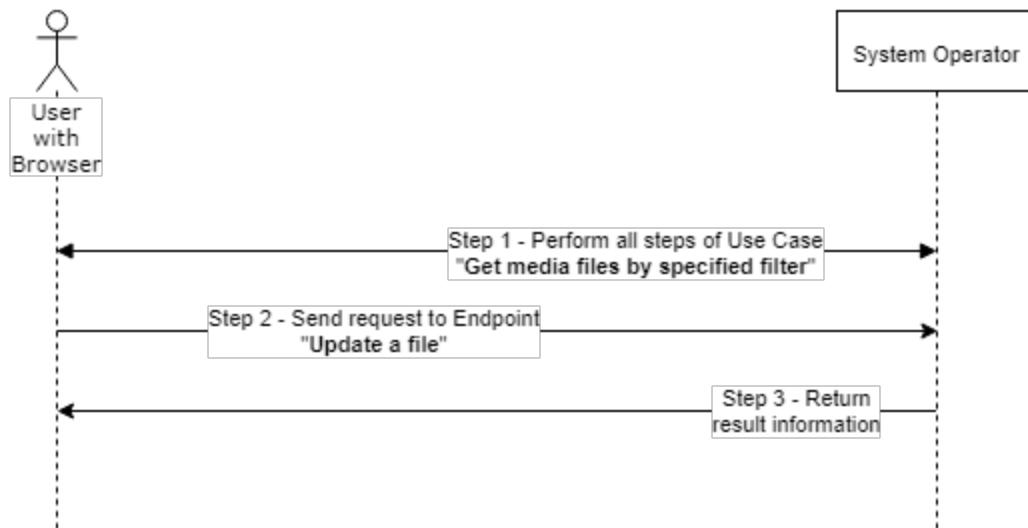
Post Conditions

The updated file is available.

Result example

```
{  
    "file": {  
        "id": "string",  
        "ownerId": "string",  
        "mediaType": "string",  
        "name": "string",  
        "url": "string",  
        "md5": "string",  
        "sha1": "string",  
        "size": 0,  
        "used": false,  
        "createdAt": "2018-08-16T14:36:20.982Z",  
        "expiresAt": "2018-08-16T14:36:20.982Z",  
        "tag": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Update a file scheme

Use case: Update a file**Basic FFlow****Optional Web UI Flow**

Upload a file description

Use Case Name

Upload a file

Brief Description

A User or External Entity on behalf of a User with role permission MEDIA_FILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Upload a file”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: MEDIA_FILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to Endpoint “Upload a file”.

Endpoint URL: POST /media-files

Parameters:

```
{
  "file_name": "string"
}
```

3. System Operator returns complete file information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “Upload a file”.

Endpoint URL: POST /media-files

Parameters:

```
{
  "file_name": "string"
}
```

3. System Operator returns complete file information to User (See Result example below).

Post Conditions

File uploaded and complete file information is available.

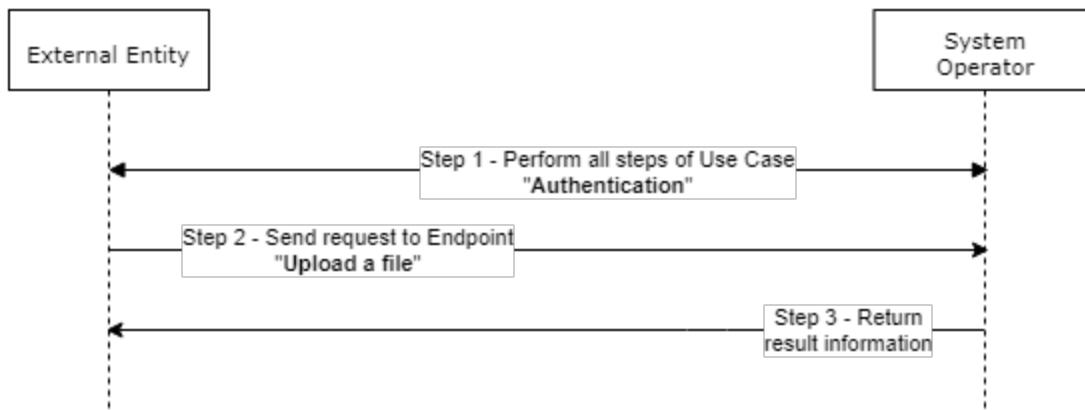
Result example

```
{  
    "file": {  
        "id": "string",  
        "ownerId": "string",  
        "mediaType": "string",  
        "name": "string",  
        "url": "string",  
        "md5": "string",  
        "sha1": "string",  
        "size": 0,  
        "used": false,  
        "createdAt": "2018-08-16T14:36:20.969Z",  
        "expiresAt": "2018-08-16T14:36:20.969Z",  
        "tag": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

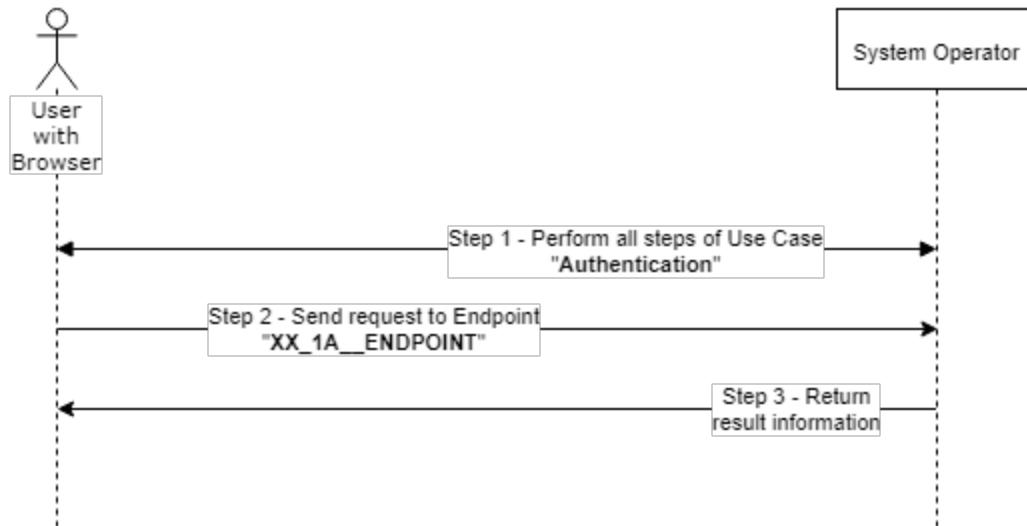
Upload a file scheme

Use case: Upload a file

Basic FFlow



Optional Web UI Flow



Profile - identification

Approve organization's identification description

Use Case Name

Approve the organization's identification

Brief Description

A User or External Entity on behalf of a User with role permission ORGANIZATION_STATUS_MANAGER will go through all steps of “Get users” Use Case, and then send a request to Endpoint “Approve organization's identification”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ORGANIZATION_STATUS_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Approve organization's identification”.

Endpoint URL: POST /profiles/{id}/approve

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Approve organization's identification”.

Endpoint URL: POST /profiles/{id}/approve

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

Identification is approved.

Result example

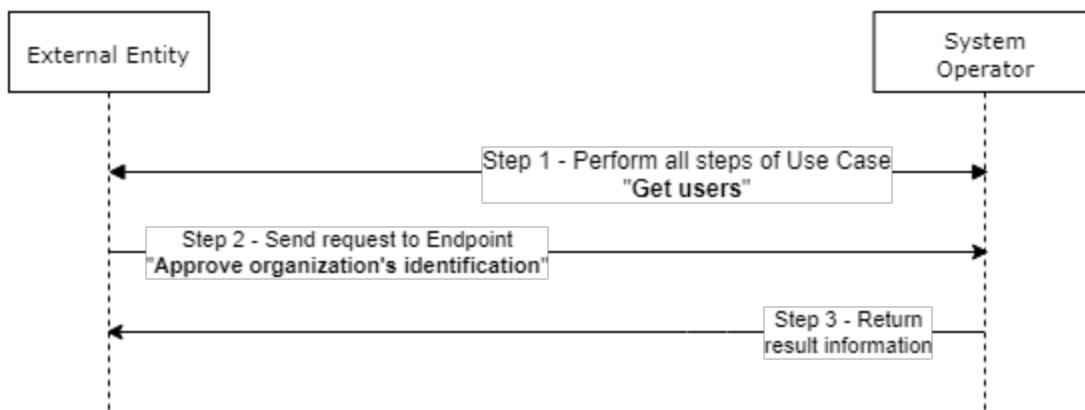
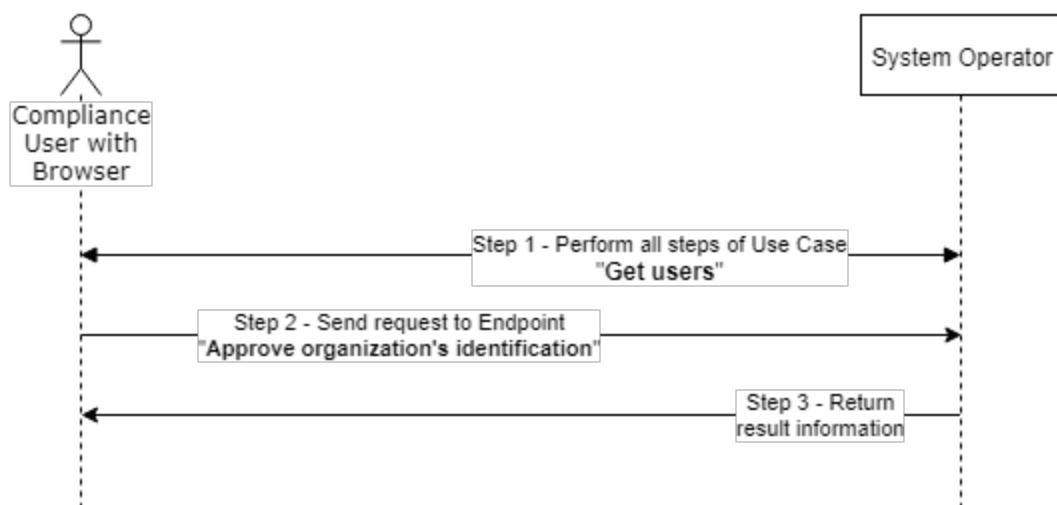
```
{
  "profile": {
    "person": {
      "namePlain": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "nameIntl": {
        "first": "string",
        "last": "string",
        "middle": "string"
      }
    }
  }
}
```

```
        "description": "string"
    },
    "contact": {
        "id": "string",
        "nickname": "string",
        "firstName": "string",
        "lastName": "string",
        "address": "string",
        "city": "string",
        "country": "string",
        "postCode": "string",
        "email": "string",
        "phoneNumber": "string",
        "photoPath": "string",
        "socialMedias": [
            {
                "id": "string",
                "socialMediaType": "string",
                "socialMediaReference": "string"
            }
        ],
        "accountNumber": "string",
        "swiftBic": "string",
        "ownerId": "string",
        "linkedUserId": "string"
    },
    "type": "base",
    "status": "none",
    "business": {
        "companyName": "string",
        "legal": "individual",
        "type": "eshop",
        "vat": "string",
        "administrator": {
            "firstName": "string",
            "lastName": "string",
            "email": "string",
            "phone": "string"
        }
    },
    "address": {
        "country": "AD",
        "zipCode": "string",
        "city": "string",
        "street": "string",
        "houseNumber": "string"
    },
    "additional": {
    }
},
```

```
    "security":{  
        "twoFactorsAuthEnabled":false,  
        "transactionNotification":{  
            "phone":false,  
            "email":false  
        },  
        "authorizationNotification":{  
            "phone":false,  
            "email":false  
        }  
    },  
},
```

```
"status": "ok",
"message": "string"
}
```

Approve organization's identification scheme

Use case: Approve organization's identification**Basic FFlow****Optional Web UI Flow**

Decline a reset identification request of user description

Use Case Name

Decline a reset identification request of the user.

Brief Description

A User or External Entity on behalf of a User with role permission RESET_IDENTIFICATION_INITIATOR will go through all steps of “View pending reset identification requests” Use Case, and then send a request to Endpoint “Decline a reset identification request of user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: RESET_IDENTIFICATION_INITIATOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View pending reset identification requests”.
2. External Entity sends a request to Endpoint “Decline a reset identification request of user”.

Endpoint URL: POST /profiles/{userId}/reset-identification-requests/{id}/decline

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View pending reset identification requests”.
2. A user sends a request to Endpoint “Decline a reset identification request of user”.

POST /profiles/{userId}/reset-identification-requests/{id}/decline

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

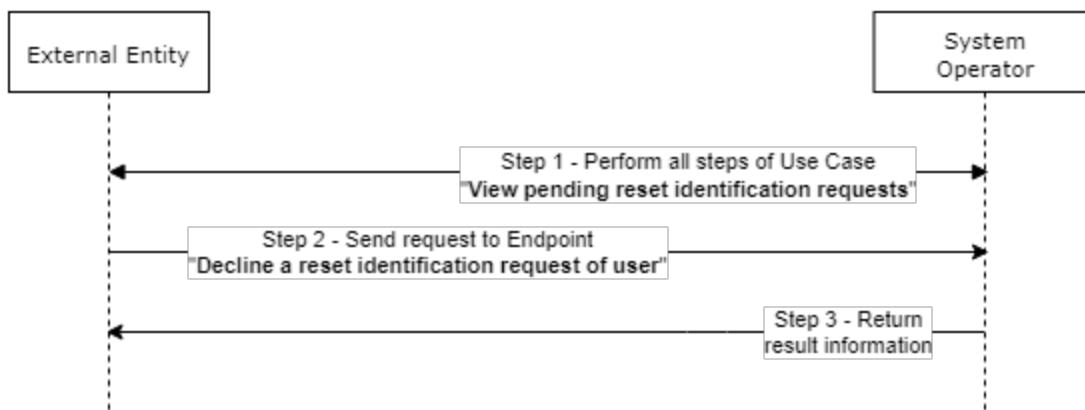
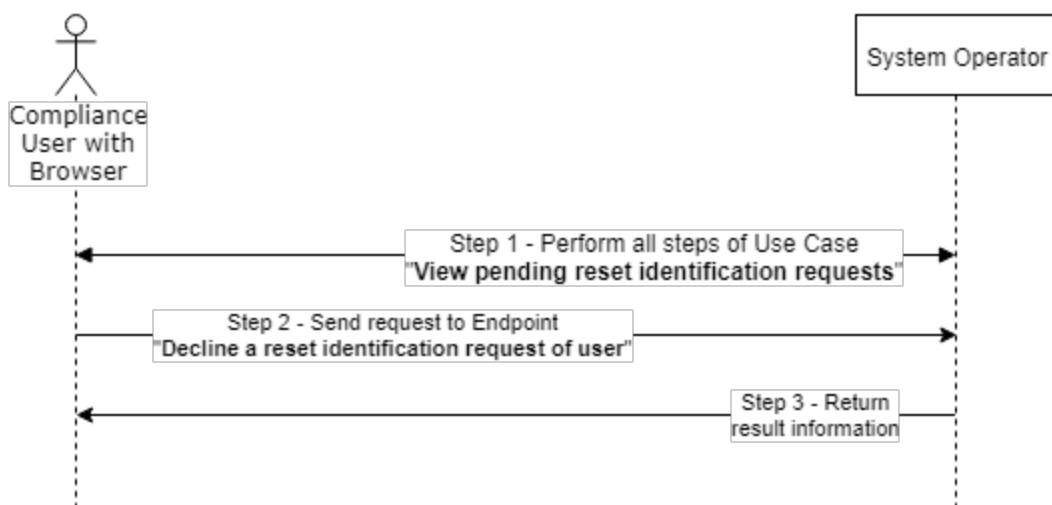
Post Conditions

Reset is declined.

Result example

```
{  
    "request": {  
        "id": "string",  
        "description": "string",  
        "organizationId": "string",  
        "status": "APPROVED",  
        "initiatorUserId": "string",  
        "initiatorRole": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Decline a reset identification request of user scheme

Use case: Decline a reset identification request of user**Basic FFlow****Optional Web UI Flow**

Decline organization's identification description

Use Case Name

Decline organization's identification

Brief Description

A User or External Entity on behalf of a User with role permission ORGANIZATION_STATUS_MANAGER will go through all steps of “Get users” Use Case and then send a request to Endpoint “Decline organization's identification”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
ORGANIZATION_STATUS_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Decline organization's identification”.

Endpoint URL: POST /profiles/{id}/decline

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Decline organization's identification”.

Endpoint URL: POST /profiles/{id}/decline

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

Identification is declined.

Result example

```
{
  "profile": {
    "person": {
      "namePlain": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "nameIntl": {
        "name": "string"
      }
    }
  }
}
```

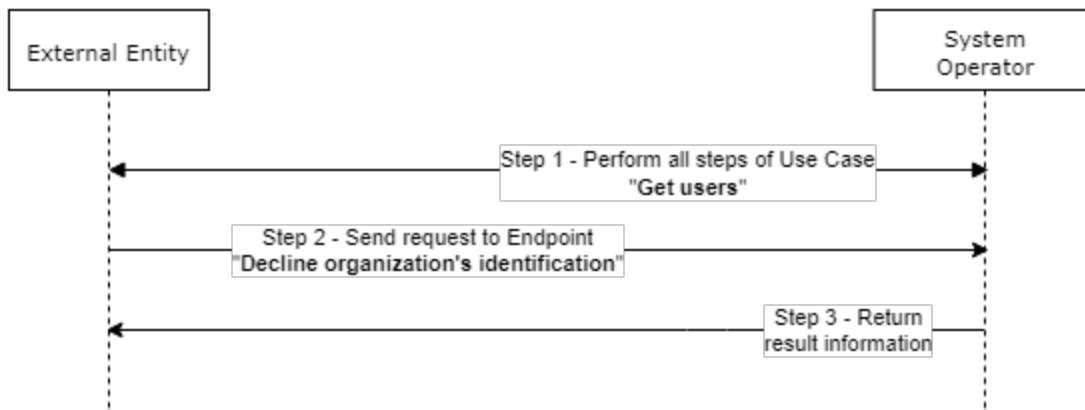
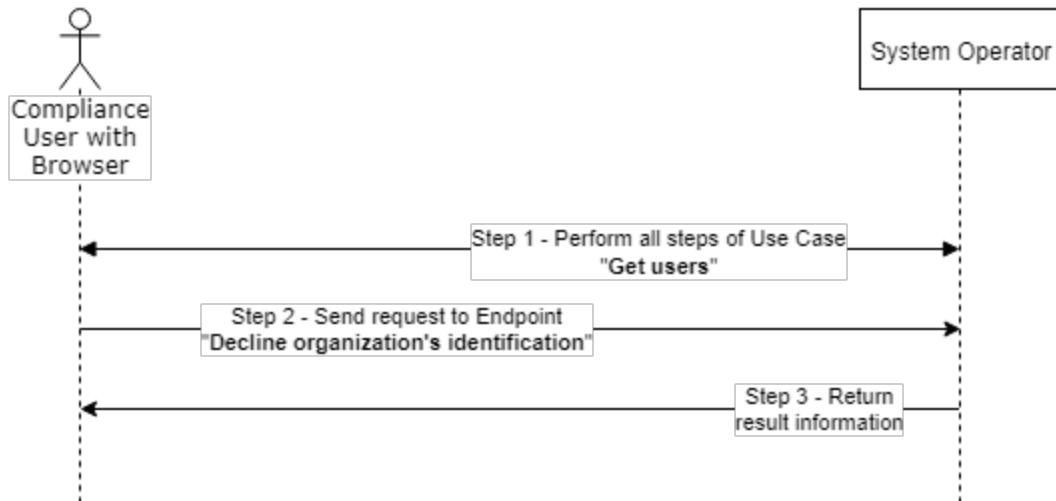
```
        "first":"string",
        "last":"string",
        "middle":"string"
    },
    "description":"string"
},
"contact":{
    "id":"string",
    "nickname":"string",
    "firstName":"string",
    "lastName":"string",
    "address":"string",
    "city":"string",
    "country":"string",
    "postCode":"string",
    "email":"string",
    "phoneNumber":"string",
    "photoPath":"string",
    "socialMedias": [
        {
            "id":"string",
            "socialMediaType":"string",
            "socialMediaReference":"string"
        }
    ],
    "accountNumber":"string",
    "swiftBic":"string",
    "ownerId":"string",
    "linkedUserId":"string"
},
"type":"base",
"status":"none",
"business": {
    "companyName":"string",
    "legal":"individual",
    "type":"eshop",
    "vat":"string",
    "administrator": {
        "firstName":"string",
        "lastName":"string",
        "email":"string",
        "phone":"string"
    }
},
"address": {
    "country":"AD",
    "zipCode":"string",
    "city":"string",
    "street":"string",
    "houseNumber":"string"
}
```

```
},
"additional":{

},
"security":{
    "twoFactorsAuthEnabled":false,
    "transactionNotification":{
        "phone":false,
        "email":false
    },
    "authorizationNotification":{
        "phone":false,
        "email":false
    }
}
},
```

```
"status": "ok",
"message": "string"
}
```

Decline organization's identification scheme

Use case: Decline organization's identification**Basic FLow****Optional Web UI Flow**

Reset an identification of user and send mandatory fields of profile to update for reidentification description

Use Case Name

Reset an identification of the user and send mandatory fields of the profile to update for re-identification

Brief Description

A User or External Entity on behalf of a User with role permission RESET_IDENTIFICATION_INITIATOR will go through all steps of “View pending reset identification requests” Use Case, and then send a request to Endpoint “Reset an identification of the user and send mandatory fields of the profile to update for re-identification”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: RESET_IDENTIFICATION_INITIATOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View pending reset identification requests”.
2. External Entity sends a request to Endpoint “Reset an identification of the user and send mandatory fields of the profile to update for re-identification”.

Endpoint URL: POST /profiles/{userId}/reset

Parameters:

```
{
  "fields": [
    "string"
  ]
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View pending reset identification requests”.
2. A user sends a request to Endpoint “Reset an identification of the user and send mandatory fields of the profile to update for re-identification”.

Endpoint URL: POST /profiles/{userId}/reset

Parameters:

```
{
  "fields": [
    "string"
  ]
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

Reset is done.

Result example

```
{
  "profile": {
    "person": {
      "namePlain": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "nameIntl": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "description": "string"
    },
    "contact": {
      "id": "string",
      "nickname": "string",
      "firstName": "string",
      "lastName": "string",
      "address": "string",
      "city": "string",
      "country": "string",
      "postCode": "string",
      "email": "string",
      "phoneNumber": "string",
      "photoPath": "string",
      "socialMedias": [
        {
          "id": "string",
          "socialMediaType": "string",
          "socialMediaReference": "string"
        }
      ],
      "accountNumber": "string",
      "swiftBic": "string",
      "ownerId": "string",
      "linkedUserId": "string"
    },
    "type": "base",
    "status": "none",
    "business": {
      "companyName": "string",
      "industry": "string",
      "size": "string",
      "sector": "string",
      "stage": "string"
    }
  }
}
```

```
    "legal": "individual",
    "type": "eshop",
    "vat": "string",
    "administrator": {
        "firstName": "string",
        "lastName": "string",
        "email": "string",
        "phone": "string"
    },
},
"address": {
    "country": "AD",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "houseNumber": "string"
},
"additional": {

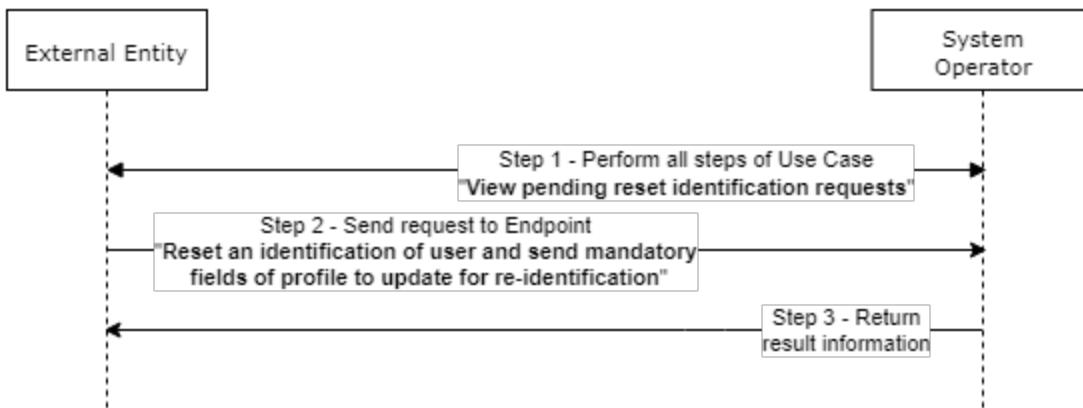
},
"security": {
    "twoFactorsAuthEnabled": false,
    "transactionNotification": {
        "phone": false,
        "email": false
    },
    "authorizationNotification": {
        "phone": false,
        "email": false
    }
}
},
```

```
"status": "ok",
"message": "string"
}
```

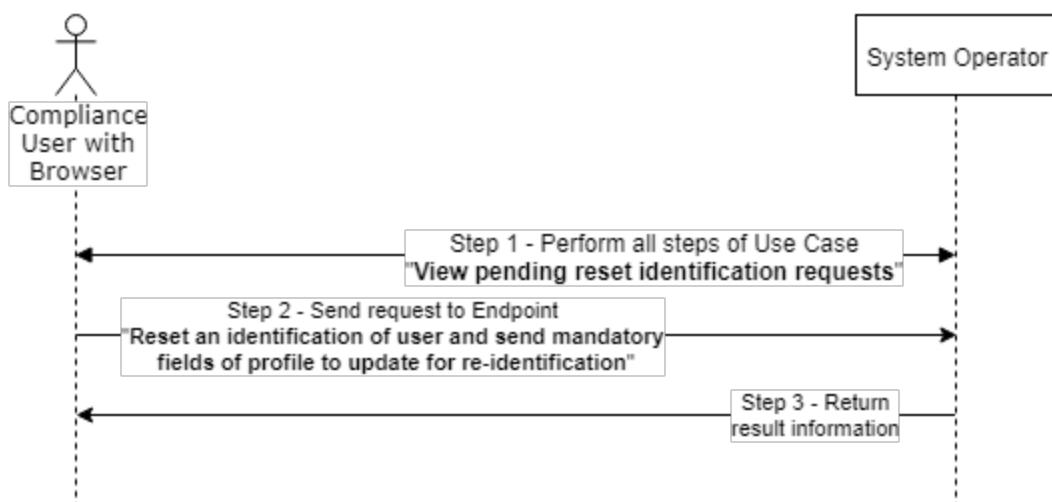
Reset an identification of user and send mandatory fields of profile to update for reidentification scheme

Use case: Reset an identification of user and send mandatory fields of profile to update for re-identification

Basic FFlow



Optional Web UI Flow



Reset an identification of user description

Use Case Name

Reset an identification of the user

Brief Description

A User or External Entity on behalf of a User with role permission RESET_IDENTIFICATION_INITIATOR will go through all steps of “View pending reset identification requests” Use Case, and then send a request to Endpoint “Reset an identification of user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: RESET_IDENTIFICATION_INITIATOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View pending reset identification requests”.
2. External Entity sends a request to Endpoint “Reset an identification of user”.

Endpoint URL: POST /profiles/{userId}/reset-identification-requests/{requestId}/approve

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View pending reset identification requests”.
2. A user sends a request to Endpoint “Reset an identification of user”.

Endpoint URL: POST /profiles/{userId}/reset-identification-requests/{requestId}/approve

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

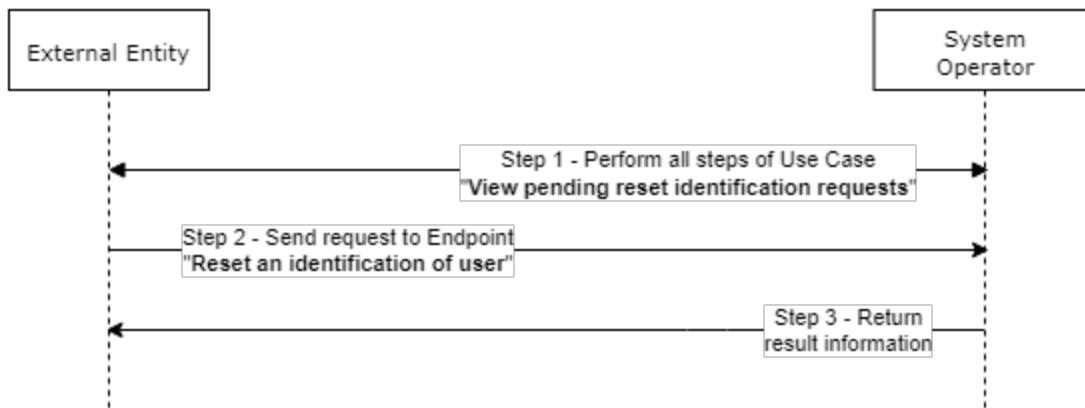
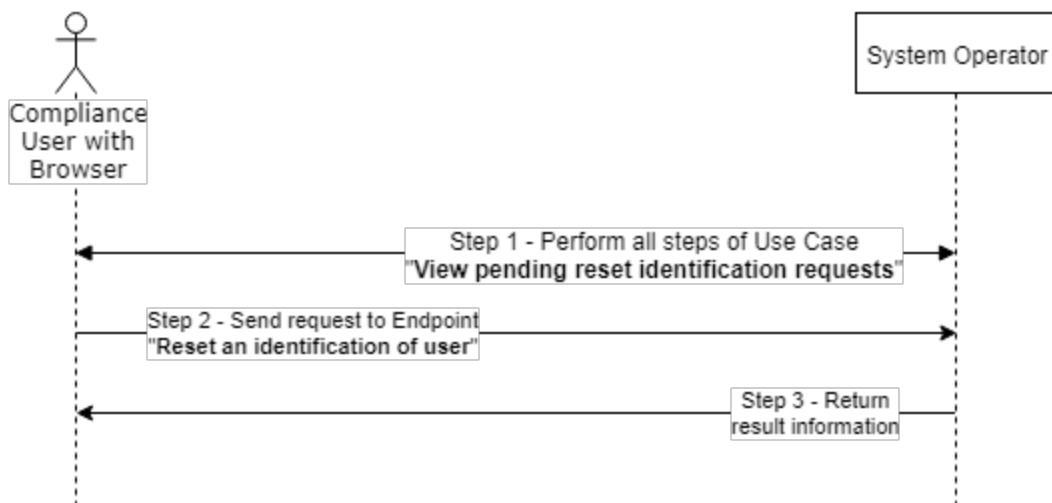
Post Conditions

Reset is done.

Result example

```
{  
    "request": {  
        "id": "string",  
        "description": "string",  
        "organizationId": "string",  
        "status": "APPROVED",  
        "initiatorUserId": "string",  
        "initiatorRole": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Reset an identification of user scheme

Use case: Reset an identification of user**Basic FFlow****Optional Web UI Flow**

View pending reset identification requests description

Use Case Name

View pending reset identification requests

Brief Description

A User or External Entity on behalf of a User with role permissions RESET_IDENTIFICATION_INITIATOR will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View pending reset identification request”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: RESET_IDENTIFICATION_INITIATOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View pending reset identification request”.

Endpoint URL: POST /profiles/view-reset-identification-requests

Parameter:

```
{
  "filter": {
    "status": "APPROVED",
    "organizationId": "string"
  },
  "sort": {
    "organizationId": "asc",
    "status": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns Requested list to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View pending reset identification request”.

Endpoint URL: POST /profiles/view-reset-identification-requests

Parameter:

```
{
  "filter": {
    "status": "APPROVED",
    "organizationId": "string"
  },
  "sort": {
    "organizationId": "asc",
    "status": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns Requested list to User. (See Result example below)

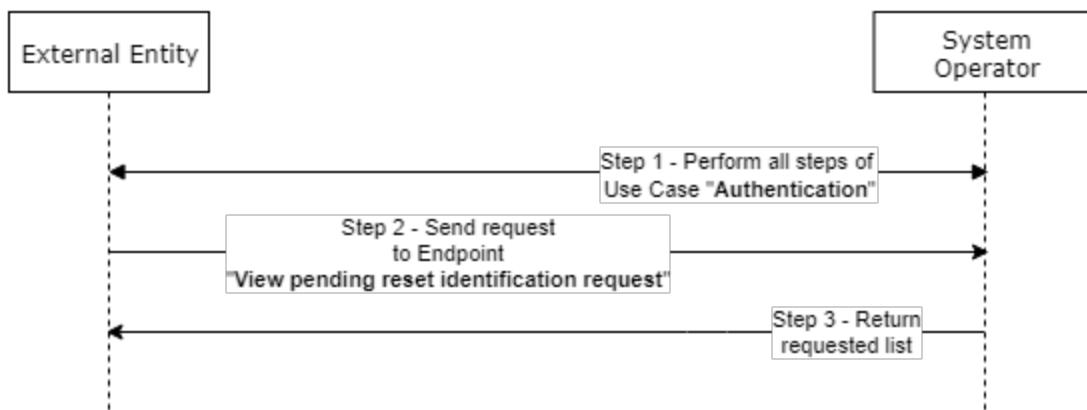
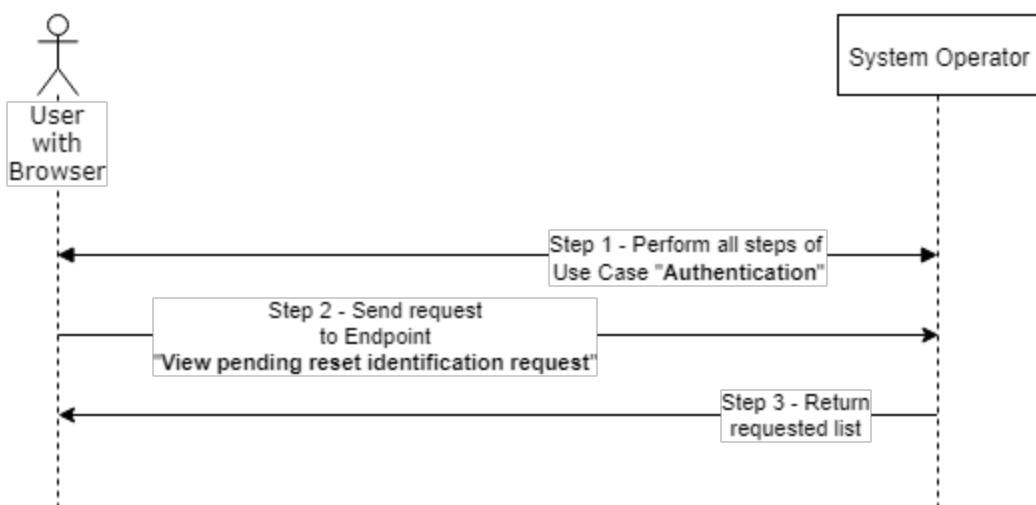
Post Conditions

List of requests is available

Result example

```
{
  "status": "ok",
  "message": "processed successfully",
  "pageNumber": 0,
  "pageSize": 20,
  "totalRecords": 1,
  "totalPages": 1,
  "records": [
    {
      "id": "b7b57423-f62a-4bfe-a983-c488065089bf",
      "description": "Profile reset identification",
      "organizationId": "e4652636-e4c1-400e-abfd-7ef7bcbfc494",
      "status": "PENDING",
      "initiatorUserId": "b70e0ef1-45f9-4c3d-8726-b59d67f0f5ff",
      "initiatorRole": "individual"
    }
  ]
}
```

[View pending reset identification requests scheme](#)

Use case: View pending reset identification request**Basic FFlow****Optional Web UI Flow**

Profile - information

Confirm contact verification request using one time password description

Use Case Name

Confirm contact verification request using the one-time password

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, check SMS on the phone or Email to get OTP, and then send a request to Endpoint “Confirm contact verification request using the one-time password”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. Email or SMS service provider.
3. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity obtains OTP from the appropriate communication service provider.
3. External Entity sends a request to Endpoint “Confirm contact verification request using the one-time password”.

Endpoint URL: POST /profiles/my/contact/confirm

Parameters:

```
{
  "login": "string",
  "otp": "string"
}
```

4. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user obtains OTP from appropriate communication service provider.
3. A user sends a request to Endpoint “Confirm contact verification request using the one-time password”.

Endpoint URL: POST /profiles/my/contact/confirm

Parameters:

```
{
  "login": "string",
  "otp": "string"
}
```

4. System Operator returns result information to User (See Result example below).

Post Conditions

Contact is verified.

Result example

```
{
  "profile": {
    "person": {
      "namePlain": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "nameIntl": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "description": "string"
    },
    "contact": {
      "id": "string",
      "nickname": "string",
      "firstName": "string",
      "lastName": "string",
      "address": "string",
      "city": "string",
      "country": "string",
      "postCode": "string",
      "email": "string",
      "phoneNumber": "string",
      "photoPath": "string",
      "socialMedias": [
        {
          "id": "string",
          "socialMediaType": "string",
          "socialMediaReference": "string"
        }
      ],
      "accountNumber": "string",
      "password": "string"
    }
  }
}
```

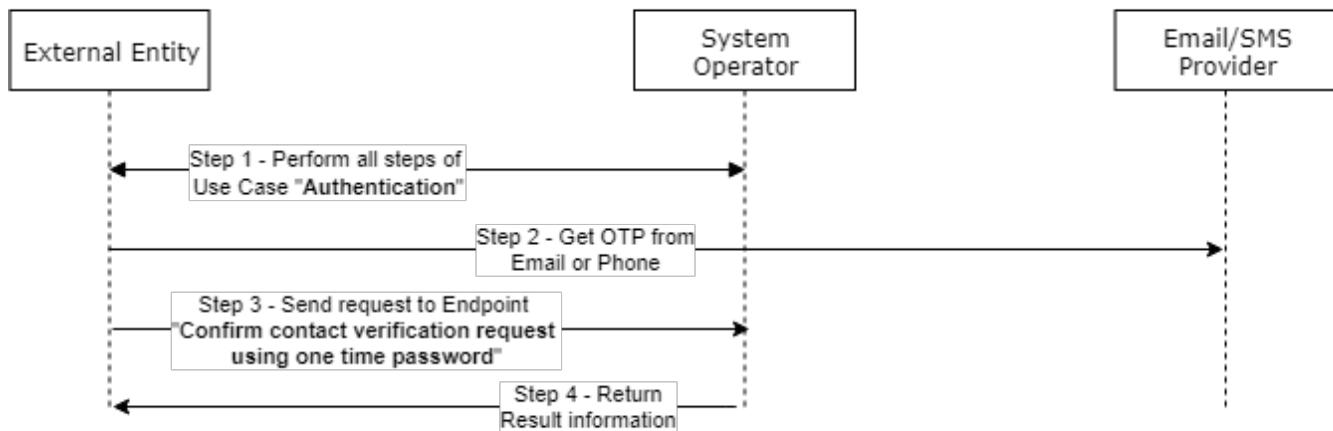
```
        "swiftBic": "string",
        "ownerId": "string",
        "linkedUserId": "string"
    },
    "type": "base",
    "status": "none",
    "business": {
        "companyName": "string",
        "legal": "individual",
        "type": "eshop",
        "vat": "string",
        "administrator": {
            "firstName": "string",
            "lastName": "string",
            "email": "string",
            "phone": "string"
        }
    },
    "address": {
        "country": "AD",
        "zipCode": "string",
        "city": "string",
        "street": "string",
        "houseNumber": "string"
    },
    "additional": {
    },
    "security": {
        "twoFactorsAuthEnabled": false,
        "transactionNotification": {
            "phone": false,
            "email": false
        },
        "authorizationNotification": {
            "phone": false,
            "email": false
        }
    }
},
```

```
"status": "ok",
"message": "string"
}
```

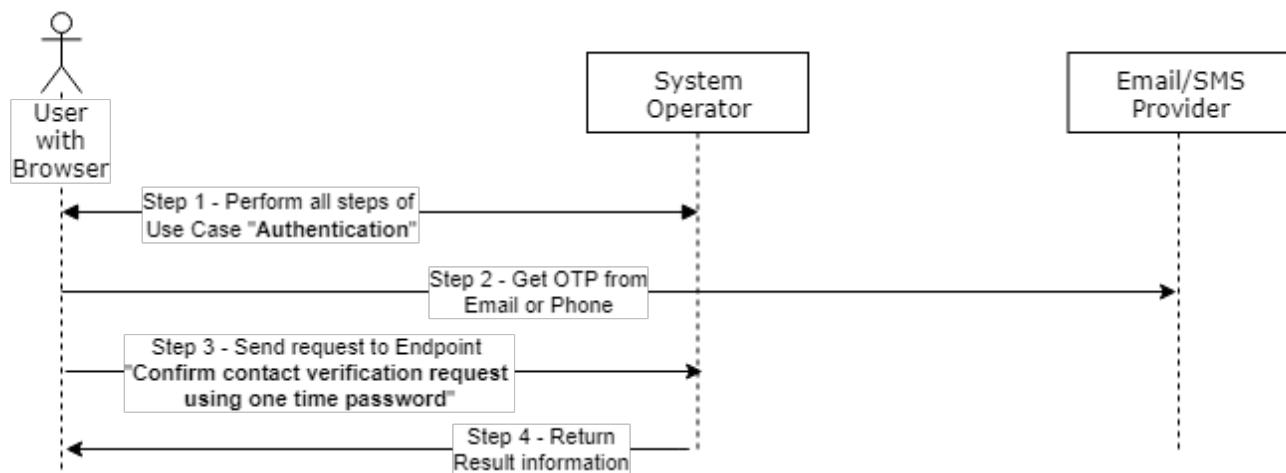
Confirm contact verification request using one time password scheme

Use case: Confirm contact verification request using one time password

Basic FLow



Optional Web UI Flow



Create a reset identification request description

Use Case Name

Create a reset identification request

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Create a reset identification request”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Create a reset identification request”.

Endpoint URL: POST /profiles/my/reset-identification

Parameters:

```
{
  "description": "string"
}
```

“description”: “string” - (optional)Description

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a reset identification request”.

Endpoint URL: POST /profiles/my/reset-identification

Parameters:

```
{
  "description": "string"
}
```

“description”: “string” - (optional)Description

3. System Operator returns xx result information to User (See Result example below).

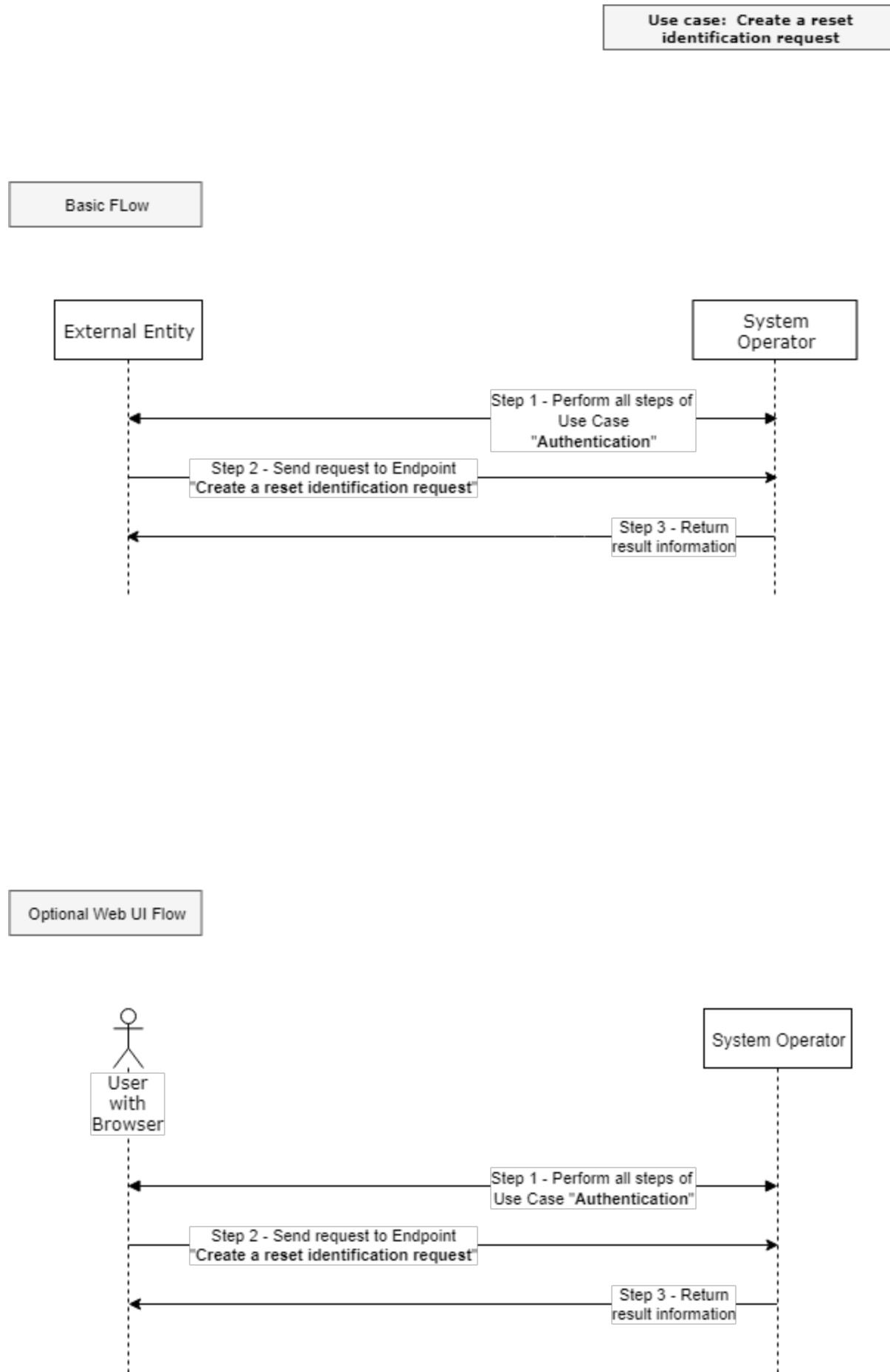
Post Conditions

Request sent.

Result example

```
{  
    "request": {  
        "id": "string",  
        "description": "string",  
        "organizationId": "string",  
        "status": "APPROVED",  
        "initiatorUserId": "string",  
        "initiatorRole": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Create a reset identification request scheme



Create contact verification request description

Use Case Name

Create contact verification request

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Create contact verification request”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. Email or SMS service provider.
3. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Create contact verification request”.

Endpoint URL: POST /profiles/my/contact

Parameters:

```
{
  "login": "string"
}
```

3. System Operator creates, saves, and sends OTP to email or Phone number as SMS
4. System Operator returns OTP location to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create contact verification request”.

Endpoint URL: POST /profiles/my/contact

Parameters:

```
{
  "login": "string"
}
```

3. System Operator creates, saves, and sends OTP to email or Phone number as SMS
4. System Operator returns OTP location to User (See Result example below).

Post Conditions

OTP is available.

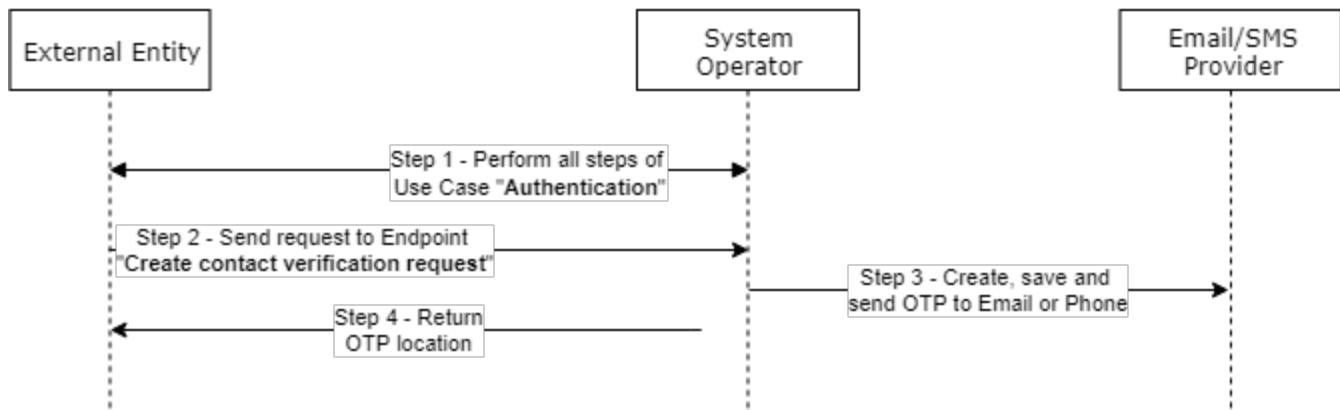
Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "action": "EMAIL_SENT"  
}
```

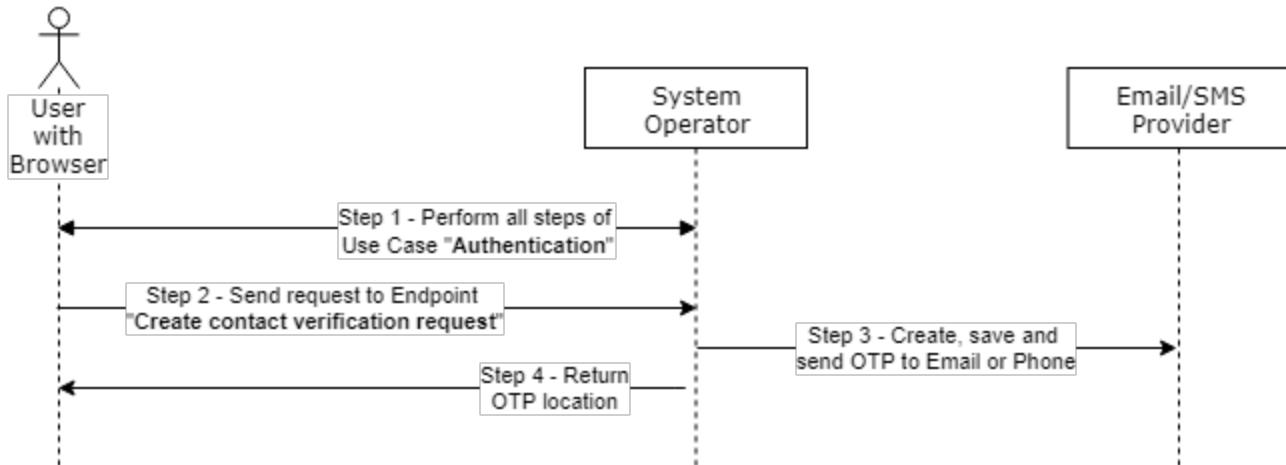
Create contact verification request scheme

Use case: Create contact verification request

Basic FFlow



Optional Web UI Flow



Get user profile description

Use Case Name

Get user profile

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get user profile”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to Endpoint “Get user profile”.

Endpoint URL: GET /profiles/my

Parameters: Security TOKEN

3. System Operator returns User profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “Get user profile”.

Endpoint URL: GET /profiles/my

Parameters: Security TOKEN

3. System Operator returns User profile information to User (See Result example below).

Post Conditions

Profile information is available.

Result example

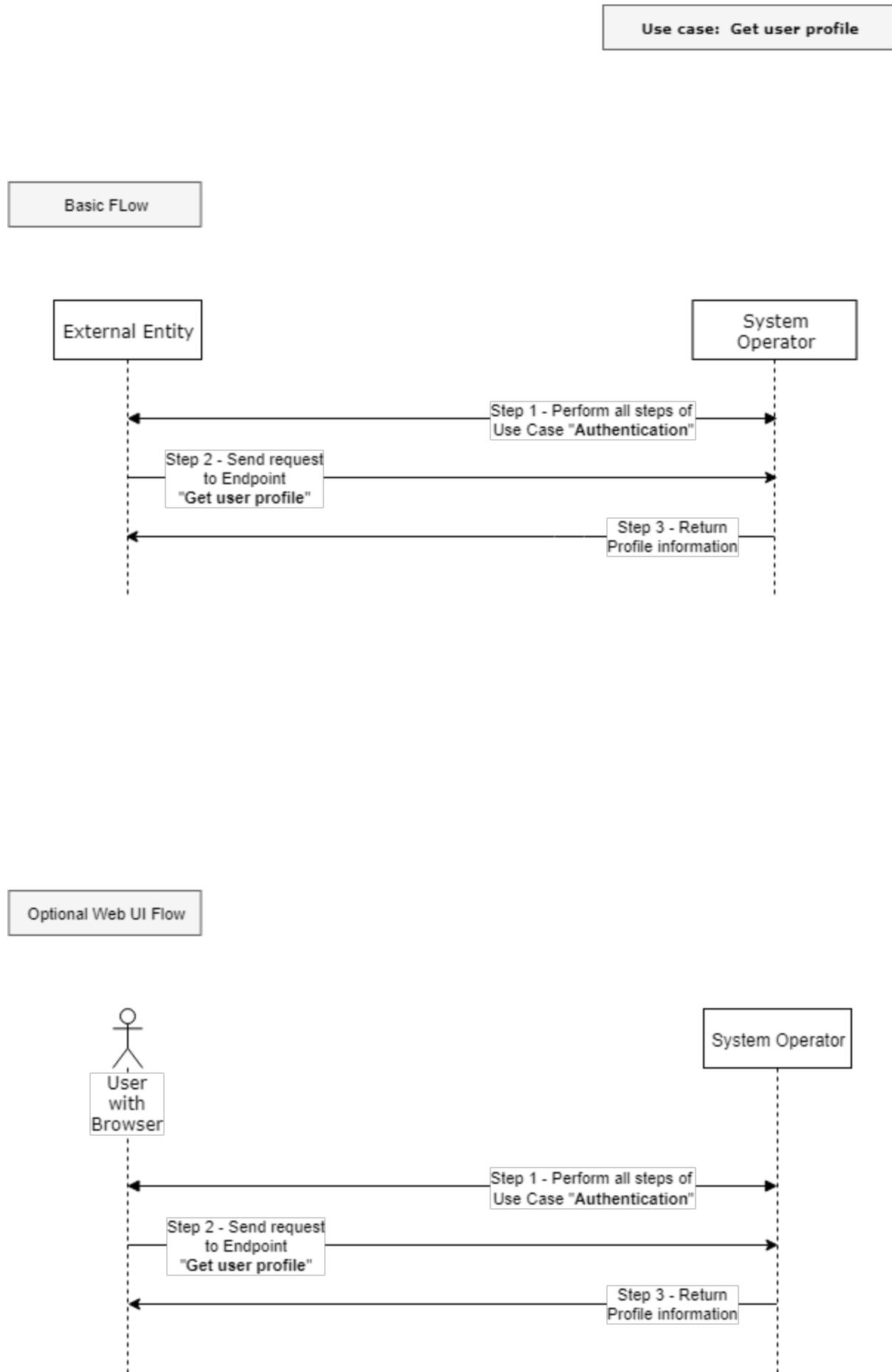
```
{
  "profile": {
    "person": {
      "namePlain": {
        "first": "string",
        "last": "string",
        "middle": "string"
      },
      "nameIntl": {
        "first": "string",
        "last": "string",
        "middle": "string"
      }
    }
  }
}
```

```
        },
        "description": "string"
    },
    "contact": {
        "id": "string",
        "nickname": "string",
        "firstName": "string",
        "lastName": "string",
        "address": "string",
        "city": "string",
        "country": "string",
        "postCode": "string",
        "email": "string",
        "phoneNumber": "string",
        "photoPath": "string",
        "socialMedias": [
            {
                "id": "string",
                "socialMediaType": "string",
                "socialMediaReference": "string"
            }
        ],
        "accountNumber": "string",
        "swiftBic": "string",
        "ownerId": "string",
        "linkedUserId": "string"
    },
    "type": "base",
    "status": "none",
    "business": {
        "companyName": "string",
        "legal": "individual",
        "type": "eshop",
        "vat": "string",
        "administrator": {
            "firstName": "string",
            "lastName": "string",
            "email": "string",
            "phone": "string"
        }
    },
    "address": {
        "country": "AD",
        "zipCode": "string",
        "city": "string",
        "street": "string",
        "houseNumber": "string"
    },
    "additional": {},
    "security": {
```

```
        "twoFactorsAuthEnabled": false,
        "transactionNotification": {
            "phone": false,
            "email": false
        },
        "authorizationNotification": {
            "phone": false,
            "email": false
        }
    },
}
```

```
"status": "ok",
"message": "string"
}
```

Get user profile scheme



Multiple profile updates description

Use Case Name

Multiple profile updates

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Get user profile” Use Case, and then send multiple requests to Endpoints “Update additional information”, “Update user address”, “Update business information”, “Update user password”, “Update person information”, “Update person information”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get user profile”.
2. External Entity sends a request to Endpoint “Update additional information”.

Endpoint URL: PATCH /profiles/my/additional

Parameters:

```
{
  "additional": {
    }
}
```

3. External Entity sends a request to Endpoint “Update user address”.

Endpoint URL: PATCH /profiles/my/address

Parameters:

```
{
  "address": {
    "country": "AD",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "houseNumber": "string"
  }
}
```

4. External Entity sends a request to Endpoint “Update business information”.

Endpoint URL: PATCH /profiles/my/business

Parameters:

```
{
    "companyName": "string",
    "type": "eshop",
    "vat": "string",
    "administrator": {
        "firstName": "string",
        "lastName": "string",
        "email": "string",
        "phone": "string"
    }
}
```

5. External Entity sends a request to Endpoint “Update user password”.

Endpoint URL: PATCH /profiles/my/password

Parameters:

```
{
    "currentUserPassword": "string",
    "newUserPassword": "string"
}
```

6. External Entity sends a request to Endpoint “Update person information”.

Endpoint URL: PATCH /profiles/my/person

Parameters:

```
{
    "person": {
        "namePlain": {
            "first": "string",
            "last": "string",
            "middle": "string"
        },
        "nameIntl": {
            "first": "string",
            "last": "string",
            "middle": "string"
        },
        "description": "string"
    }
}
```

7. External Entity sends a request to Endpoint “Update security settings”.

Endpoint URL: PATCH /profiles/my/security-settings

Parameters:

```
{
  "security": {
    "twoFactorsAuthEnabled": false,
    "secretWord": "string",
    "transactionNotification": {
      "phone": false,
      "email": false
    },
    "authorizationNotification": {
      "phone": false,
      "email": false
    }
  }
}
```

8. System Operator returns User profile information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get user profile”.
2. A user sends a request to Endpoint “Update additional information”.

Endpoint URL: PATCH /profiles/my/additional

Parameters:

```
{
  "additional": {
    ...
  }
}
```

3. A user sends a request to Endpoint “Update user address”.

Endpoint URL: PATCH /profiles/my/address

Parameters:

```
{
  "address": {
    "country": "AD",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "houseNumber": "string"
  }
}
```

4. A user sends a request to Endpoint “Update business information”.

Endpoint URL: PATCH /profiles/my/business

Parameters:

```
{
  "companyName": "string",
  "type": "eshop",
  "vat": "string",
  "administrator": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string"
  }
}
```

5. A user sends a request to Endpoint “Update user password”.

Endpoint URL: PATCH /profiles/my/password

Parameters:

```
{
  "currentUserPassword": "string",
  "newUserPassword": "string"
}
```

6. A user sends a request to Endpoint “Update person information”.

Endpoint URL: PATCH /profiles/my/person

Parameters:

```
{
  "person": {
    "namePlain": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "nameIntl": {
      "first": "string",
      "last": "string",
      "middle": "string"
    },
    "description": "string"
  }
}
```

7. A user sends a request to Endpoint “Update security settings”.

Endpoint URL: PATCH /profiles/my/security-settings

Parameters:

```
{
  "security": {
    "twoFactorsAuthEnabled": false,
    "secretWord": "string",
    "transactionNotification": {
      "phone": false,
      "email": false
    },
    "authorizationNotification": {
      "phone": false,
      "email": false
    }
  }
}
```

8. System Operator returns User profile information to User (See Result example below).

Post Conditions

Profile information is updated.

Result example

```
{
  "profile": {
    "person": {
```

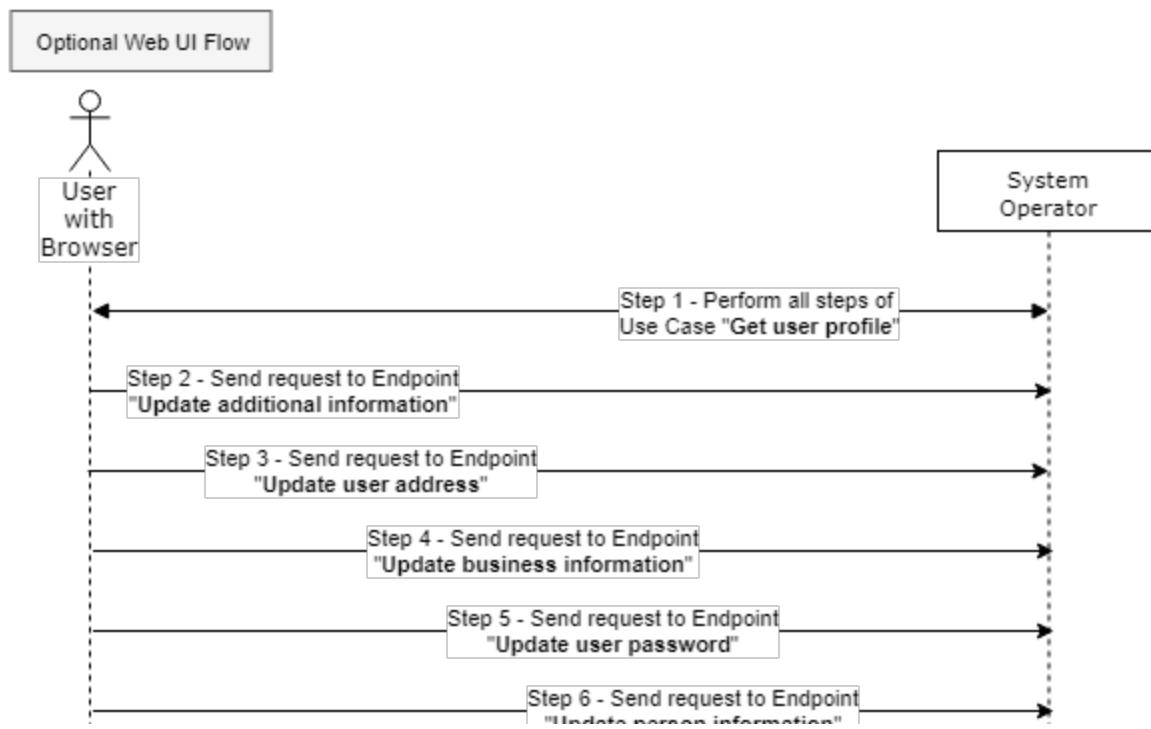
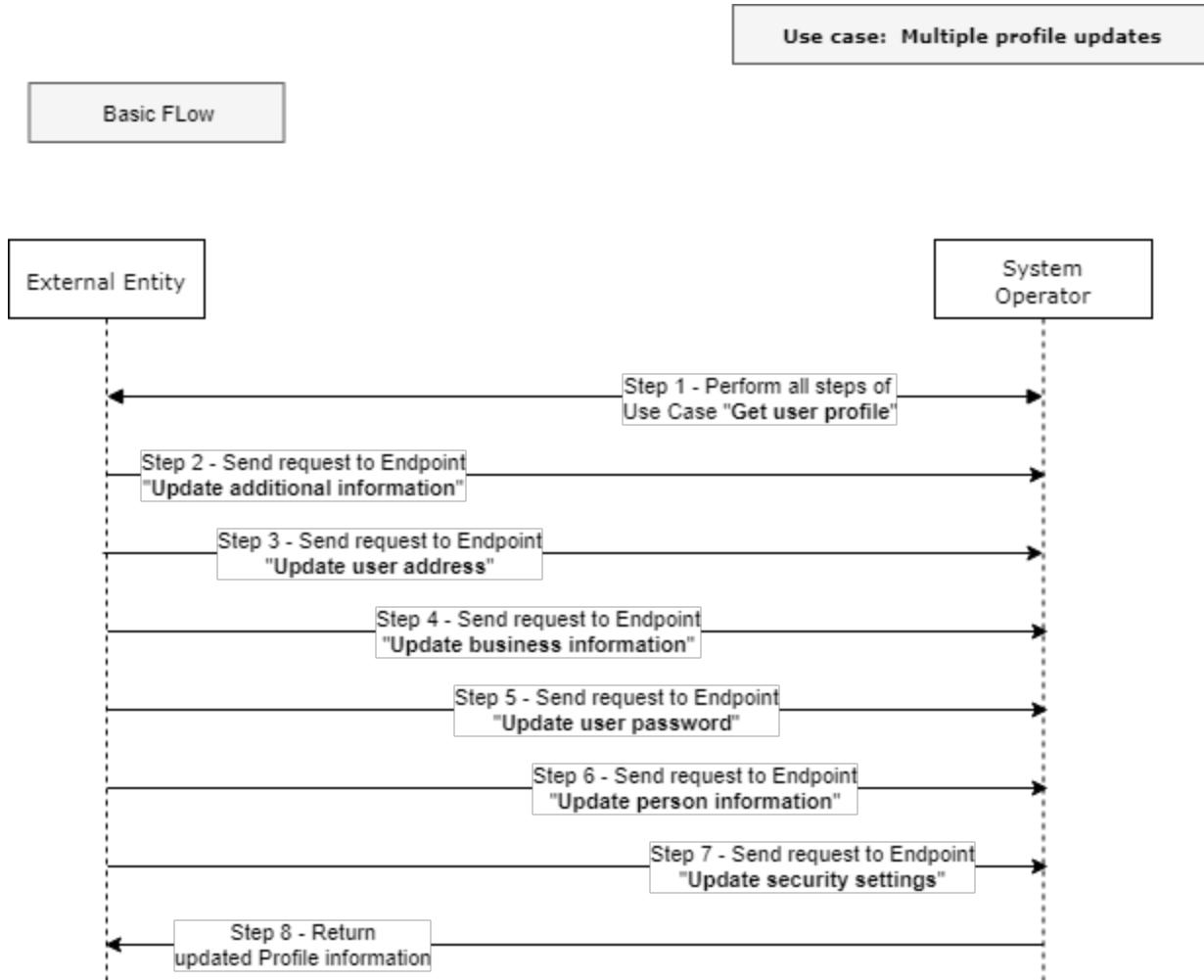
```
"namePlain":{  
    "first":"string",  
    "last":"string",  
    "middle":"string"  
},  
"nameIntl":{  
    "first":"string",  
    "last":"string",  
    "middle":"string"  
},  
"description":"string"  
,  
"contact":{  
    "id":"string",  
    "nickname":"string",  
    "firstName":"string",  
    "lastName":"string",  
    "address":"string",  
    "city":"string",  
    "country":"string",  
    "postCode":"string",  
    "email":"string",  
    "phoneNumber":"string",  
    "photoPath":"string",  
    "socialMedias": [  
        {  
            "id":"string",  
            "socialMediaType":"string",  
            "socialMediaReference":"string"  
        }  
    ],  
    "accountNumber":"string",  
    "swiftBic":"string",  
    "ownerId":"string",  
    "linkedUserId":"string"  
},  
"type":"base",  
"status":"none",  
"business":{  
    "companyName":"string",  
    "legal":"individual",  
    "type":"eshop",  
    "vat":"string",  
    "administrator":{  
        "firstName":"string",  
        "lastName":"string",  
        "email":"string",  
        "phone":"string"  
    }  
},  
},
```

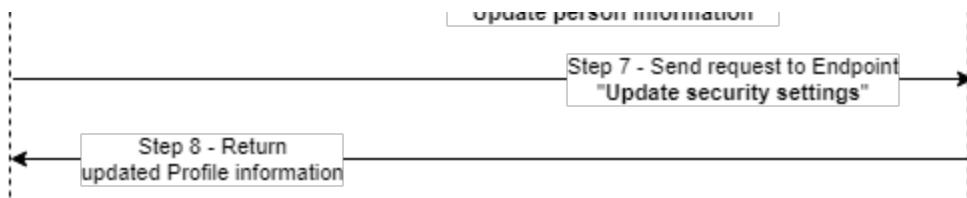
```
"address": {
    "country": "AD",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "houseNumber": "string"
},
"additional": {

},
"security": {
    "twoFactorsAuthEnabled": false,
    "transactionNotification": {
        "phone": false,
        "email": false
    },
    "authorizationNotification": {
        "phone": false,
        "email": false
    }
}
},
```

```
"status": "ok",
"message": "string"
}
```

Multiple profile updates scheme





Resend one time password to confirm contact description

Use Case Name

Resend one-time password to confirm contact

Brief Description

A User or External Entity on behalf of a User with role permission PROFILE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Resend one-time password to confirm contact”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PROFILE_OWNER.
2. Email or SMS service provider.
3. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Resend one-time password to confirm contact”.

Endpoint URL: POST /profiles/my/contact

Parameters:

```
{
  "login": "string"
}
```

3. System Operator creates, saves, and re-sends OTP to email or Phone number as SMS
4. System Operator returns OTP location to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Resend one-time password to confirm contact”.

Endpoint URL: POST /profiles/my/contact

Parameters:

```
{  
    "login": "string"  
}
```

3. System Operator creates, saves, and re-sends OTP to email or Phone number as SMS

4. System Operator returns OTP location to User (See Result example below).

Post Conditions

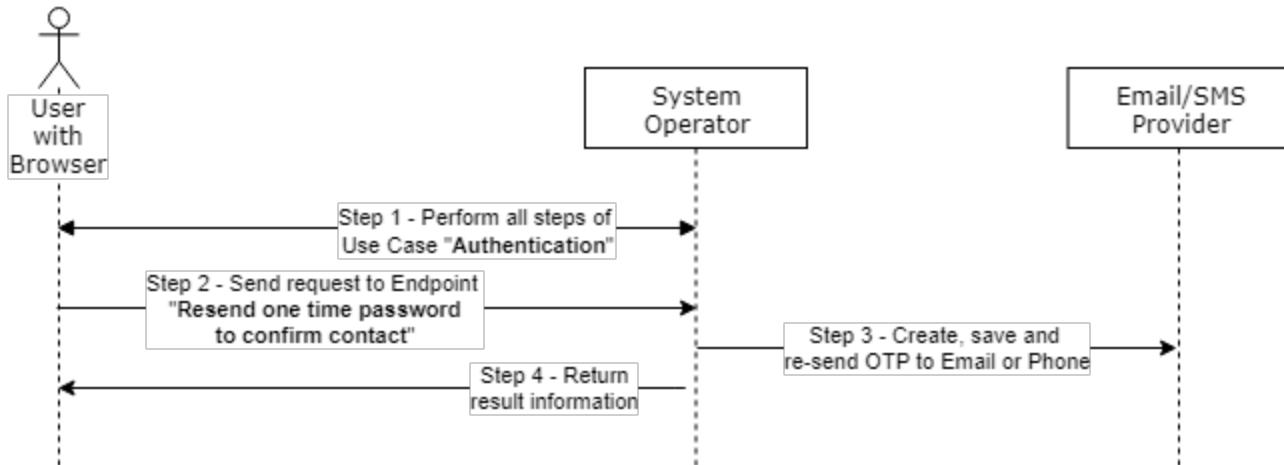
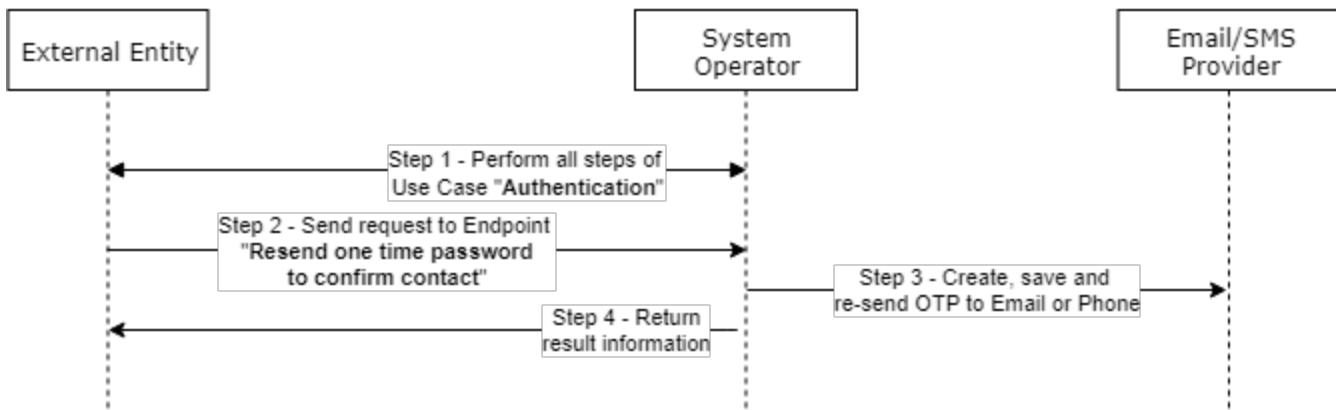
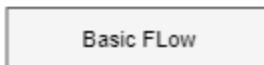
OTP is available.

Result example

```
{  
    "status": "ok",  
    "message": "string"  
}
```

Resend one time password to confirm contact scheme

Use case: Resend one time password to confirm contact



Profile - transactions confirmation

Confirm update of transaction confirmation settings record description

Use Case Name

Confirm update of transaction confirmation settings record

Brief Description

A User or External Entity on behalf of a User with role permission TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER will go through all steps of “Request update of transaction confirmation settings record” Use Case, and then send a request to Endpoint “Confirm update of transaction confirmation settings record”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.
3. Communication channel provider - mobile phone operator or email server.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Request update of transaction confirmation settings record”.
2. The external entity obtains an OTP or PIN via SMS or Email.
3. External Entity sends a request to Endpoint “Confirm update of transaction confirmation settings record”.

Endpoint URL: POST /profiles/my/transactions-confirmation-settings/{recordId}/confirm-update

Parameters:

```
{
  "value": "string"
}
```

“value”: “string” - Confirmation value (OTP, PIN etc).

4. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Request update of transaction confirmation settings record”.
2. A user obtains an OTP or PIN via SMS or Email.
3. A user sends a request to Endpoint “Confirm update of transaction confirmation settings record”.

Endpoint URL: POST /profiles/my/transactions-confirmation-settings/{recordId}/confirm-update

Parameters:

```
{
  "value": "string"
}
```

"value": "string" - Confirmation value (OTP, PIN etc).

4. System Operator returns result information to User (See Result example below).

Post Conditions

The record is confirmed.

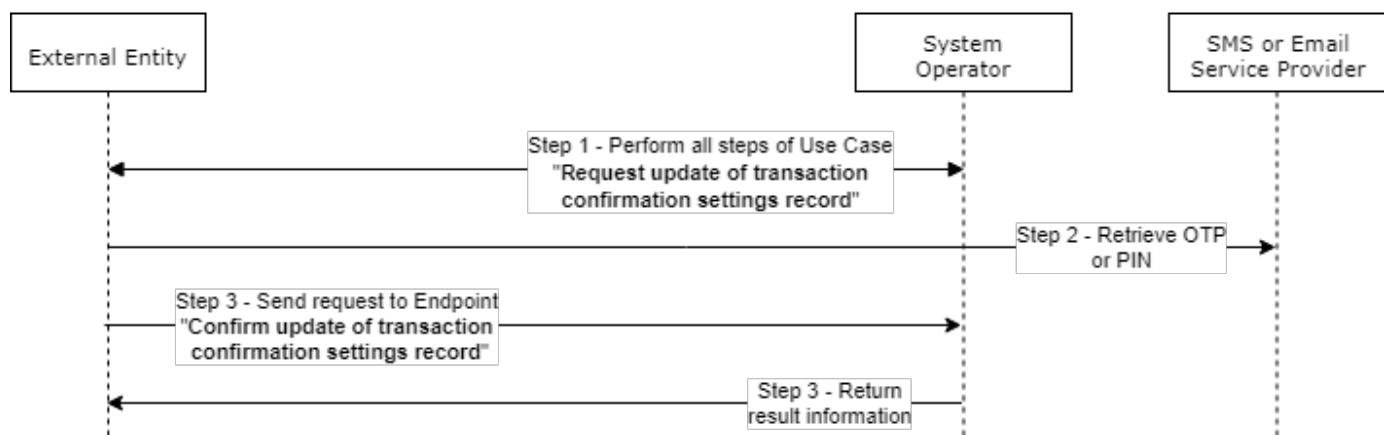
Result example

```
{  
  "settings": {  
    "recordId": "string",  
    "processType": "string",  
    "confirmationType": "string",  
    "active": false  
  },  
  "status": "ok",  
  "message": "string"  
}
```

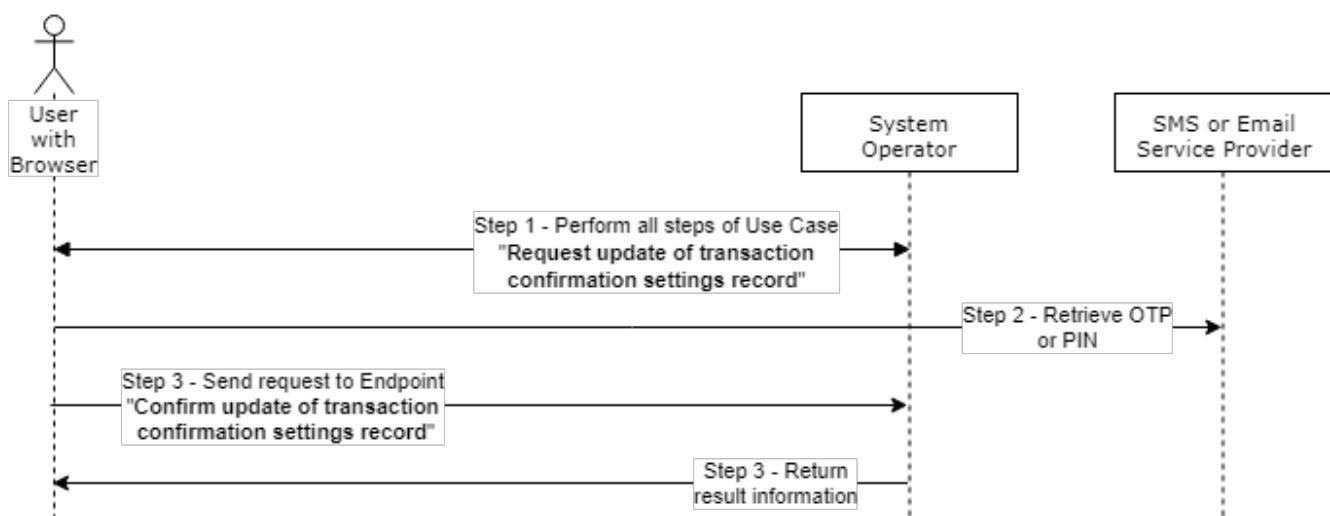
Confirm update of transaction confirmation settings record scheme

Use case: Confirm update of transaction confirmation settings record

Basic FLow



Optional Web UI Flow



Create transaction confirmation settings record description

Use Case Name

Create transaction confirmation settings record

Brief Description

A User or External Entity on behalf of a User with role permission TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER will go through all steps of “A authentication” Use Case, and then send a request to the following Endpoints “Get current transactions confirmation settings”, “Retrieve list of business process types eligible for confirmations set up”, “Create transaction confirmation settings record”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get current transactions confirmation settings”.
2. Perform all steps of Use Case “Retrieve the list of business process types eligible for confirmations set up”.
3. External Entity sends a request to Endpoint “Create transaction confirmation settings record”.

Endpoint URL: POST /profiles/my/transactions-confirmation-settings

Parameters:

```
{
  "processType": "string",
  "confirmationType": "string",
  "active": false
}
```

4. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get current transactions confirmation settings”.
2. Perform all steps of Use Case “Retrieve the list of business process types eligible for confirmations set up”.
3. A user sends a request to Endpoint “Create transaction confirmation settings record”.

Endpoint URL: POST /profiles/my/transactions-confirmation-settings

Parameters:

```
{
  "processType": "string",
  "confirmationType": "string",
  "active": false
}
```

4. System Operator returns xx result information to User (See Result example below).

Post Conditions

Transaction confirmation settings record is available.

Result example

```
{  
    "settings": {  
        "id": "string",  
        "processType": "string",  
        "confirmationType": "string",  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

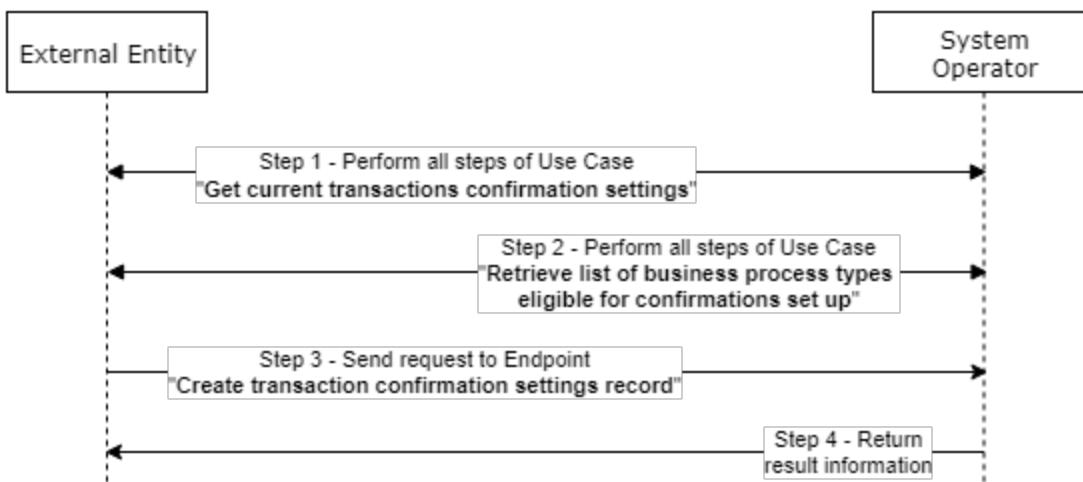
"settings": - Confirmation settings

"id": "string", - Record identifier

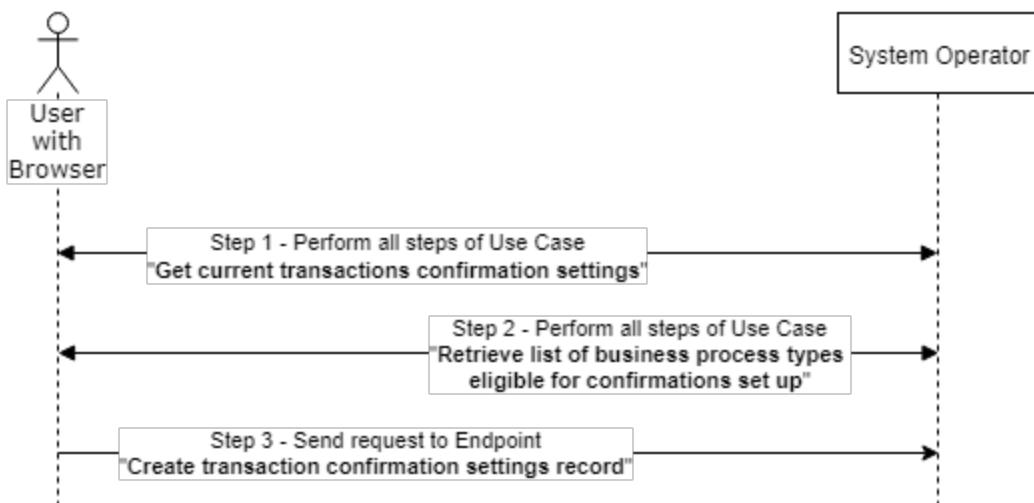
Create transaction confirmation settings record scheme

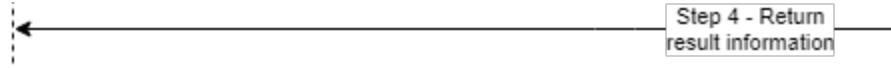
Use case: Create transaction confirmation settings record

Basic FFlow



Optional Web UI Flow





Get current transactions confirmation settings description

Use Case Name

Get current transactions confirmation settings

Brief Description

A User or External Entity on behalf of a User with role permission TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER will go through all steps of “A authentication” Use Case, and then send a request to Endpoint “Get current transactions confirmation settings”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get current transactions confirmation settings”.

Endpoint URL: GET /profiles/my/transactions-confirmation-settings

Parameters: Security TOKEN

3. System Operator returns current transactions confirmation settings to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get current transactions confirmation settings”.

Endpoint URL: GET /profiles/my/transactions-confirmation-settings

Parameters: Security TOKEN

3. System Operator returns current transactions confirmation settings to User (See Result example below).

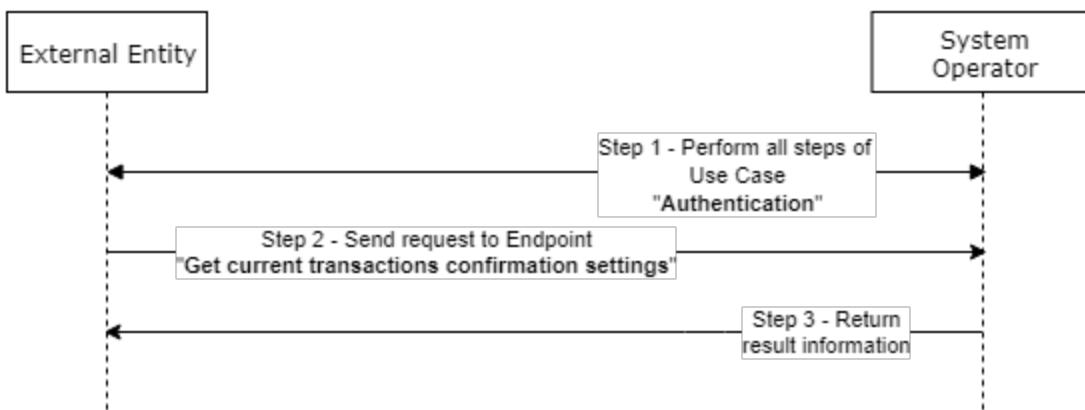
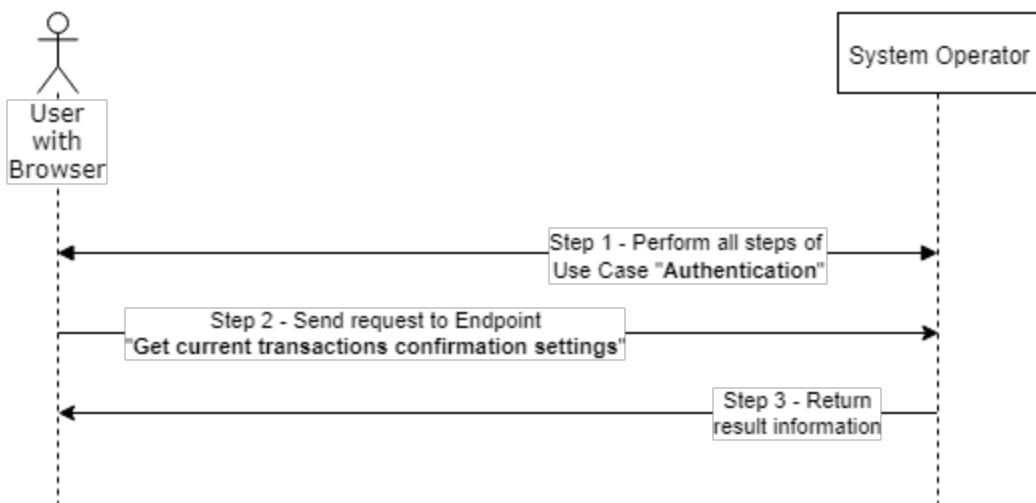
Post Conditions

List of current transactions confirmation settings is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "processType": "string",  
            "confirmationType": "string",  
            "active": false  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Get current transactions confirmation settings scheme

Use case: Get current transactions confirmation settings**Basic FFlow****Optional Web UI Flow**

Request update of transaction confirmation settings record description

Use Case Name

Request update of transaction confirmation settings record

Brief Description

A User or External Entity on behalf of a User with role permission TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER will go through all steps of “A authentication” Use Case, and then send a request to Endpoint “Request update of transaction confirmation settings record”.

As a result of this request, the System Operator sends OTP/PIN to the mobile phone as SMS or via email.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get current transactions confirmation settings”.
2. External Entity sends a request to Endpoint “Request update of transaction confirmation settings record”.

Endpoint URL: POST

/profiles/my/transactions-confirmation-settings/{id}/request-update

Parameters: {id} Identifier of the confirmation settings record

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get current transactions confirmation settings”.
2. A user sends a request to Endpoint “Request update of transaction confirmation settings record”.

Endpoint URL: POST

/profiles/my/transactions-confirmation-settings/{id}/request-update

Parameters: {id} Identifier of the confirmation settings record

3. System Operator returns result information to User (See Result example below).

Post Conditions

The request was created.

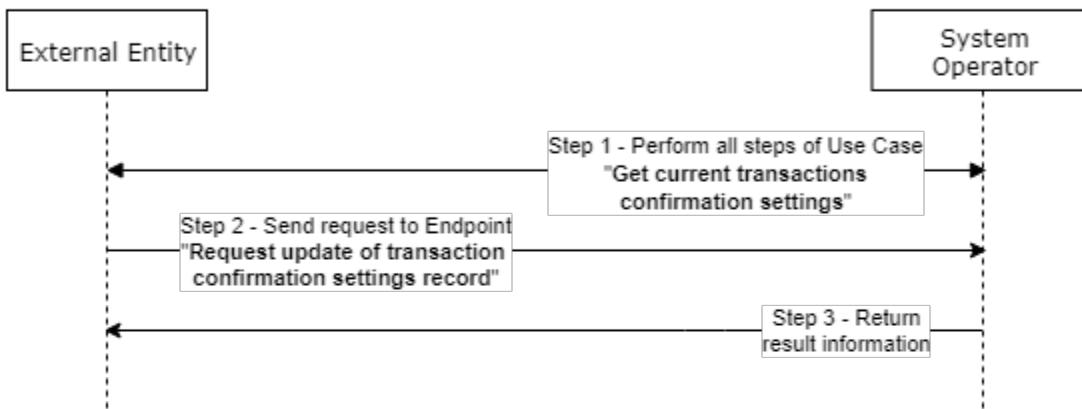
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

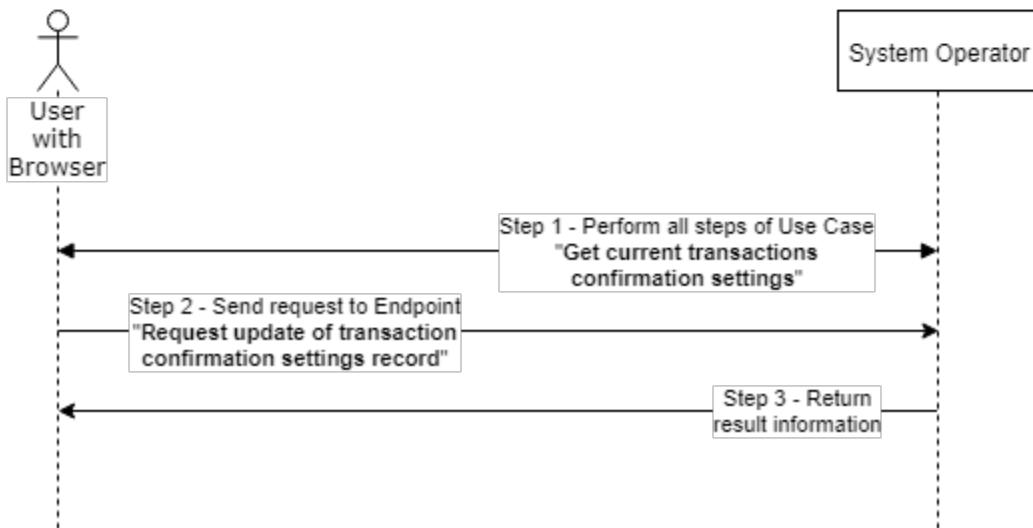
Request update of transaction confirmation settings record scheme

Use case: Request update of transaction confirmation settings record

Basic FLow



Optional Web UI Flow



Retrieve list of business process types eligible for confirmations set up description

Use Case Name

Retrieve the list of business process types eligible for confirmations set up

Brief Description

A User or External Entity on behalf of a User with role permission TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER will go through all steps of “A authentication” Use Case, and then send a request to Endpoint “Retrieve the list of business process types eligible for confirmations set up”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TRANSACTIONS_CONFIRMATION_SETTINGS_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Retrieve the list of business process types eligible for confirmations set up”.

Endpoint URL:

POST /profiles/my/transactions-confirmation-settings/view-eligible-business-process-types

Parameters: Security TOKEN

3. System Operator returns requested list to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Retrieve the list of business process types eligible for confirmations set up”.

Endpoint URL:

POST /profiles/my/transactions-confirmation-settings/view-eligible-business-process-types

Parameters: Security TOKEN

3. System Operator returns returns requested list to User (See Result example below).

Post Conditions

Requested list is available.

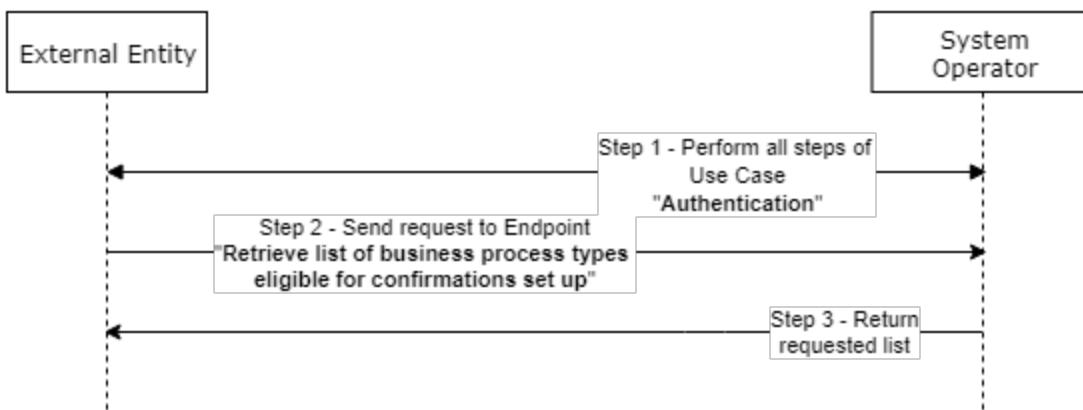
Result example

```
{
  "records": [
    "string"
  ],
  "status": "ok",
  "message": "string"
}
```

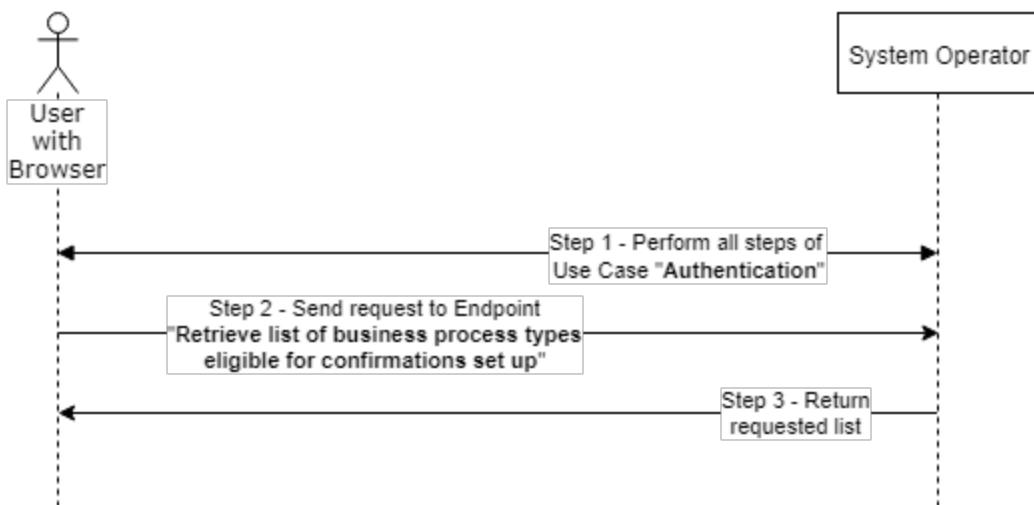
Retrieve list of business process types eligible for confirmations set up scheme

Use case: Retrieve list of business process types eligible for confirmations set up

Basic FFlow



Optional Web UI Flow



Top up via bank

Create a request to top up via bank description

Use Case Name

Create a request to top up via bank

Brief Description

A User or External Entity on behalf of a User with role permission “BANK_TOP_UP_CREATION_EXECUTOR” will go through all steps of “Authentication” Use Case (to obtain token), and then send a request to Endpoint “Create a request to top up via bank”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: “BANK_TOP_UP_CREATION_EXECUTOR”.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Create a request to top up via bank”.

Endpoint URL: POST /bank-top-ups/create-request

Parameters: TOKEN - identifies authenticated user

```
{
  "coin": "string",
  "amount": 1000,
  "bankDetails": {
    "fullName": "string"
  }
}
```

3. System Operator returns request to top up via bank information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a request to top up via bank”.

Endpoint URL: POST /bank-top-ups/create-request

Parameters: TOKEN - identifies authenticated user

```
{  
    "coin": "string",  
    "amount": 1000,  
    "bankDetails": {  
        "fullName": "string"  
    }  
}
```

3. System Operator returns request to top up via bank information to user. (See Result example below)

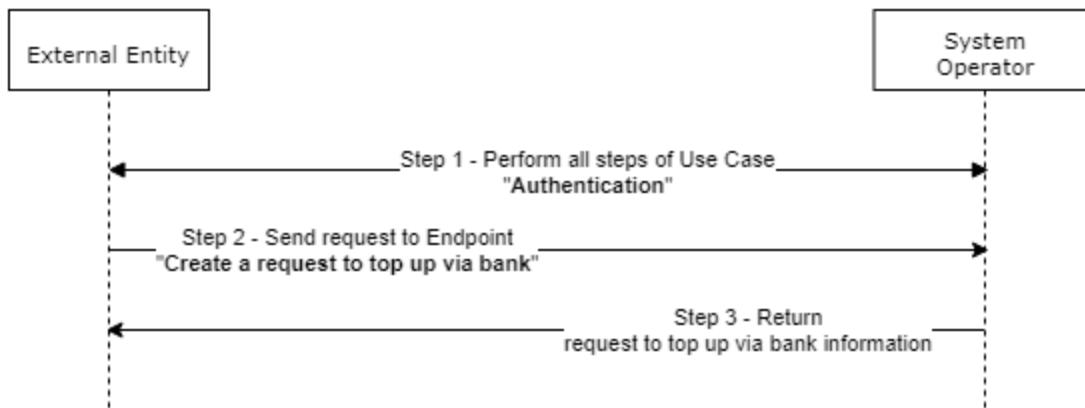
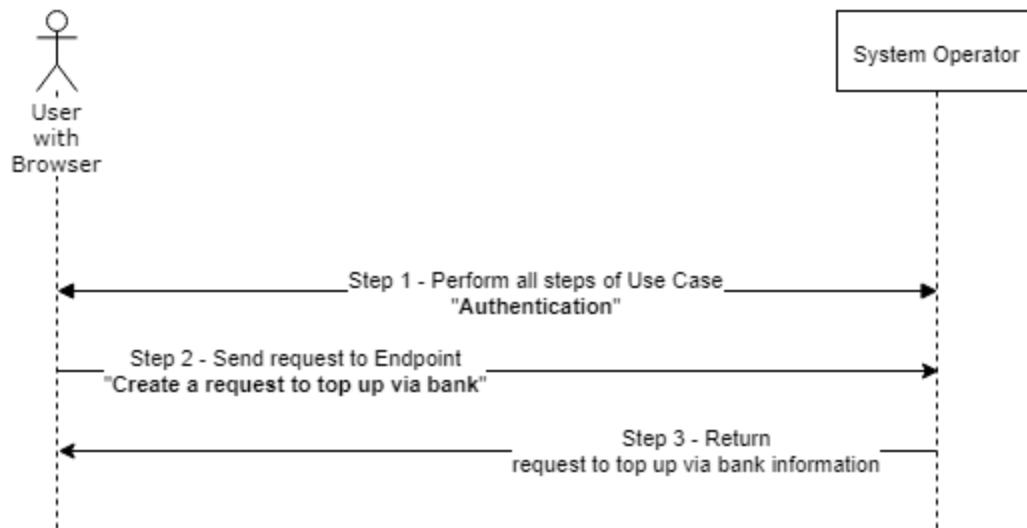
Post Conditions

Request to top up via bank is created.

Result example

```
{  
    "status": "ok",  
    "message": "processed successfully",  
    "process": {  
        "id": "3522b156-81eb-47b3-8191-55632f595ac1",  
        "createdAt": "2019-06-21T12:12:09.142+02:00",  
        "updatedAt": "2019-06-21T12:12:09.142+02:00",  
        "type": "bank_topup",  
        "status": "pending",  
        "requestIdentifier": 1977156821,  
        "requestStatus": "pending",  
        "transactions": [ ],  
        "children": [ ],  
        "errorMessage": null,  
        "bankDetails": {  
            "fullName": "Big Bank",  
            "account": null,  
            "iban": null,  
            "bic": null,  
            "swift": null,  
            "name": null,  
            "address": null  
        },  
        "cashAmount": 1000,  
        "clientCoin": {  
            "serial": "708389635438",  
            "organizationId": "ba926ble-e03e-49ff-a496-ac690f5dab7c",  
            "organizationName": "individual",  
            "technical": false,  
            "type": "client",  
            "issuer": {  
                "id": "005a5496-d353-4020-8c35-dba8635a7858",  
                "sn": "EUR",  
                "currency": "EUR"  
            }  
        }  
    }  
}
```

Create a request to top up via bank scheme

Use case: Create a request to top up via bank**Basic FFlow****Optional Web UI Flow**

Accept to create request to top up via bank description

Use Case Name

Accept the create request to top up via bank

Brief Description

A User or External Entity on behalf of a User with role permission “BANK_TOP_UP_VALIDATION_EXECUTOR” will go through all steps of “Authentication” Use Case (to obtain token), and then send a request to Endpoint “Accept the create request to top up via bank”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: “BANK_TOP_UP_VALIDATION_EXECUTOR”.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Accept the create request to top up via bank”.

Endpoint URL: POST /bank-top-ups/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

requestIdentifier - identify the request to top up via bank

3. System Operator returns top up via bank information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Accept the create request to top up via bank”.

Endpoint URL: POST /bank-top-ups/{requestIdentifier}/accept

Parameters: TOKEN - identifies authenticated user

requestIdentifier - identify the request to top up via bank

3. System Operator returns top up via bank information to user. (See Result example below)

Post Conditions

Request to top up via bank is accepted.

Result example

```
{
  "status": "ok",
  "message": "processed successfully",
  "process": {
    "id": "3522b156-81eb-47b3-8191-55632f595ac1",
    "createdAt": "2019-06-21T12:12:09.142+02:00",
    "updatedAt": "2019-06-21T16:41:41.363+02:00",
    "type": "bank_topup",
  }
}
```

```
        "status": "processed",

        "requestIdentifier": 1977156821,

        "requestStatus": "processed",

        "transactions": [

            {

                "id": "6acef0b4-7fe9-4581-9f85-a1d6fc156ba2",

                "parentId": null,

                "type": "issue",

                "from": null,

                "to": {

                    "serial": "icsOmyRlFTao",

                    "organizationId": "ba926ble-e03e-49ff-a496-ac690f5dab7c",

                    "organizationName": "individual",

                    "technical": true,

                    "type": "client",

                    "issuer": {

                        "id": "005a5496-d353-4020-8c35-dba8635a7858",

                        "sn": "EUR",

                        "currency": "EUR"

                    }

                },

                "amount": 1000,

                "performedAt": "2019-06-21T16:41:41.363+02:00",

                "issuer": {


```

```
        "id": "005a5496-d353-4020-8c35-dba8635a7858" ,  
        "sn": "EUR" ,  
        "currency": "EUR"  
    }  
},  
{  
    "id": "3e05fac3-d9fe-48e9-8001-3ca976c9b3fd" ,  
    "parentId": null ,  
    "type": "merge" ,  
    "from": {  
        "serial": "icsOmyRlFTao" ,  
        "organizationId":  
        "ba926b1e-e03e-49ff-a496-ac690f5dab7c" ,  
        "organizationName": "individual" ,  
        "technical": true ,  
        "type": "client" ,  
        "issuer": {  
            "id": "005a5496-d353-4020-8c35-dba8635a7858" ,  
            "sn": "EUR" ,  
            "currency": "EUR"  
        }  
},  
    "to": {  
        "serial": "708389635438" ,  
        "organizationId":  
        "ba926b1e-e03e-49ff-a496-ac690f5dab7c" ,  
    }
```

```
        "organizationName": "individual",
        "technical": false,
        "type": "client",
        "issuer": {
            "id": "005a5496-d353-4020-8c35-dba8635a7858",
            "sn": "EUR",
            "currency": "EUR"
        },
        "amount": 1000,
        "performedAt": "2019-06-21T16:41:41.363+02:00",
        "issuer": {
            "id": "005a5496-d353-4020-8c35-dba8635a7858",
            "sn": "EUR",
            "currency": "EUR"
        }
    },
    "children": [],
    "errorMessage": null,
    "bankDetails": {
        "fullName": "Big Bank",
        "account": null,
        "iban": null,
        "bic": null,
    }
]
```

```
        "swift": null,  
  
        "name": null,  
  
        "address": null  
  
    },  
  
    "cashAmount": 1000,  
  
    "clientCoin": {  
  
        "serial": "708389635438",  
  
        "organizationId": "ba926b1e-e03e-49ff-a496-ac690f5dab7c",  
  
        "organizationName": "individual",  
  
        "technical": false,  
  
        "type": "client",  
  
        "issuer": {  
  
            "id": "005a5496-d353-4020-8c35-dba8635a7858",  
  
            "sn": "EUR",  
  
            "currency": "EUR"  
  
        }  
  
    }  

```

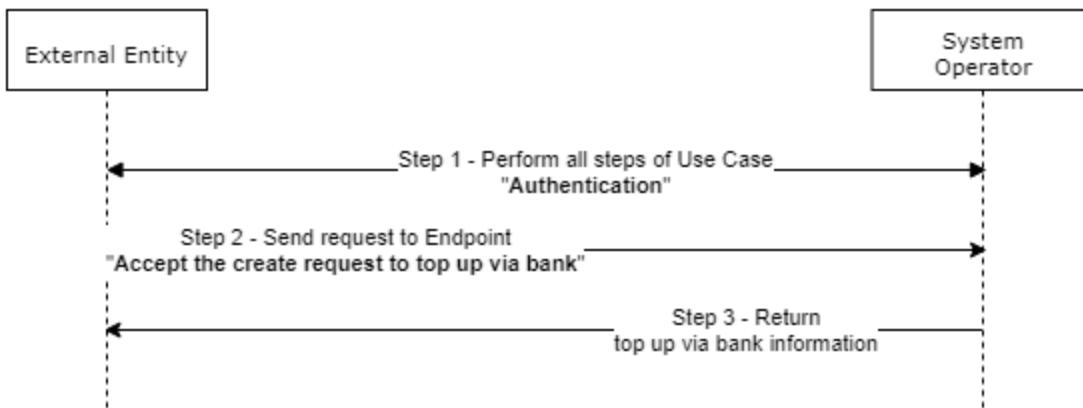
```
}
```

```
}
```

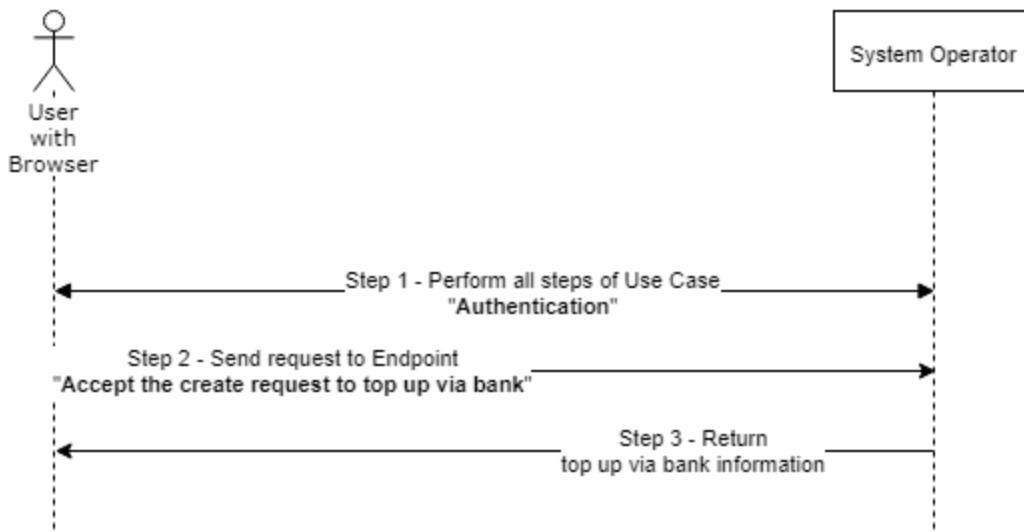
Accept to create request to top up via bank scheme

Use case: Accept the create request to top up via bank

Basic FFlow



Optional Web UI Flow



Decline to create request to top up via bank description

Use Case Name

Decline the create request to top up via bank

Brief Description

A User or External Entity on behalf of a User with role permission “BANK_TOP_UP_VALIDATION_EXECUTOR” will go through all steps of “Authentication” Use Case (to obtain token), and then send a request to Endpoint “Decline the create request to top up via bank”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: “BANK_TOP_UP_VALIDATION_EXECUTOR”.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Decline the create request to top up via bank”.

Endpoint URL: POST /bank-top-ups/{requestIdentifier}/decline

Parameters: TOKEN - identifies authenticated user

requestIdentifier - identify the request to top up via bank

3. System Operator returns declined top up via bank information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Decline the create request to top up via bank”.

Endpoint URL: POST /bank-top-ups/{requestIdentifier}/decline

Parameters: TOKEN - identifies authenticated user

requestIdentifier - identify the request to top up via bank

3. System Operator returns declined top up via bank information to user. (See Result example below)

Post Conditions

Request to top up via bank is declined.

Result example

```
{
  "status": "ok",
  "message": "processed successfully",
  "process": {
    "id": "eb5efca4-b9b8-4908-8c81-c9bebdcfd84a",
    "createdAt": "2019-06-21T16:58:37.423+02:00",
    "updatedAt": "2019-06-21T16:59:09.059+02:00",
  }
}
```

```
"type": "bank_topup",

"status": "declined",

"requestIdentifier": 1258501759,

"requestStatus": "declined",

"transactions": [ ],

"children": [ ],

"errorMessage": null,

"bankDetails": {

    "fullName": "Big Bank",

    "account": null,

    "iban": null,

    "bic": null,

    "swift": null,

    "name": null,

    "address": null

} ,

"cashAmount": 9000,

"clientCoin": {

    "serial": "708389635438",

    "organizationId": "ba926b1e-e03e-49ff-a496-ac690f5dab7c",

    "organizationName": "individual",

    "technical": false,

    "type": "client",

    "issuer": {

        "id": "005a5496-d353-4020-8c35-dba8635a7858"

    }

}
```

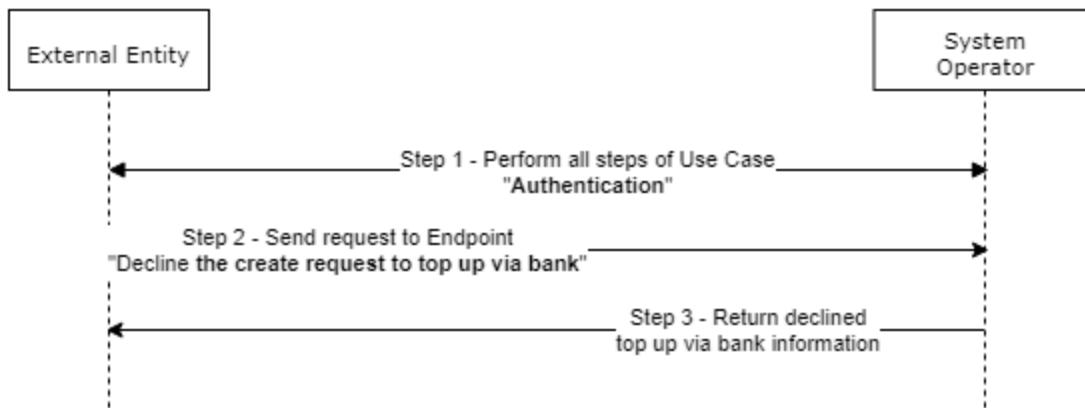
```
        "sn": "EUR",  
        "currency": "EUR"  
    }  
}  
}  
}
```



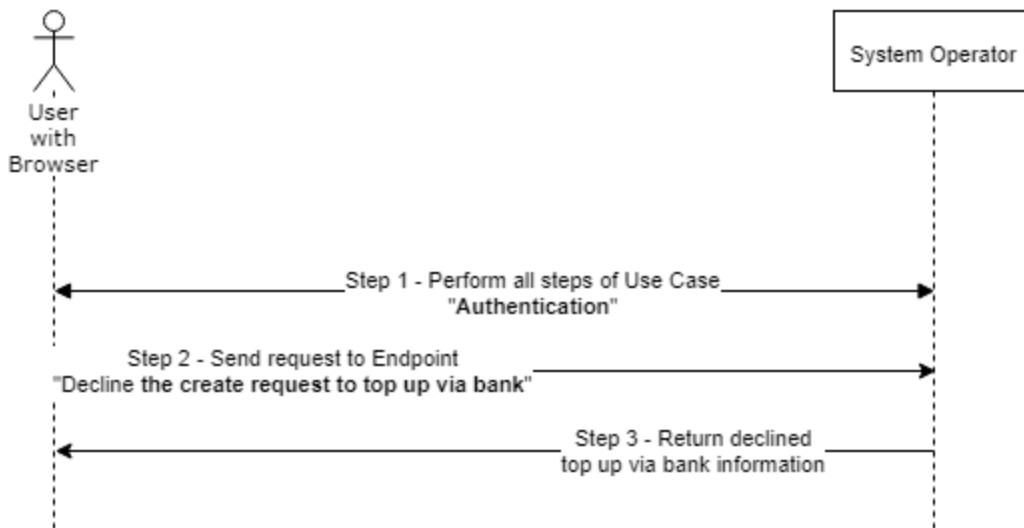
Decline to create request to top up via bank scheme

Use case: Decline the create request to top up via bank

Basic FFlow



Optional Web UI Flow



Calculate commission for top up via bank description

Use Case Name

Calculate commission for top-up via bank

Brief Description

When System Operator Users add money to their Coins System Operator charges some commission for this operations.

The amount is calculated based on the parameters set up in the System Operator configurations, actual deposit amount and on the existing contracts with a User.

Actors

1. System Operator using SDK.Finance software.
2. External Entity using System Operator API.

Preconditions

This can only be done by the authenticated entity.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request with all the needed information to calculate the transaction commission.

Endpoint URL: /bank-top-ups/calculate

```
{
  "coin": "string",
  "amount": 0
}
```

3. System Operator validates arrived information, calculates and returns Commission Amount.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The user sends a request with all the needed information to calculate the transaction commission.

Endpoint URL: /bank-top-ups/calculate

```
{
  "coin": "string",
  "amount": 0
}
```

3. System Operator validates arrived information, calculates and returns a web page with Commission Amount.

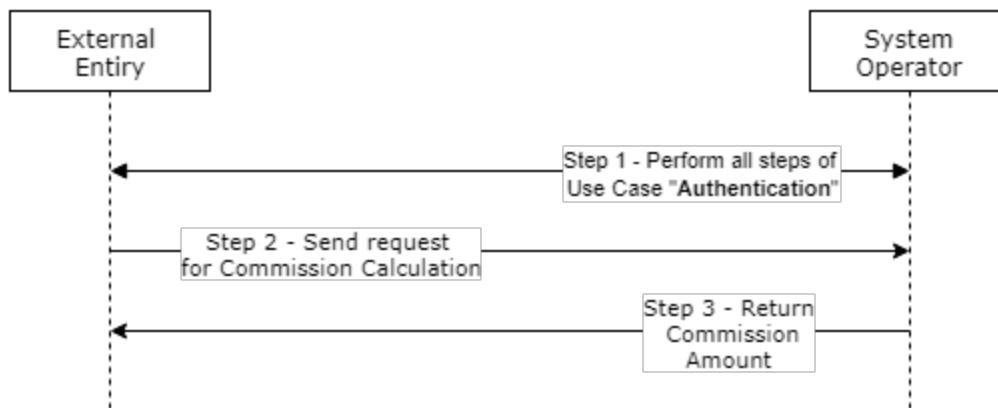
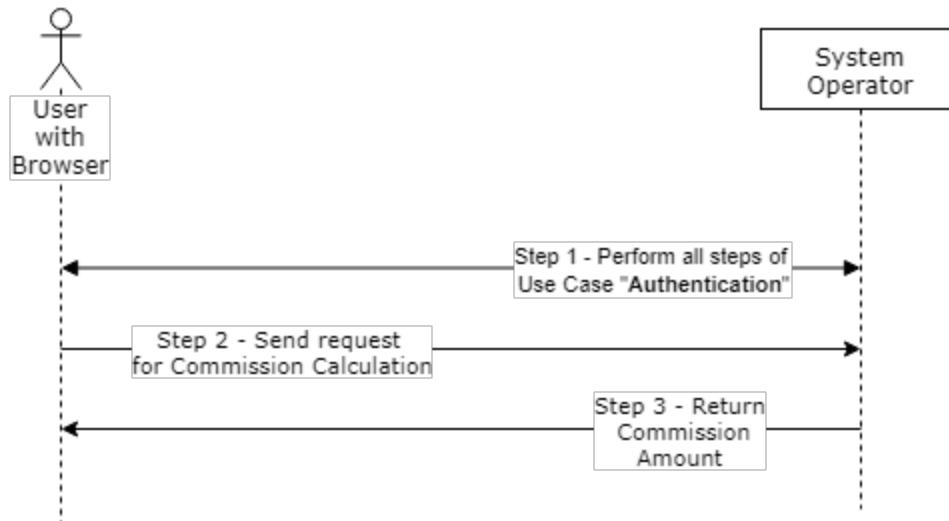
Post Conditions

The Commission Amount calculated exactly as specified in the Contract.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for top up via bank scheme

Use case: Calculate commission for top up via bank**Basic FLow****Optional Web UI Flow**

Perform top up via bank description

Use Case Name

Perform top up via bank

Brief Description

Owner of a Coin or External Entity acting as Owner with the help of other actors adds money to the Coin.

In the process, another Use Case is executed. This Use Case involves some steps outside of System Operator information movement. The owner needs to physically go to a Bank where System Operator has an account and deposit Hard Cash to System Operator's bank account.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running SDK.Finance software and exposing portfolio of financial APIs.]

Preconditions

1. The Coin must be active and in good standing.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. System Operator informs Coin Owner about the procedure for Top-up Via Bank.
2. Coin Owner goes to System Operator's Bank, deposits some Hard Cash and gets from the Bank all transaction details.
3. Coin Owner contacts Accountant and communicates all deposit transaction details.
4. External Entity representing Accountant contacts the Bank and verifies that the money was really deposited.
5. External Entity representing Accountant performs all steps of Use Case "Authentication".
6. External Entity updates the owner's Coin balance to reflect the deposit.

Endpoint URL: POST /bank-top-ups

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "bankDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

7. System Operator returns result status.

Optional Flow with Web Browser UI

1. System Operator informs Coin Owner about the procedure for Top-up operation Via Bank.
2. Coin Owner goes to System Operator's Bank, deposits some Hard Cash and gets from the Bank all deposit transaction details.
3. Coin Owner contacts Accountant and communicates all deposit transaction details.

4. Accountant contacts the Bank and verifies that the money was really deposited.
5. Accountant performs all steps of Use Case “Authentication”.
6. External Entity updates the owner’s Coin balance to reflect the deposit.

Endpoint URL: POST /bank-top-ups

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "bankDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

7. System Operator returns result status.

Post Conditions

1. After some time the Coin Owner sees the deposited money in the Coin.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-06-25T11:17:03.840Z",
    "updatedAt": "2018-06-25T11:17:03.840Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "value": 0
        }
      }
    ]
  }
}
```

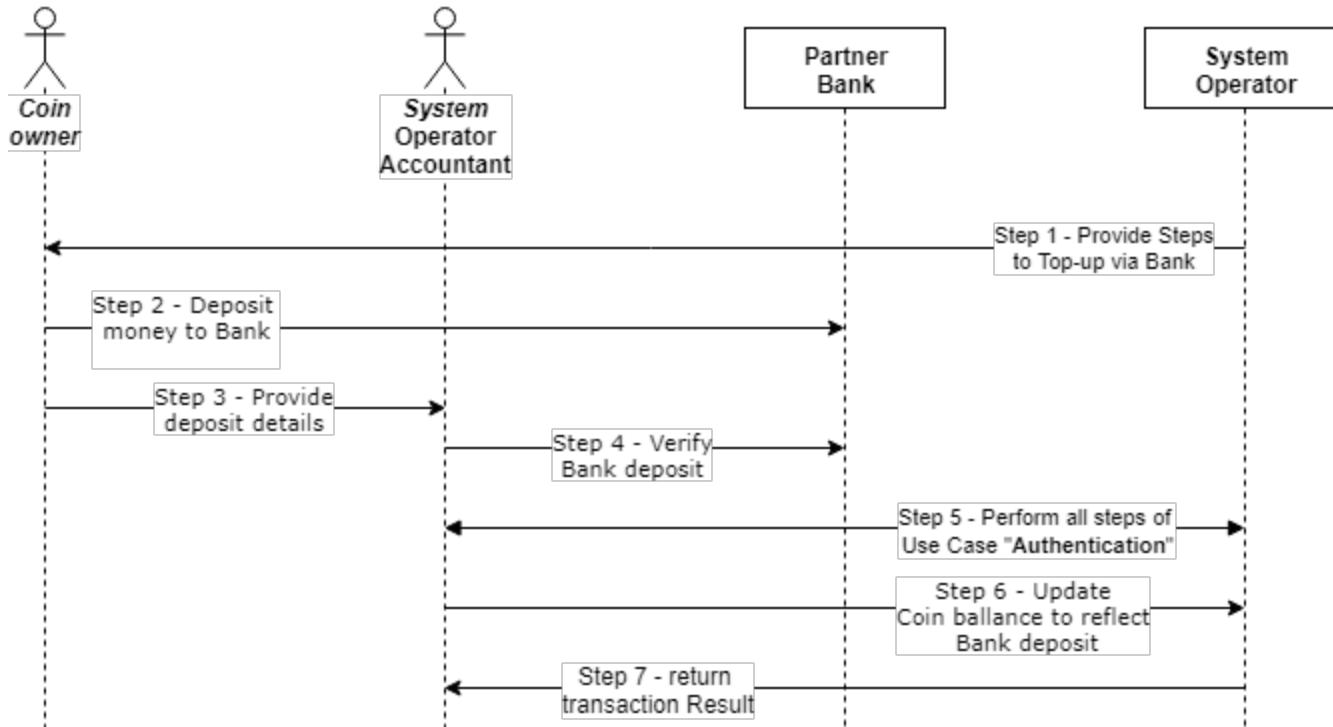
```
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "to": {
        "serial": "string",
        "organizationId": "string",
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "amount": 0,
    "performedAt": "2018-06-25T11:17:03.840Z",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
],
"children": [
    {
        ...
    }
],
"errorMessage": "string"
},
```

```
"status": "ok",
"message": "string"
}
```

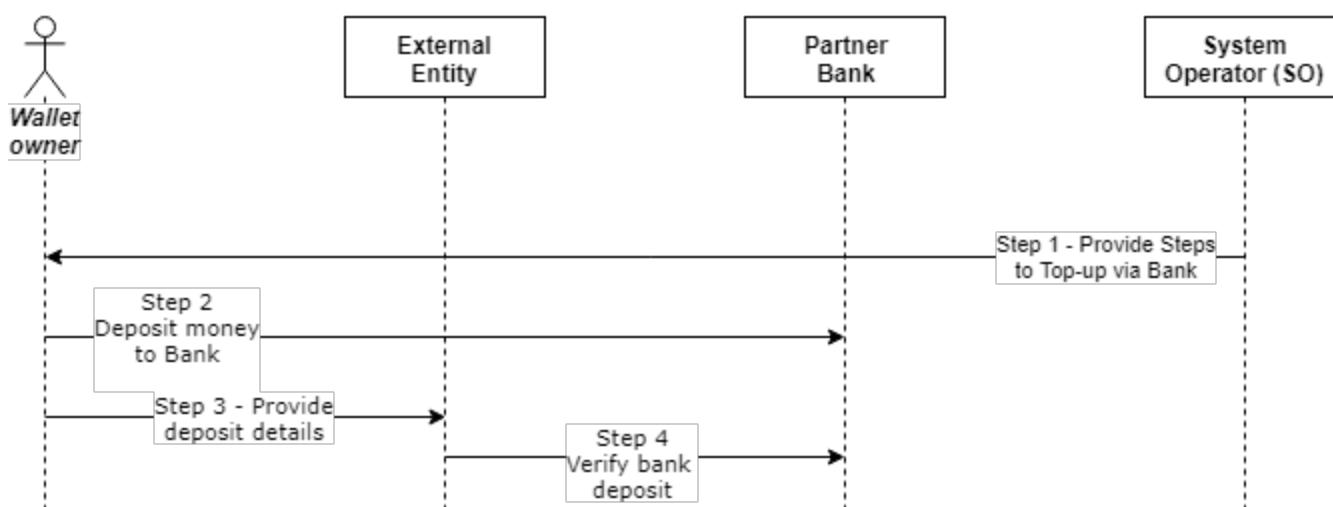
Perform top up via bank scheme

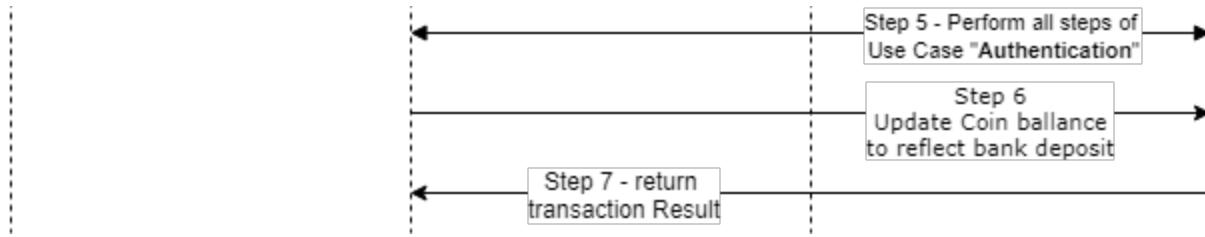


Top up wallet via bank



Optional Web UI Flow





Top up via cash desk

Calculate commission for top-up via cash desk description

Use Case Name

Calculate commission for top-up via cash desk

Brief Description

When System Operator Users adds money to their Coins, System Operator charges some commission for this operations. The amount is calculated based on the parameters set up in the System Operator configurations, actual deposit amount and on the existing contracts with a User.

Actors

1. System Operator using SDK.Finance software.
2. External Entity using System Operator API.

Preconditions

This can only be done by the authenticated entity.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request with all the needed information to calculate the transaction commission.

Endpoint URL: POST /cash-desk-top-ups/calculate

Parameters:

```
{
  "amount": 0,
  "serial": "string",
  "fullName": "string",
  "passportData": "string"
}
```

3. System Operator validates arrived information, calculates and returns Commission Amount.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request with all the needed information to calculate the transaction commission.

Endpoint URL: /cash-desk-top-ups/calculate

Parameters:

```
{  
    "amount":0,  
    "serial":"string",  
    "fullName":"string",  
    "passportData":"string"  
}
```

3. System Operator validates arrived information, calculates and returns a web page with Commission Amount.

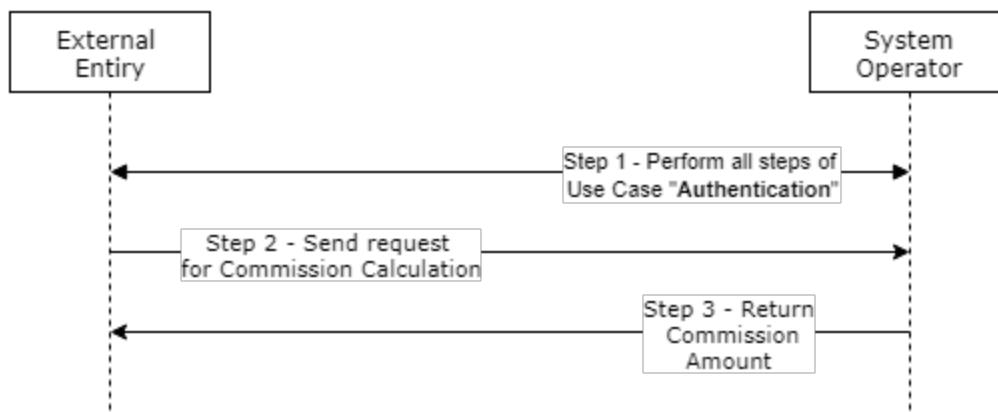
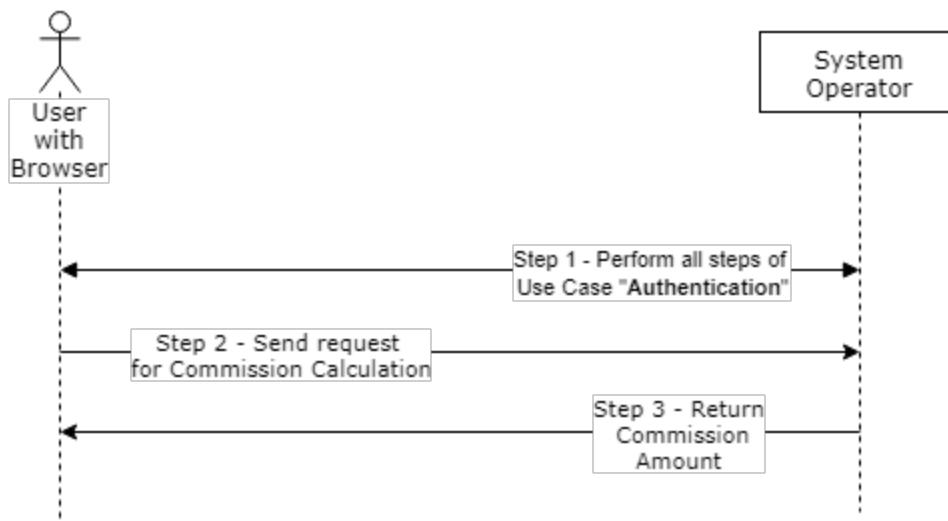
Post Conditions

The Commission Amount calculated exactly as specified in the Contract.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for top-up via cash desk scheme

Use case: Calculate commission for top-up via cash desk**Basic FLow****Optional Web UI Flow**

Top up via cash desk description

Use Case Name

Top up via cash desk

Brief Description

A person comes to a Cashier Office and wants to add money to a Coin. The person may not be the actual Coin owner, just needs to know valid Coin identifiers like telephone or email address. Cashier interacts with System Operator API Platform, runs the predefined procedure, accepts the hard cash, adds received amount to a Coin balance and provides a person with transaction receipt.

Actors

1. Any individual who wants to Top-up existing wallet at System Operator.
2. Cashier - official representative of System Operator.
3. System Operator Platform.

Preconditions

An Individual who wants to populate a Coin needs to know valid User Identifier.

A user should have a Coin with desired Currency.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. External Entity System Operator Cashier starts Cash Desk Day - Use Case Open working day.

Endpoint URL: POST /working-days

Parameters: TOKEN

2. An individual comes to the office, gives all identifying information and wants to add money to a Coin.
3. External Entity Cashier sends Request to see all User Coins - Get coins owned by the current user

Endpoint URL: GET /coins

Parameters: TOKEN identifying a User

4. System Operator Platform returns the list of Coins.
5. External Entity Cashier selects Coin, enters the amount to deposit, and submits to System Operator Platform Request - Top up via cash desk

Endpoint URL: POST /cash-desk-top-ups

Parameters:

```
{
  "amount": 0,
  "serial": "156407768719",
  "fullName": "EUR",
  "passportData": "string"
}
```

6. External Entity Cashier sends Request - Calculate commission for top-up via cash desk

Endpoint URL: POST /cash-desk-top-ups/calculate

Parameters:

```
{
  "amount":0,
  "serial":"string",
  "fullName":"string",
  "passportData":"string"
}
```

7. System Operator calculates and returns Top-up transaction Commission
8. External Entity calculates total amount and gives Hard Cash to the Cashier.

Optional Flow with Web Browser UI

1. Cashier starts Cash Desk Day.

Endpoint URL: POST /working-days

Parameters: TOKEN

2. An individual comes to the office, gives all identifying information and wants to add money to a Coin.
3. Cashier sends Request to see all User Coins - Get coins owned by the current user

Endpoint URL: GET /coins

Parameters: TOKEN identifying a User

4. System Operator Platform returns the list of Coins.

5. Cashier selects Coin, enters the amount to deposit, and submits to System Operator Platform Request - Top up via cash desk

Endpoint URL: POST /cash-desk-top-ups

Parameters:

```
{
  "amount":0,
  "serial": "156407768719",
  "fullName": "EUR",
  "passportData": "string"
}
```

6. Cashier sends Request - Calculate commission for top-up via cash desk

Endpoint URL: POST /cash-desk-top-ups/calculate

Parameters:

```
{
  "amount":0,
  "serial": "string",
  "fullName": "string",
  "passportData": "string"
}
```

7. System Operator calculates and returns Top-up transaction Commission

8. Individual calculates total amount and gives Hard Cash to the Cashier.

Post Conditions

The amount deposited is reflected in the corresponding Coin.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-06-28T08:20:53.948Z",
    "updatedAt": "2018-06-28T08:20:53.948Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "to": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "amount": 0,
        "performedAt": "2018-06-28T08:20:53.948Z",
        "issuer": {
          "id": "string",
          "sn": "string",
          "currency": "string"
        }
      }
    ]
  }
}
```

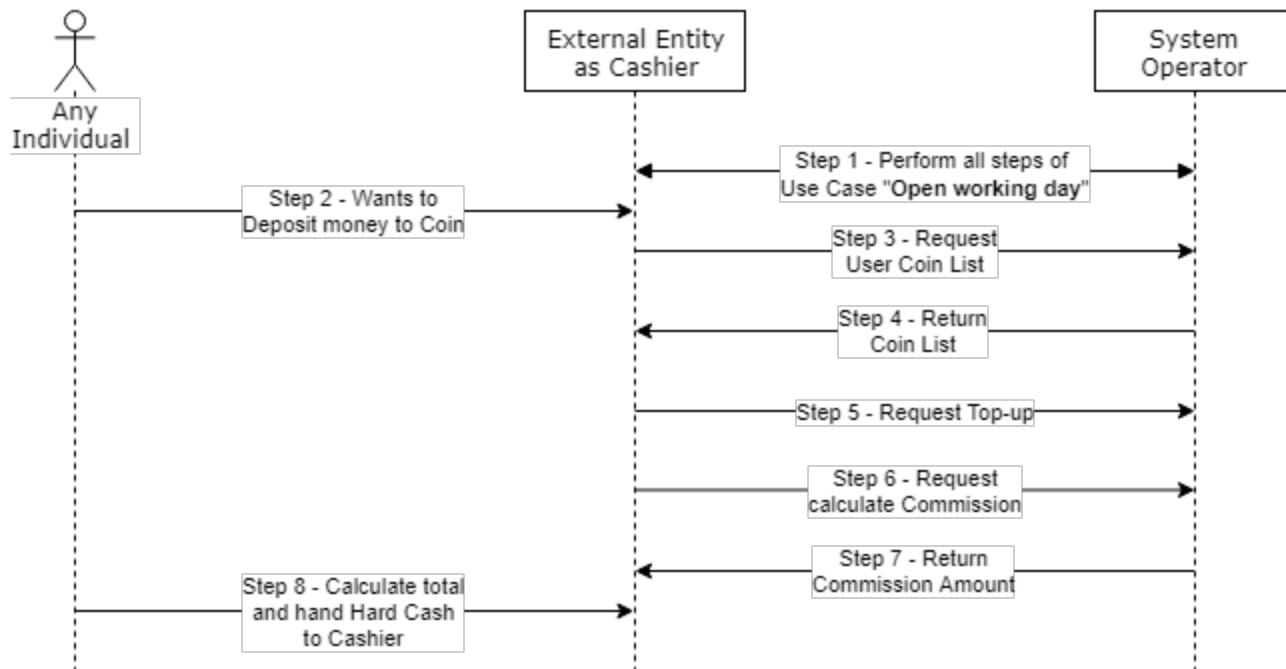
```
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
},
],
"children": [
{
    ...
},
],
"errorMessage": "string"
},
}
```

```
"status": "ok",
"message": "string"
}
```

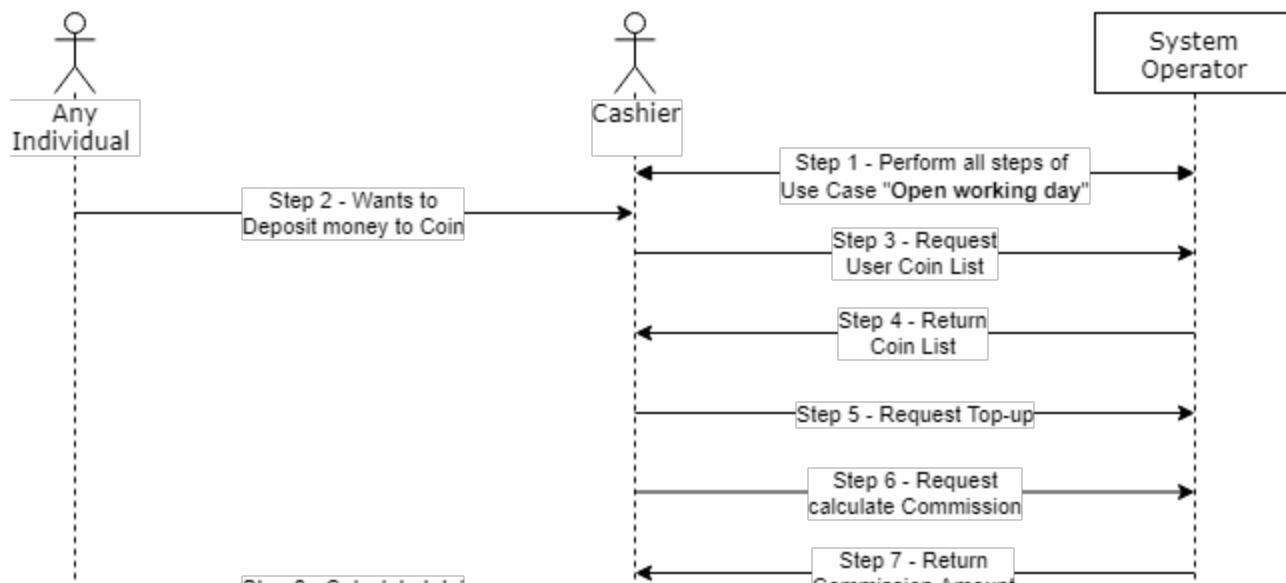
Top up via cash desk scheme

Use case: Top up via cash desk

Basic Flow



Optional Web UI Flow





Withdrawal via bank

Calculate commission for withdrawal via bank description

Use Case Name

Calculate commission for withdrawal via bank

Brief Description

A User or External Entity on behalf of a User with role permission BANK_WITHDRAWAL_CREATION_EXECUTOR will go through all steps of “Get coins owned by current user” Use Case and then send a request to Endpoint “Calculate commission for withdrawal via bank”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
BANK_WITHDRAWAL_CREATION_EXECUTOR
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get coins owned by current user”.
2. External Entity sends a request to Endpoint “Calculate commission for withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/calculate

Parameter:

```
{
  "coin": "string",
  "amount": 0
}
```

“coin”: “string”, - Serial number of the coin where funds can be withdrawn from

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get coins owned by current user”.
2. A user sends a request to Endpoint “Calculate commission for withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/calculate

Parameter:

```
{  
    "coin": "string",  
    "amount": 0  
}
```

"coin": "string", - Serial number of the coin where funds can be withdrawn from

3. System Operator returns result information to User (See Result example below).

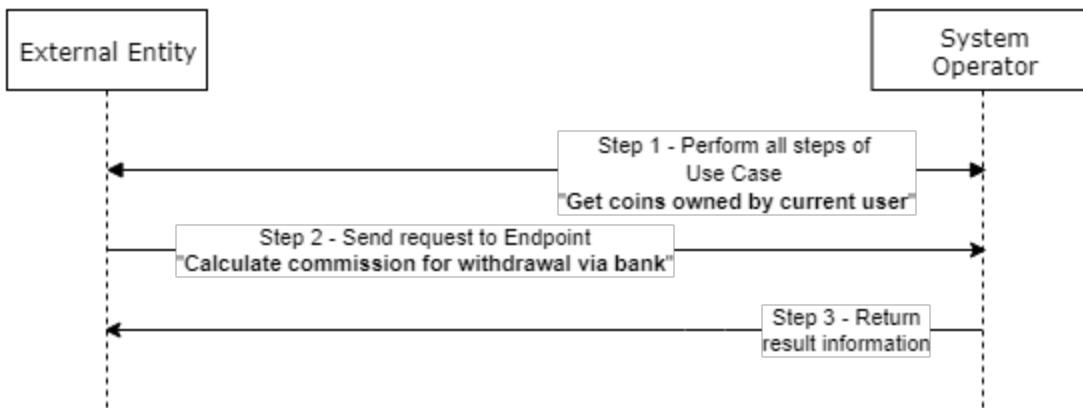
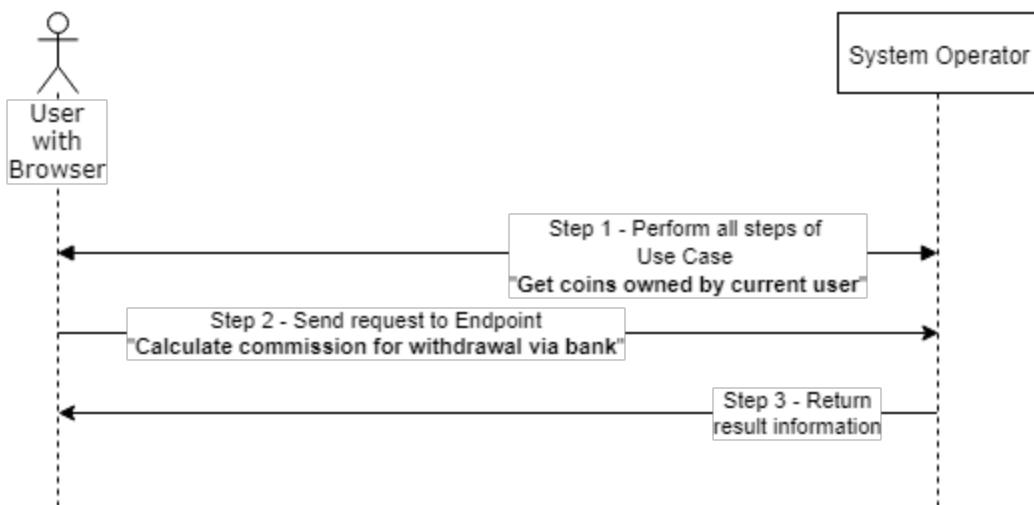
Post Conditions

Right commission amount is available.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for withdrawal via bank scheme

Use case: Calculate commission for withdrawal via bank**Basic FFlow****Optional Web UI Flow**

Create a request to withdraw via bank description

Use Case Name

Create a request to withdraw via bank

Brief Description

During this Use Case, an External Entity or an Individual sends to make a withdrawal from a particular wallet via bank account associated with this wallet/currency.

During execution, there is also a step to calculate a commission amount for this transaction executing use case “Calculate commission for withdrawal via bank”. If wallet user agrees to commision amount the user executes use case “Accept withdrawal via bank”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. External Entity acting as a User performs all steps of Use Case “Authentication”.
2. External Entity acting as a User sends a request to make Withdrawal via Bank

Endpoint URL: POST /bank-withdrawals

Parameter:

```
{
  "coin": "string",
  "amount": 0,
  "bankDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator creates a Withdrawal and returns ID.

4. External Entity acting as User sends a request “Calculate commission for withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/calculate

Parameter:

```
{
  "coin": "string",
  "amount": 0
}
```

5. System Operator calculates and returns commission amount.

Response:

```
{
  "status": "ok",
  "message": "string",
  "transactionAmount": 0,
  "senderAmountPush": 0,
  "recipientAmountPush": 0,
  "commissionAmountPush": 0,
  "issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
  }
}
```

6. External Entity as User reviews details and sends a request “Accept withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/{requestIdentifier}/accept

7. External Entity as Accountant reviews a request and sends a request “Approve withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/{requestIdentifier}/approve (need to create)

8. System operator sends a request to the partner bank to transfer money to a User’s Bank Account.

9. System Operator’s partner bank transfers the money to User’s Bank Account.

10. User withdraws the money from Bank account.

Optional Flow with Web Browser UI

1. The user performs all steps of Use Case “Authentication”.

2. The user sends a request to make Withdrawal via Bank

Endpoint URL: POST /bank-withdrawals

Parameter:

```
{
  "coin": "string",
  "amount": 0,
  "bankDetails": {
    "fullName": "string",
    "account": "string",
    "iban": "string",
    "bic": "string",
    "swift": "string",
    "name": "string",
    "address": "string"
  }
}
```

3. System Operator creates a Withdrawal and returns ID.
4. User sends a request “Calculate commission for withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/calculate

Parameter:

```
{
  "coin": "string",
  "amount": 0
}
```

5. System Operator calculates and returns commission amount.

Response:

```
{
  "status": "ok",
  "message": "string",
  "transactionAmount": 0,
  "senderAmountPush": 0,
  "recipientAmountPush": 0,
  "commissionAmountPush": 0,
  "issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
  }
}
```

6. User reviews details and sends a request “Accept withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/{requestIdentifier}/accept

7. Accountant reviews a request and sends a request “Approve withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/{requestIdentifier}/approve (need to create)

8. System operator sends a request to the partner bank to transfer money to a User’s Bank Account.

9. System Operator’s partner bank transmits the money to User’s Bank Account.

10. User withdraws the money from Bank account.

Alternative flow

At step 7 above an Account can send request “Decline withdrawal via bank”.

Endpoint URL: POST /bank-withdrawals/{requestIdentifier}/decline

Post Conditions

External Entity or User can withdraw the money.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-07-23T09:40:15.022Z",
    "updatedAt": "2018-07-23T09:40:15.022Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "to": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        }
      }
    ]
  }
}
```

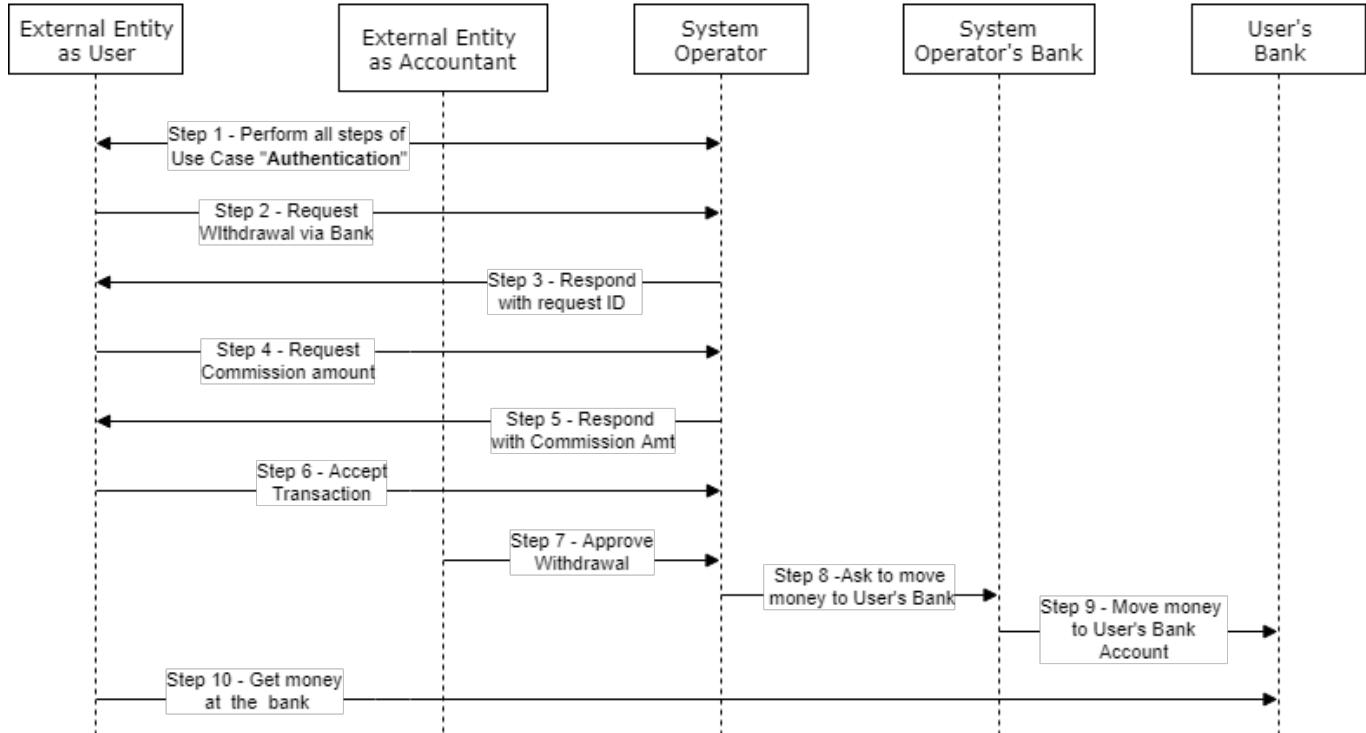
```
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
},
"amount": 0,
"performedAt": "2018-07-23T09:40:15.022Z",
"issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
}
],
"children": [
    {
        ...
    },
    ...
],
"errorMessage": "string"
},
```

```
"status": "ok",  
"message": "string"  
}
```

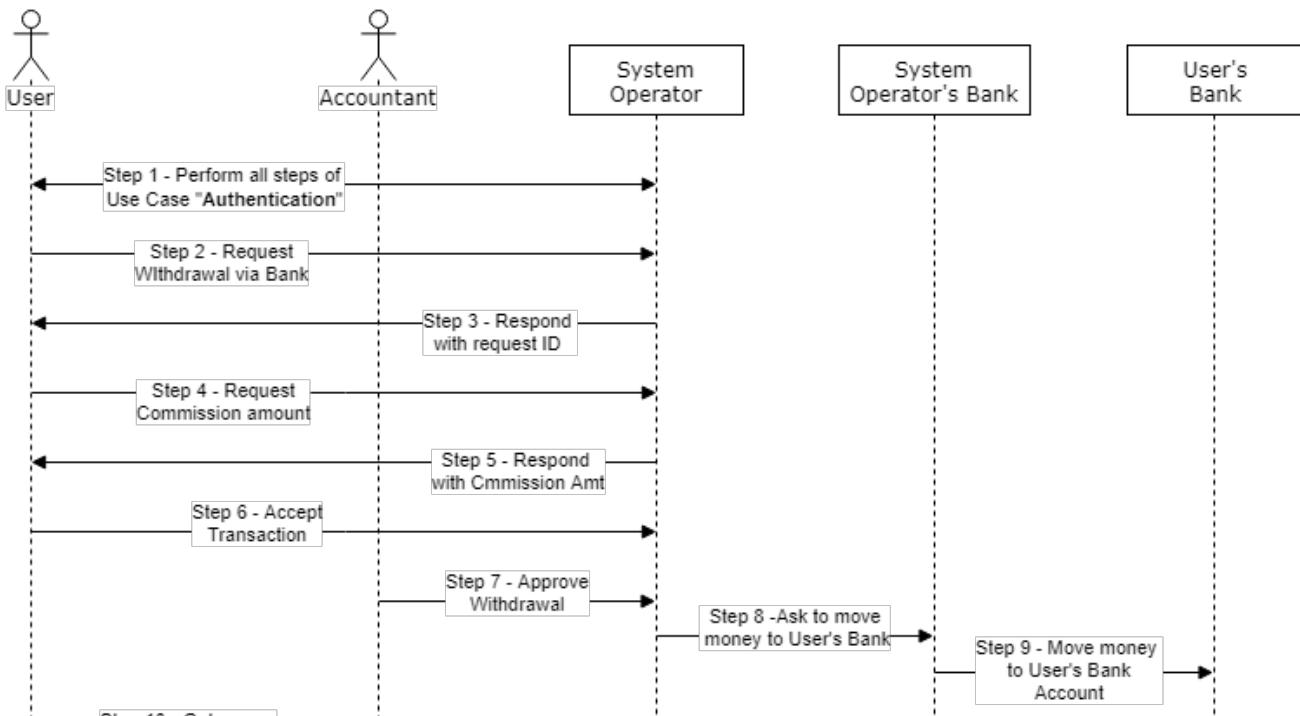
Create a request to withdraw via bank scheme

Use case: Create a request to withdraw via bank

Basic Flow



Optional Web UI Flow





Lift limit for withdrawal via bank description

Use Case Name

Lift limit for withdrawal via bank

Brief Description

External entity sends to System Operator API all information needed to change withdrawal amount limit.

Actors

1. External Entity communicating with System Operator API and acting on behalf of CFO.
2. System Operator.

Preconditions

This can only be done by the authenticated entity.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity

1. Perform all steps of Use Case “Authentication”.
2. External Entity acting as System Operator CFO sends to System Operator API a request with all information needed to change a withdrawal via Bank maximum amount.

Endpoint URL: /bank-withdrawals/{requestIdentifier}/lift-limit

Request parameters:

```
{
  "id": "string",
  "parentId": "string",
  "type": "transfer",
  "from": {
    "serial": "string",
    "organizationId": "string",
    "type": "regular_commission",
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    }
  },
  "to": {
    "serial": "string",
    "organizationId": "string",
    "type": "regular_commission",
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    }
  },
  "limitAmount": 0
}
```

3. System Operator returns a notification with the operation result: ok/ fail

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. System Operator CFO sends to System Operator API a request with all information needed to change a withdrawal via Bank maximum amount.

Endpoint URL: /bank-withdrawals/{requestIdentifier}/lift-limit

Request parameters:

```
{
  "id": "string",
  "parentId": "string",
  "type": "transfer",
  "from": {},
  "to": {
    "serial": "string",
    "organizationId": "string",
    "type": "regular_commission",
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    }
  },
  "limitAmount": 0
}
```

3. System Operator returns a notification with the operation result: ok/ fail

Post Conditions

New withdrawal limit has taken effect.

Result

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-06-28T08:20:53.921Z",
    "updatedAt": "2018-06-28T08:20:53.921Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        }
      }
    ]
  }
}
```

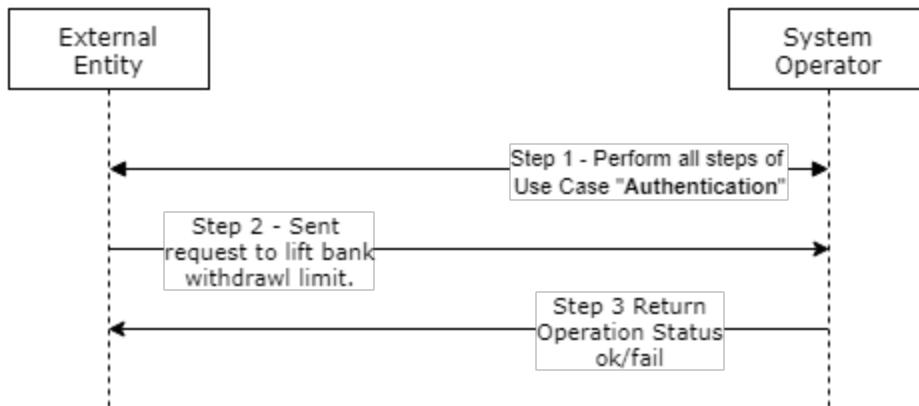
```
        "sn": "string",
        "currency": "string"
    }
},
"to": {
    "serial": "string",
    "organizationId": "string",
    "organizationName": "string",
    "technical": false,
    "type": "regular_commission",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
},
"amount": 0,
"performedAt": "2018-06-28T08:20:53.921Z",
"issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
}
},
"children": [
{
}
],
"errorMessage": "string"
},
```

```
"status": "ok",
"message": "string"
}
```

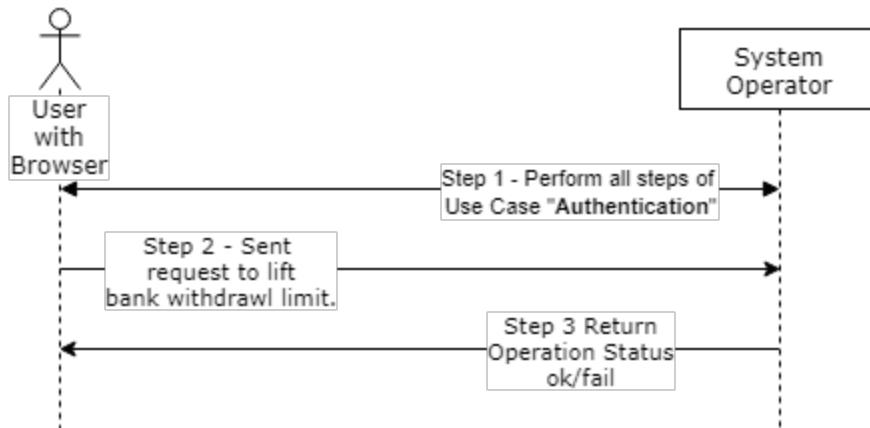
Lift limit for withdrawal via bank scheme

Use case: Lift limit for withdrawal via bank

Basic FLow



Optional Web UI Flow



Withdraw via cash desk

Calculate commission for withdrawal via specified cash desk description

Use Case Name

Calculate commission for withdrawal via specified cash desk

Brief Description

External Entity acts as a registered User inside System Operator.

System Operator is the company using “SDK.Finance” software.

External Entity calls System Operator API function to get the list of Coins for Authenticated user.

Each Coin is associated with only one Currency. User selects a Coin of interest.

External Entity requests all cash desks which handle the selected Coins/Currencies.

Now, having a Coin and a Cash desk commission amount could be calculated.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. Valid User Profile
2. Valid Coin

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
 2. External Entity requests the list of Coins of Authenticated User.
- Endpoint URL: GET /coins
3. System Operator returns a List of all Coins.

```
{
  "coins": [
    {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      }
    },
    ...
  ],
  "status": "ok",
  "message": "string"
}
```

4. External Entity sends a request to System Operator to get a list of Cash Desks operating in specified Currency.

Endpoint URL: /cash-desks/view

Parameters:

```
{
    "coinSerial": "string"
}
```

5. System Operator returns a list of Cash Desks.

6. External Entity sends a request to get commission amount

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
    "coin": "string",
    "amount": 0,
    "receiverName": "string"
}
```

7. System Operator returns a commission amount.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. User requests list of Coins of Authenticated User.

Endpoint URL: GET /coins

3. System Operator returns a List of all Coins.

```
{
    "coins": [
        {
            "serial": "string",
            "name": "string",
            "amount": 0,
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        }
    ],
    "status": "ok",
    "message": "string"
}
```

4. The user sends a request to System Operator to get a list of Cash Desks operating in specified Currency.

Endpoint URL: /cash-desks/view

Parameters:

```
{
    "coinSerial": "string"
}
```

5. System Operator returns a list of Cash Desks.

6. The user sends a request to get commission amount

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
    "coin": "string",
    "amount": 0,
    "receiverName": "string"
}
```

7. System Operator returns a commission amount.

Post Conditions

Commission amount is available.

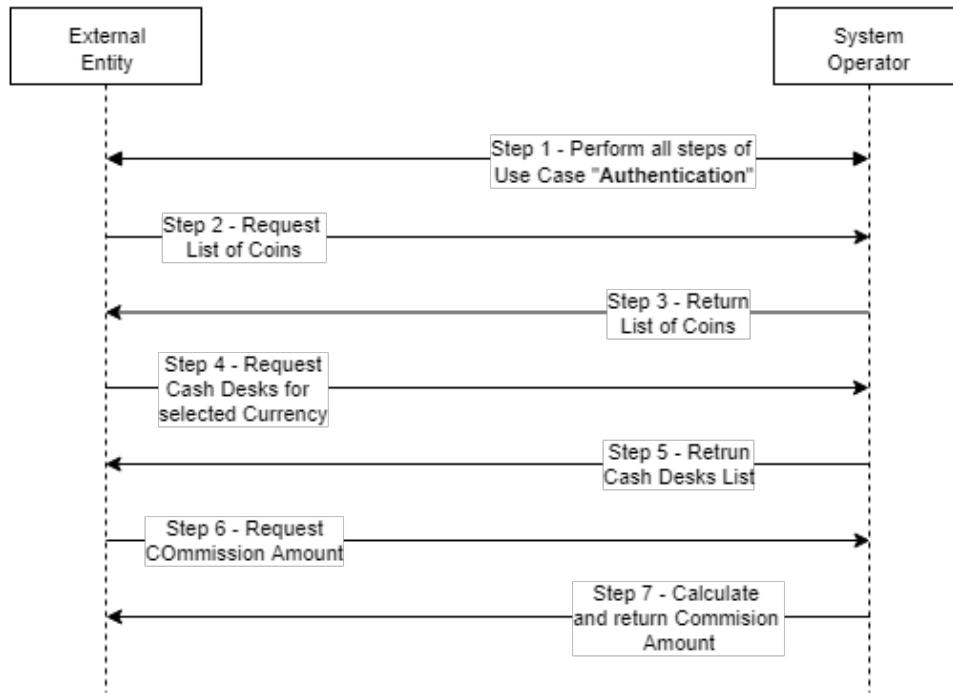
Result example

```
{
    "status": "ok",
    "message": "string",
    "transactionAmount": 0,
    "senderAmountPush": 0,
    "recipientAmountPush": 0,
    "commissionAmountPush": 0,
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
```

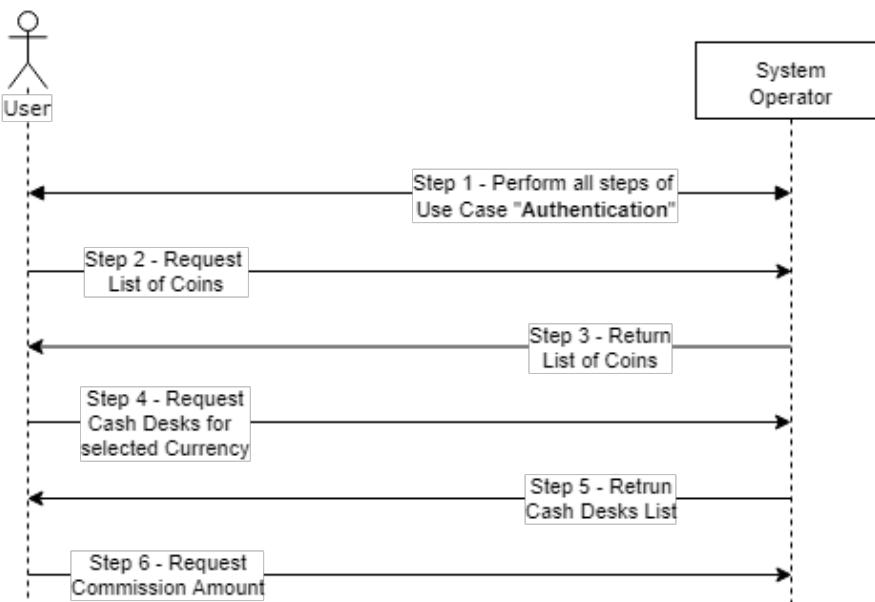
Calculate commission for withdrawal via specified cash desk scheme

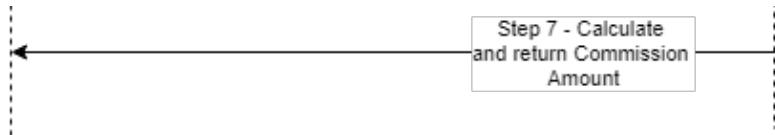
Use case: Calculate commission for withdrawal via specified cash desk

Basic Flow



Optional Web UI Flow





Execute withdrawal via specified cash desk description

Use Case Name

Execute withdrawal via specified cash desk

Brief Description

External Entity acts as a registered User inside System Operator.

System Operator is the company using “SDK.Finance” software.

External Entity calls System Operator API function with a valid Coin Identifier as the parameter.

Each Coin is associated with only one Currency.

System Operator determines a needed Currency from Coin parameter and returns to External Entity a list of all Cash Desks that operate in that Currency. User request commission amount and sends a request to execute withdrawal.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. External Entity or Cashier.
3. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. Valid User Profile
2. Valid Coin

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity .

1. Perform all steps of Use Case “Authentication”.
2. External Entity requests the list of Coins of Authenticated User.

Endpoint URL: GET /coins

3. System Operator returns a List of all Coins.

```
{
  "coins": [
    {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      }
    }
  ],
  "status": "ok",
  "message": "string"
}
```

4. External Entity sends a request to System Operator to get a list of Cash Desks operating in specified Currency.

Endpoint URL: /cash-desks/view

Parameters:

```
{
  "coinSerial": "string"
}
```

5. System Operator returns a list of Cash Desks.

6. External Entity sends a request to calculate commission amount for withdrawal via cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "receiverName": "string"
}
```

7. System Operator calculates and return the commission amount.

Response:

```
{
    "status": "ok",
    "message": "string",
    "transactionAmount": 0,
    "senderAmountPush": 0,
    "recipientAmountPush": 0,
    "commissionAmountPush": 0,
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
```

8. System Operator returns a confirmation (see result example below)

9. System operator sends Withdrawal information to a cashier.

10. Cash receiver stops at Cash desk office and gets hard cash.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. User requests list of Coins of Authenticated User.

Endpoint URL: GET /coins

3. System Operator returns a List of all Coins.

```
{
    "coins": [
        {
            "serial": "string",
            "name": "string",
            "amount": 0,
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        }
    ],
    "status": "ok",
    "message": "string"
}
```

4. The user sends a request to System Operator to get a list of Cash Desks operating in specified Currency.

Endpoint URL: /cash-desks/view

Parameters:

```
{
    "coinSerial": "string"
}
```

5. System Operator returns a list of Cash Desks.
6. The user sends a request to calculate the commission amount for withdrawal via cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
    "coin": "string",
    "amount": 0,
    "receiverName": "string"
}
```

7. System Operator calculates and return the commission amount.

Response:

```
{
    "status": "ok",
    "message": "string",
    "transactionAmount": 0,
    "senderAmountPush": 0,
    "recipientAmountPush": 0,
    "commissionAmountPush": 0,
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
```

8. System Operator returns a confirmation (see result example below)
9. System operator sends Withdrawal information to a Cashier.
10. Cash receiver stops at Cash desk office and gets hard cash.

Post Conditions

Money transfer is available at the selected cash desk.

Result example

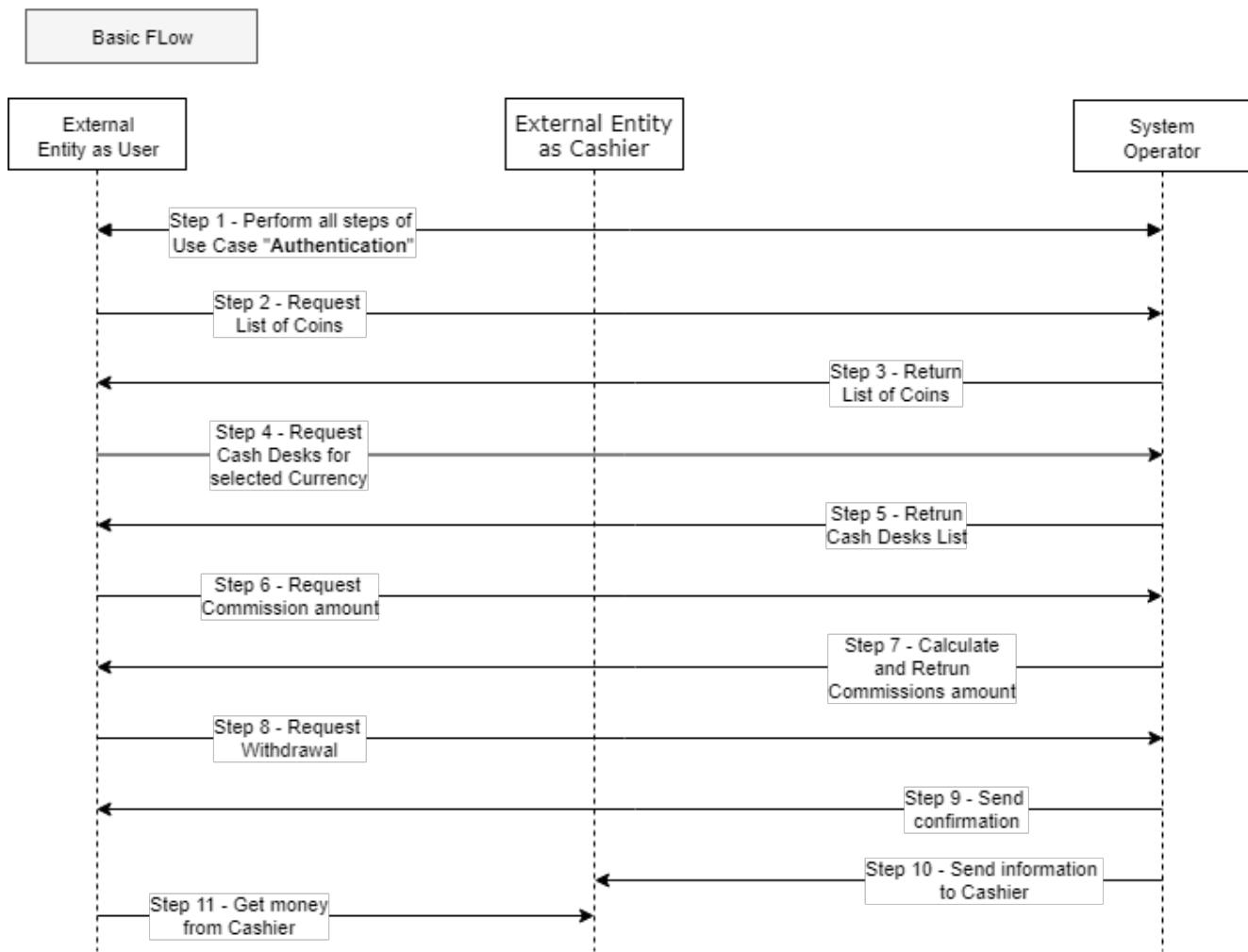
```
{
```

```
"process": {
    "id": "string",
    "createdAt": "2018-07-23T09:40:15.086Z",
    "updatedAt": "2018-07-23T09:40:15.086Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
        {
            "id": "string",
            "parentId": "string",
            "type": "transfer",
            "from": {
                "serial": "string",
                "organizationId": "string",
                "organizationName": "string",
                "technical": false,
                "type": "regular_commission",
                "issuer": {
                    "id": "string",
                    "sn": "string",
                    "currency": "string"
                }
            },
            "to": {
                "serial": "string",
                "organizationId": "string",
                "organizationName": "string",
                "technical": false,
                "type": "regular_commission",
                "issuer": {
                    "id": "string",
                    "sn": "string",
                    "currency": "string"
                }
            },
            "amount": 0,
            "performedAt": "2018-07-23T09:40:15.086Z",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        }
    ],
    "children": [
        {
        }
    ]
}
```

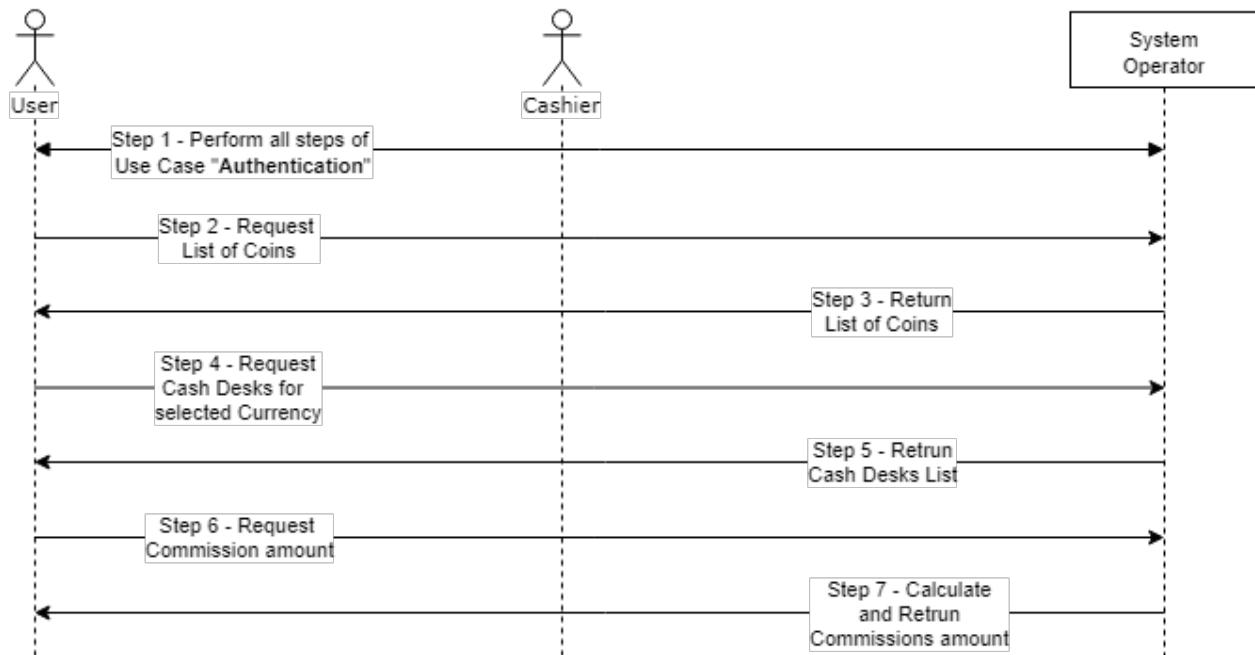
```
        ] ,  
        "errorMessage": "string"  
    } ,  
    "status": "ok" ,  
    "message": "string"  
}
```

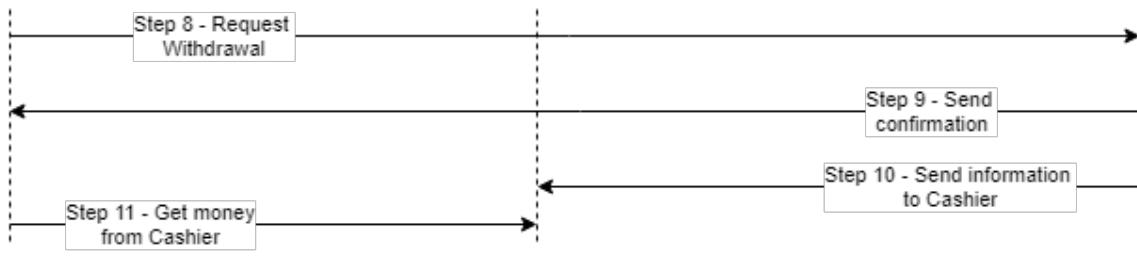
Execute withdrawal via specified cash desk scheme

Use case: Execute withdrawal via specified cash desk



Optional Web UI Flow





Withdraw via cash desk description

Use Case Name

Withdrawal Via Cash Desk

Brief Description

Use Case involves several actors, goes through multiple steps interacting with System Operator API and finally allows the designated person to receive hard cash.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator user.
2. An individual who will get the hard cash - Cash Receiver.
3. System Operator's Accountant.
4. System Operator's Cashier.

Preconditions

There is a valid registered System Operator User in System Operator Data Repository.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication" - done by Coin owner, Accountant, and Cashier.
2. External Entity requests a list of User Coins.

Endpoint URL: GET /coins

Parameters: TOKEN

3. System Operator API returns a List of User Coins.

Response:

```
{
  "coins": [
    {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "availableAmount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "active": false,
      "type": "regular_commission"
    }
  ]
}
```

4. External Entity selects a particular Coin and requests Cash Desks which operate in the Coin Currency - Get cash desks for specified coin.

Endpoint URL: POST /cash-desks/view

Parameters:

```
{
  "coinSerial": "string"
}
```

5. System Operator returns a List of Cash Desks available for the Coin Currency.

6. External Entity requests Withdrawal from a particular Cash Desk - Execute withdrawal via specified cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "receiverName": "string"
}
```

7. System Operator verifies the information, generates and returns “Request Identifier” and other transaction details.

8. External Entity requests Commission Amount calculation - Calculate commission for withdrawal via specified cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "receiverName": "string"
}
```

9. System Operator return commission amount

```
{
  "transactionAmount": 0,
  "senderAmountPush": 0,
  "recipientAmountPush": 0,
  "commissionAmountPush": 0,
  "issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
  }
}
```

10. External Entity sends a Request to Accepts proposed Withdrawal details - Accept withdrawal request.

Endpoint URL: POST /cash-desk-withdrawals/{requestIdentifier}/accept

11. External Entity acting as Accountant sends a Request - View withdrawal request details.

Endpoint URL: GET /cash-desk-withdrawals/details/{requestIdentifier}

12. System Operator returns Withdrawal transaction information.

```
{
  "requestId": "string",
  "amount": 0,
  "currency": {
    "code": "string",
    "digitalCode": "string",
    "symbol": "string",
    "name": "string",
    "description": "string"
  },
  "cashDeskId": "string",
  "processStatus": "string",
  "type": "string",
  "status": "ok",
  "message": "string"
}
```

13. External Entity acting as Accountant sends a Request - Approve withdraw the request

Endpoint URL: POST /cash-desk-withdrawals/{requestIdentifier}/approve

14. Cash receiver comes to the office and provides passport, Withdrawal Identifier, amount, etc.

15. External Entity acting as Cashier sends a Request - View withdrawal requests for specified cash desk.

Endpoint URL: GET /cash-desk-withdrawals/{cashDeskId}

16. System Operator returns a List of unprocessed Withdrawal requests.

17. External Entity acting as Cashier finds Withdrawal by Identifier, checks the Receiver name and hands the Hard Cash to the Receiver.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication” - done by Coin owner, Accountant, and Cashier.

2. Profile Owner requests a list of Coins.

Endpoint URL: GET /coins

Parameters: TOKEN

3. System Operator API returns a List of Profile Coins.

Response:

```
{
  "coins": [
    {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "availableAmount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "active": false,
      "type": "regular_commission"
    }
  ]
}
```

4. Profile Owner selects a particular Coin and requests Cash Desks which operate in the Coin Currency - Get cash desks for specified coin.

Endpoint URL: POST /cash-desks/view

Parameters:

```
{
  "coinSerial": "string"
}
```

5. System Operator returns a List of Cash Desks available

6. Profile Owner requests Withdrawal from a particular desk - Execute withdrawal via specified cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "receiverName": "string"
}
```

7. System Operator verifies the information, generates and returns “Request Identifier” and other transaction details.

8. Profile Owner requests Commission Amount calculation - Calculate commission for withdrawal via specified cash desk.

Endpoint URL: POST /cash-desk-withdrawals/{cashDeskId}/calculate

Parameters:

```
{
  "coin": "string",
  "amount": 0,
  "receiverName": "string"
}
```

9. System Operator return commission amount

```
{
  "transactionAmount": 0,
  "senderAmountPush": 0,
  "recipientAmountPush": 0,
  "commissionAmountPush": 0,
  "issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
  }
}
```

10. Profile Owner sends a Request to Accepts proposed Withdrawal details and provides a person’s name who will get the Money - Accept withdrawal request

Endpoint URL: POST /cash-desk-withdrawals/{requestIdentifier}/accept

11. Accountant sends a Request - View withdrawal request details.

Endpoint URL: GET /cash-desk-withdrawals/details/{requestIdentifier}

12. System Operator returns Withdrawal transaction information

```
{
    "requestId": "string",
    "amount": 0,
    "currency": {
        "code": "string",
        "digitalCode": "string",
        "symbol": "string",
        "name": "string",
        "description": "string"
    },
    "cashDeskId": "string",
    "processStatus": "string",
    "type": "string",
    "status": "ok",
    "message": "string"
}
```

13. Accountant sends a Request - Approve withdraw request

Endpoint URL: POST /cash-desk-withdrawals/{requestIdentifier}/approve

14. Cash Receiver comes to the office and provides: passport, Withdrawal Identifier, amount, etc.

15. Cash Desk Cashier sends a Request - View withdrawal requests for specified cash desk.

Endpoint URL: GET /cash-desk-withdrawals/{cashDeskId}

16. System Operator returns a List of unprocessed Withdrawal requests.

17. Cashier finds Withdrawal by Identifier, checks the Receiver name and hands the Hard Cash to the Receiver.

Alternate Flows

Execute Steps 1 - 12 from Basic FFlow

13. Accountant DECLINES this Withdrawal Request.

Post Conditions

Cash Receiver gets cash according to the original request

Result

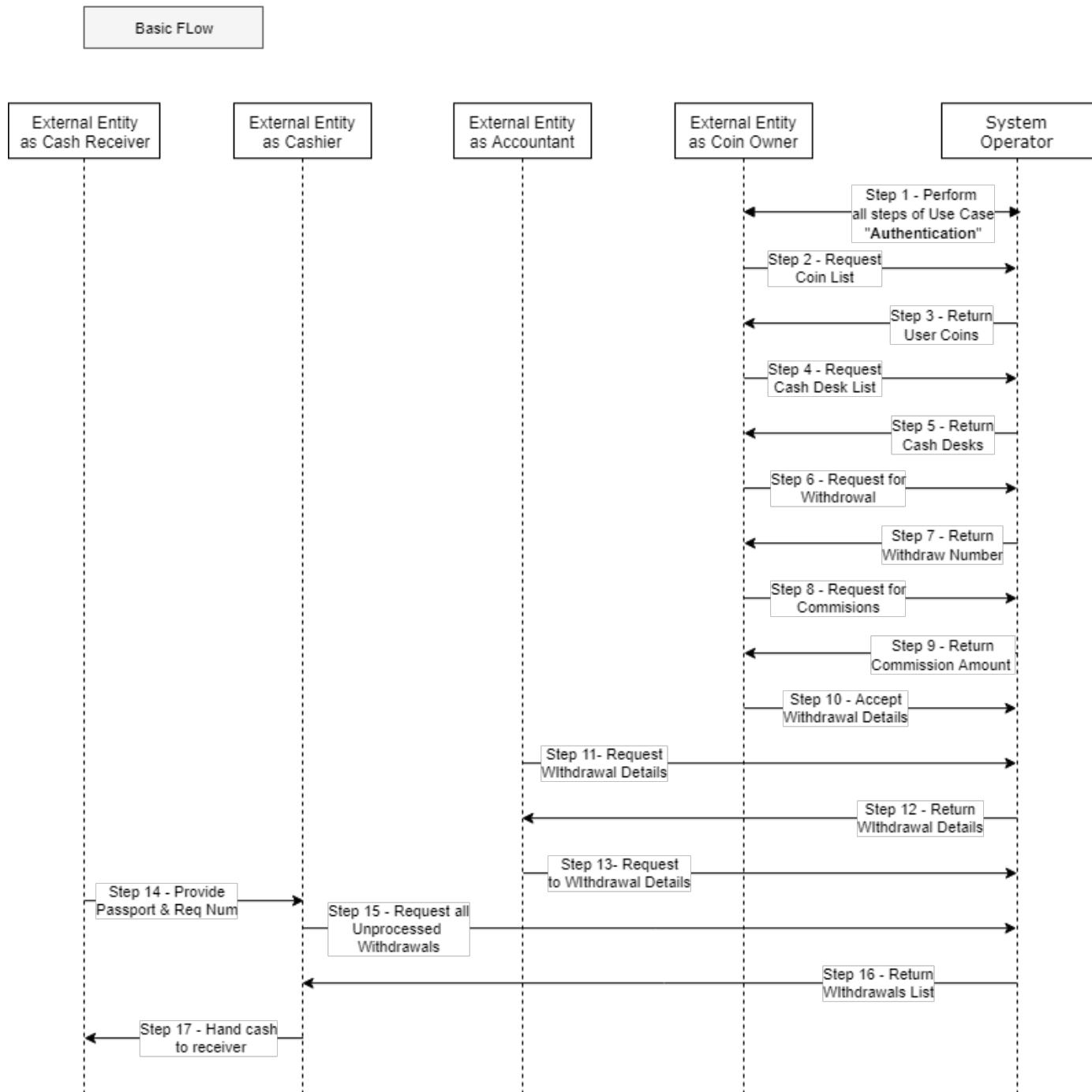
```
{
    "process": {
        "id": "string",
        "createdAt": "2018-06-28T08:20:53.970Z",
        "updatedAt": "2018-06-28T08:20:53.970Z",
        "type": "string",
        "status": "limited",
        "requestIdentifier": 0,
        "requestStatus": "limited",
        "transactions": [
            ...
        ]
    }
}
```

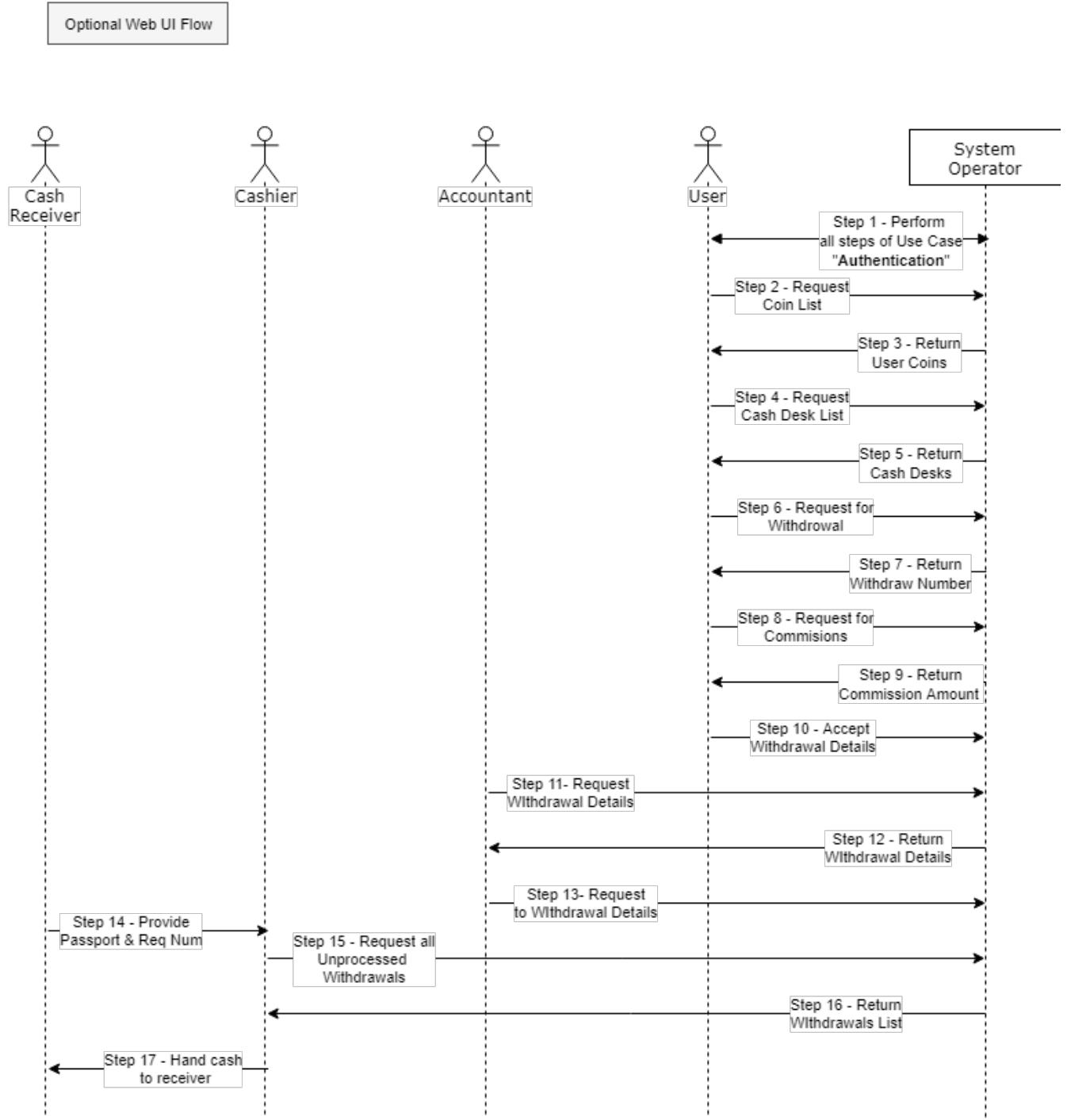
```
{  
    "id": "string",  
    "parentId": "string",  
    "type": "transfer",  
    "from": {  
        "serial": "string",  
        "organizationId": "string",  
        "organizationName": "string",  
        "technical": false,  
        "type": "regular_commission",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        }  
    },  
    "to": {  
        "serial": "string",  
        "organizationId": "string",  
        "organizationName": "string",  
        "technical": false,  
        "type": "regular_commission",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        }  
    },  
    "amount": 0,  
    "performedAt": "2018-06-28T08:20:53.970Z",  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
},  
]  
,"children": [  
    {  
        }  
],  
"errorMessage": "string"  
},
```

```
    "status": "ok",
    "message": "string"
}
```

Withdraw via cash desk scheme

Use case: Withdrawal Via Cash Desk





Transfer

Calculate commission fee for transfer description

Use Case Name

Calculate commission fee

Brief Description

When System Operator Users transfers money to the Coins of another System Operator Coin, System Operator charges some commission for this operations. The amount is calculated based on the parameters set up in the System Operator configurations, actual transfer amount and on the existing contracts with a User.

Actors

1. System Operator using SDK.Finance software.
2. External Entity using System Operator API.

Preconditions

This can only be done by the authenticated entity.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request with all the needed information to calculate the transfer commission.

Endpoint URL: POST /transfers/calculate

Parameters:

```
{
  "senderCoin": "string",
  "recipientCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator validates arrived information, calculates and returns Commission Amount.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. User from the Browser or mobile device sends a request with all needed information to calculate the tarsnfer commission.

Endpoint URL: POST /transfers/calculate

Parameters:

```
{
  "senderCoin": "string",
  "recipientCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator validates arrived information, calculates and returns a web page with Commission Amount.

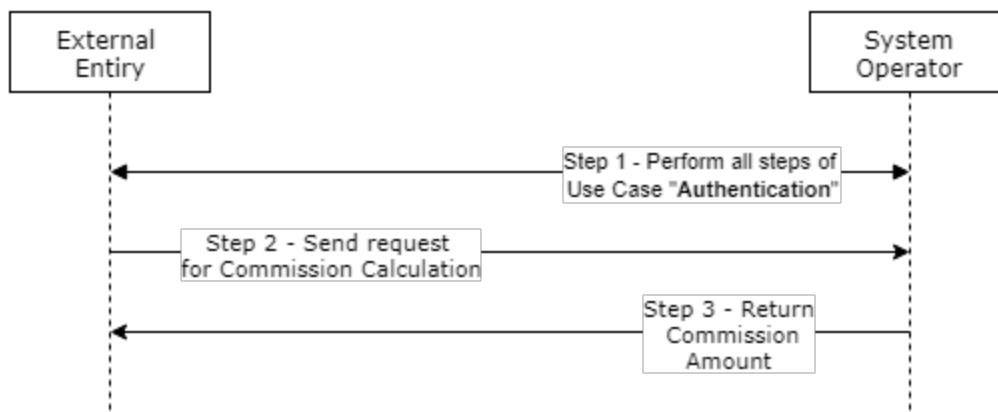
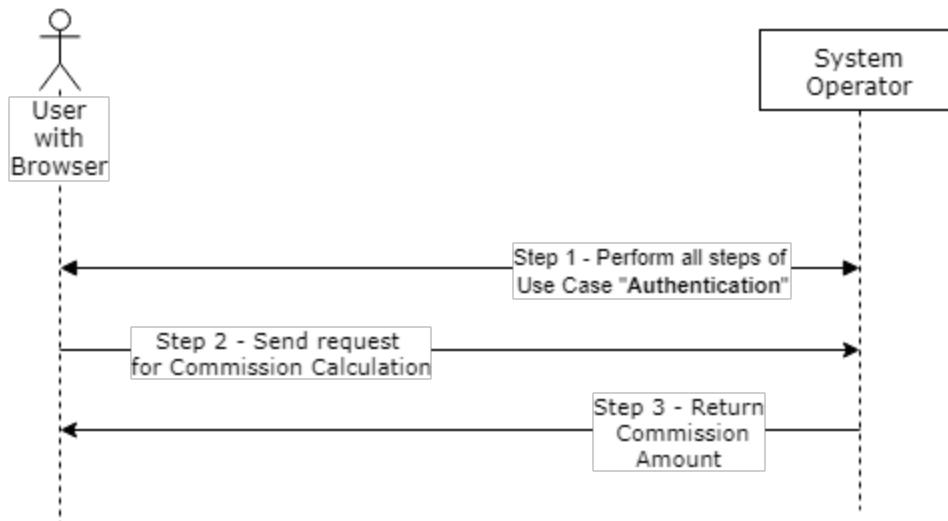
Post Conditions

The Commission Amount calculated exactly as specified in the Contract.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission fee for transfer scheme

Use case: Calculate commission fee**Basic Flow****Optional Web UI Flow**

Execute transfer transaction description

Use Case Name

Execute transfer transaction

Brief Description

An individual passes authentication steps and transfers some amount to another person's Coin.

Both Coins are of the same currency. This transaction is considered by the System Operator to be internal since it does not require any interaction with other institutions.

Actors

1. Individual Coin owner
2. System Operator

Preconditions

1. Source and target Coins must be active, in good standing, and created for the same currency.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Calculate commission fee".

2. External Entity sends a Request to execute a transfer.

Endpoint URL: POST /transfers

Parameters:

```
{
  "senderCoin": "string",
  "recipientCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator verifies that sufficient funds in Source Coin and debits it.

4. System Operator verifies currency type of the Target Coin and credits it.

5. System Operator returns confirmation status to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Calculate commission fee".
2. The user sends a request to get a page to send a transfer request.
3. System Operator returns a page for specifying transfer details.
4. User inputs transfer details and send a transfer request to System Operator.

Endpoint URL: POST /transfers

Parameters:

```
{
    "senderCoin": "string",
    "recipientCoin": "string",
    "amount": 0,
    "description": "string"
}
```

5. System Operator verifies that sufficient funds in Source Coin and debits it.
6. System Operator verifies currency type of the Target Coin and credits it.
7. System Operator returns confirmation page with transfer status to External Entity.

Post Conditions

1. The owner of the Source Coin gets OK or FAILED notification
2. The owner of the Source Coin sees the correct updated balance
3. The owner of the Target Coin sees correct updated balance.

Result example

```
{
  "process": {
    "id": "string",
    "createdAt": "2018-06-28T08:20:56.086Z",
    "updatedAt": "2018-06-28T08:20:56.086Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
      {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
          "serial": "string",
          "organizationId": "string",
          "organizationName": "string",
          "technical": false,
          "type": "regular_commission",
          "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
          }
        },
        "to": {
          "serial": "string",
          "organizationId": "string",
          "technical": false
        }
      }
    ]
  }
}
```

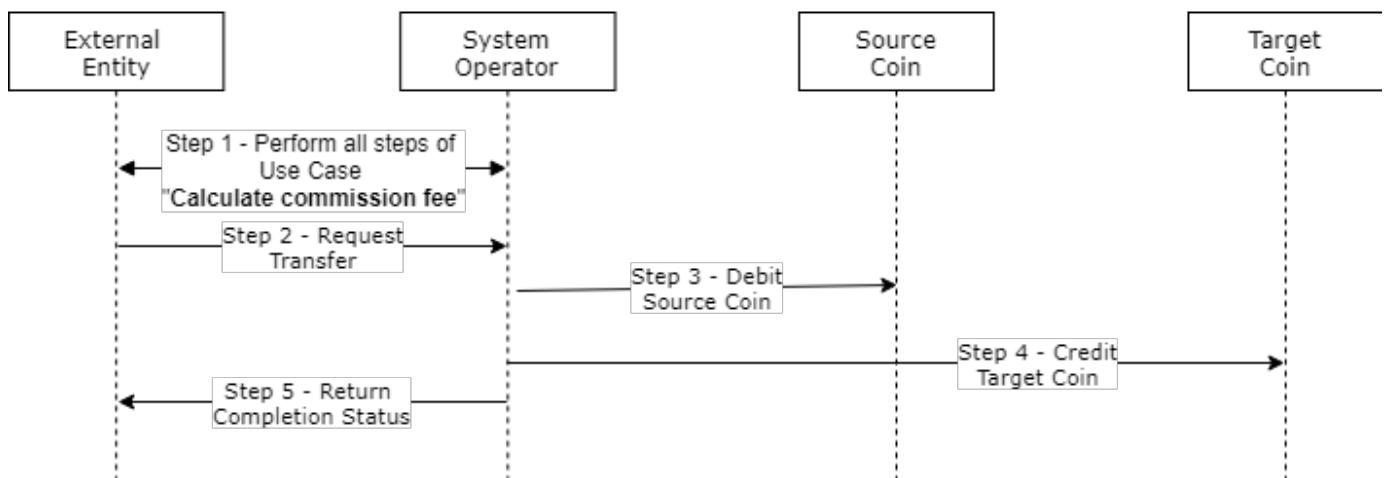
```
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "amount": 0,
    "performedAt": "2018-06-28T08:20:56.086Z",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
},
"children": [
    {
        ...
    },
    ...
],
"errorMessage": "string"
},
```

```
"status": "ok",
"message": "string"
}
```

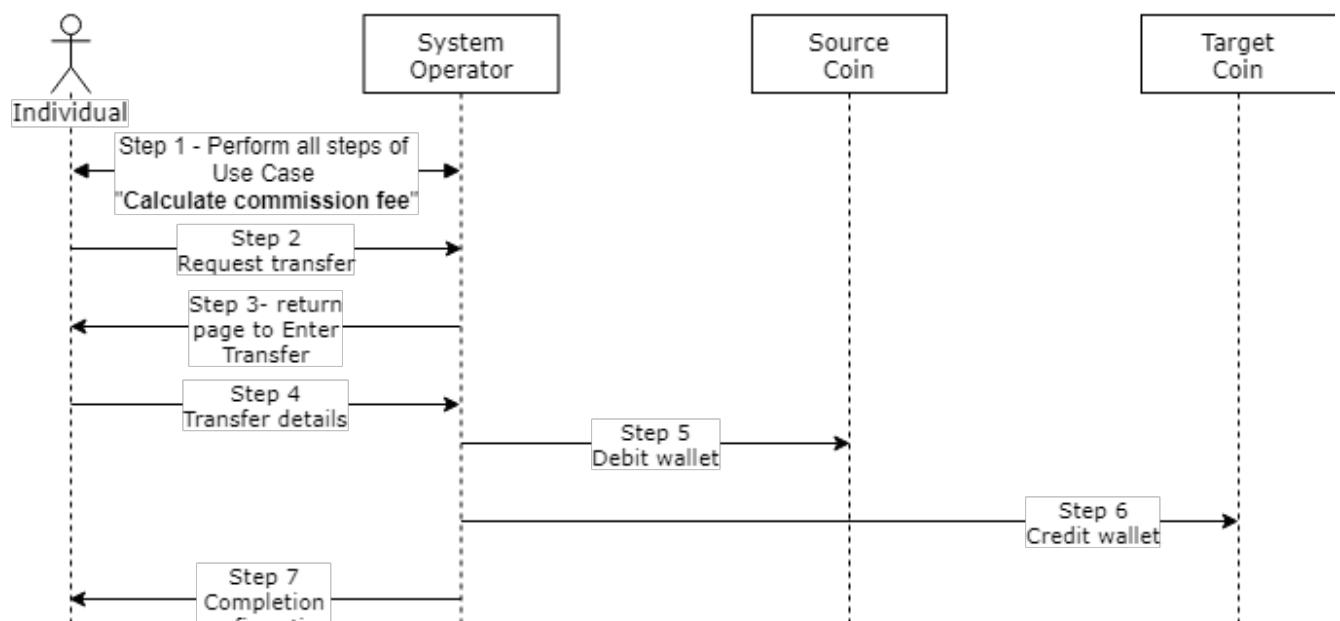
Execute transfer transaction scheme

Use case: Execute transfer transaction

Basic FLow



Optional Web UI Flow



;

confirmation

;

;

;

Transactions

Get user transactions description

Use Case Name

Get user transactions

Brief Description

An individual or External Entity passes authentication steps and requests to see some transactions.

System Operator returns a list of transactions which satisfies provided selection criteria.

Actors

1. External Entity or User
2. System Operator

Preconditions

Coin Owner is valid and in good standing.

Basic Flow - by default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends Request to get Transaction History

Endpoint URL: POST /transactions/view

Selection Parameters:

```
{
  "filter": {
    "ids": [
      "string"
    ],
    "types": [
      "string"
    ],
    "statuses": [
      "limited"
    ],
    "dateFrom": "2018-06-28T08:20:53.821Z",
    "dateTo": "2018-06-28T08:20:53.821Z",
    "coinSerials": [
      "string"
    ],
    "orgIds": [
      "string"
    ],
    "issuerIds": [
      "string"
    ],
    "currencyCodes": [
      "string"
    ],
    "requestIdentifiers": [
      0
    ]
  },
  "sort": {
    "date": "asc",
    "status": "asc",
    "type": "asc"
  }
}
```

3. System Operator performs selection and returns a List to External Entity.

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. The user selects a Coin Owner, defines filtering parameters and sends Request to System Operator to select User transactions.

Endpoint URL: POST /transactions/view

Selection Parameters:

```
{
  "filter": {
    "ids": [
      "string"
    ],
    "types": [
      "string"
    ],
    "statuses": [
      "limited"
    ],
    "dateFrom": "2018-06-28T08:20:53.821Z",
    "dateTo": "2018-06-28T08:20:53.821Z",
    "coinSerials": [
      "string"
    ],
    "orgIds": [
      "string"
    ],
    "issuerIds": [
      "string"
    ],
    "currencyCodes": [
      "string"
    ],
    "requestIdentifiers": [
      0
    ]
  },
  "sort": {
    "date": "asc",
    "status": "asc",
    "type": "asc"
  }
}
```

3. System Operator performs selection and returns a web page with a List.

Post Conditions

Some transaction or a message “no transactions found for your selection criteria”.

Result example

```
{
  "records": [
    {
      "id": "string",
      "label": "string"
    }
  ]
}
```

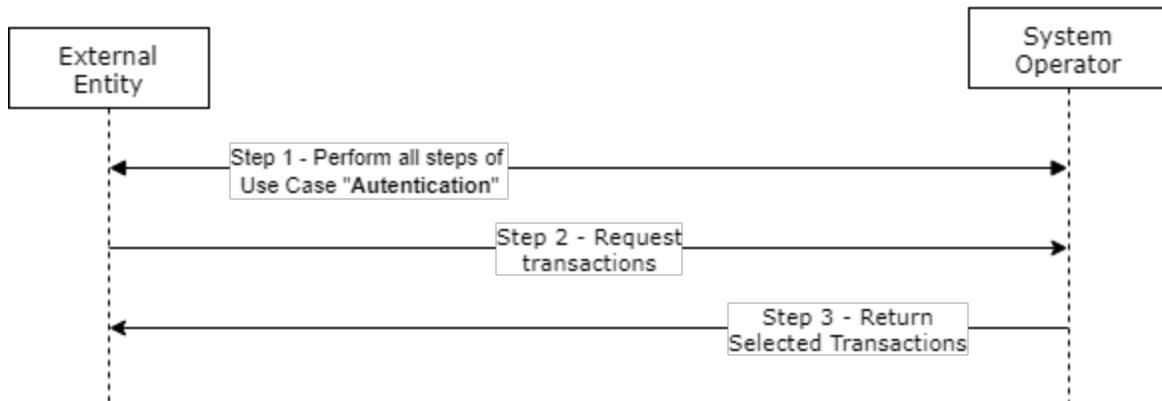
```
"createdAt": "2018-06-06T11:25:04.911Z",
"updatedAt": "2018-06-06T11:25:04.911Z",
"type": "string",
"status": "limited",
"requestIdentifier": 0,
"requestStatus": "limited",
"transactions": [
    {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        },
        "to": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        },
        "amount": 0,
        "performedAt": "2018-06-06T11:25:04.911Z",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    }
],
"errorMessage": "string"
```

```
        }  
    ]  
}
```

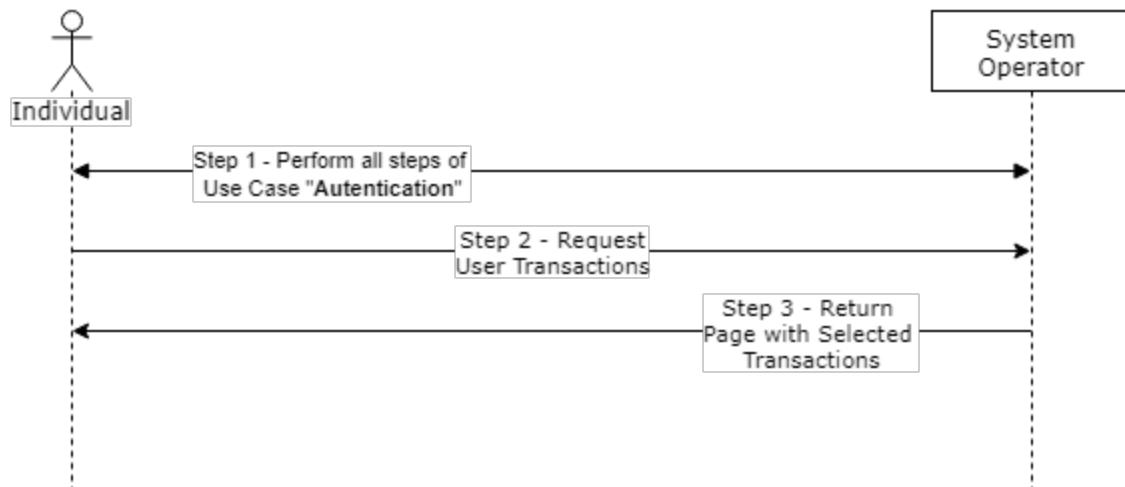
Get user transactions scheme

Use case: Get user transactions

Basic FLow



Optional Web UI Flow



Card management

Delete card from list description

Use Case Name

Delete card from the list

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Get cards" Use Case, and then send a request to Endpoint "Delete card from list". The parameter cardID is obtained from the "Get cards" Use Case. In a user flow, this card ID will be known by the user; a token is still mandatory.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR". Possible roles include an individual, payroll specialist, merchant, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get cards".
2. External Entity sends a request to Endpoint "Delete card from list".

Endpoint URL: DELETE /gate/cards/{cardId}

Parameters: TOKEN - identifies authenticated user

3. System Operator returns the system response message to the External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get cards".
2. A user sends a request to Endpoint "Delete card from list".

Endpoint URL: DELETE /gate/cards/{cardId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns the system response message to User (See Result example below).

Post Conditions

The card is available.

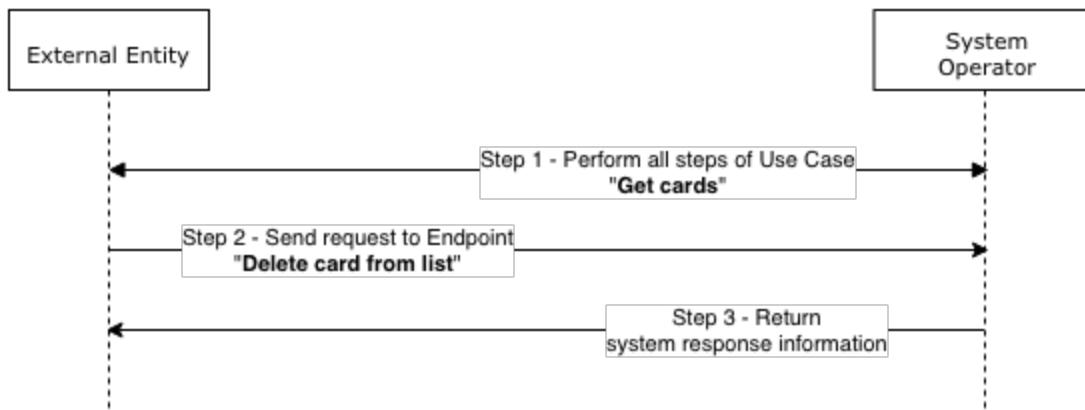
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

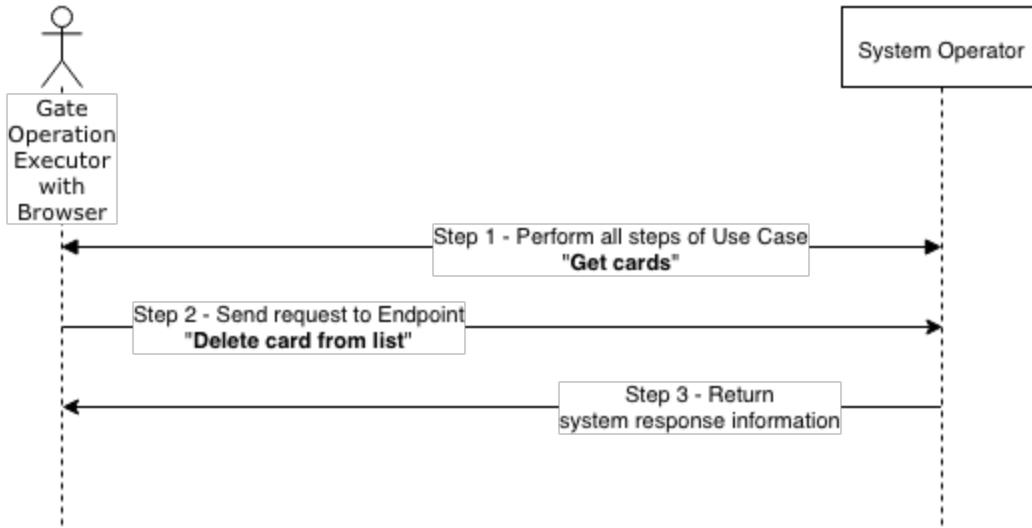
Delete card from list scheme

Use case: Delete card from list

Basic FFlow



Optional Web UI Flow



Card operations

Return funds for a completed card payment description

Use Case Name

Return funds for a completed card payment

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Filter transactions" Use Case (to obtain transaction ID), and then send a request to Endpoint "Return funds for a completed card payment".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Filter transactions".
2. External Entity sends a request to Endpoint "Return funds for a completed card payment".

Endpoint URL: POST /gate/transactions/{tx}/void

Parameters: TOKEN - identifies authenticated user

3. System Operator returns transaction details and confirmation of returned funds to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Filter transactions".
2. A user sends a request to Endpoint "Return funds for a completed card payment".

Endpoint URL: POST /gate/transactions/{tx}/void

Parameters: TOKEN - identifies authenticated user

3. System Operator returns transaction details and confirmation of returned funds to User (See Result example below).

Post Conditions

Transaction is available.

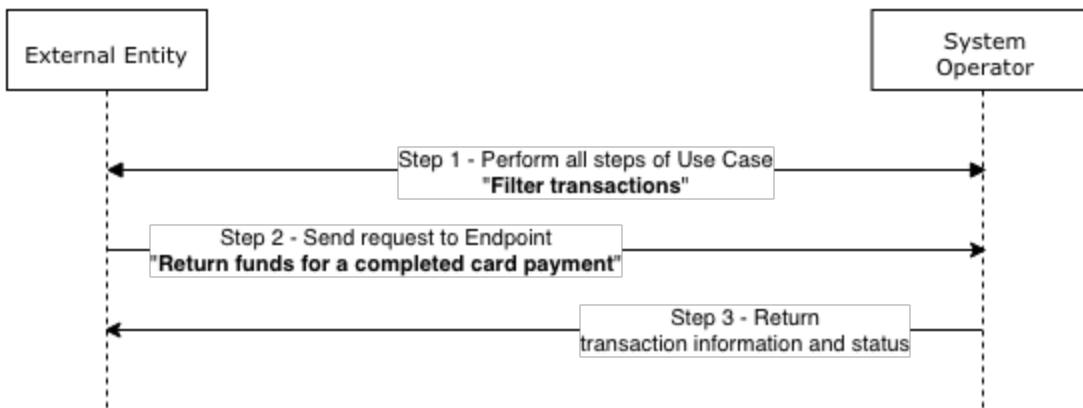
Result example

```
{  
    "transaction": {  
        "id": "string",  
        "orderId": 0,  
        "deviceId": "string",  
        "deviceOrderId": "string",  
        "type": "TOPUP",  
        "status": "INITIATED",  
        "errorCode": "UNKNOWN",  
        "coin": {  
            "serial": "string",  
            "name": "string",  
            "amount": 0,  
            "availableAmount": 0,  
            "issuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            },  
            "active": false,  
            "type": "regular_commission"  
        },  
        "paymentMethod": {  
            "accountId": "string",  
            "account": {  
                "id": "string",  
                "provider": {  
                    "name": "string"  
                },  
            },  
            "way": "string"  
        },  
        "sourceAmount": 0,  
        "amountToSend": 0,  
        "finalAmount": 0,  
        "processId": "string",  
        "payerData": {  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

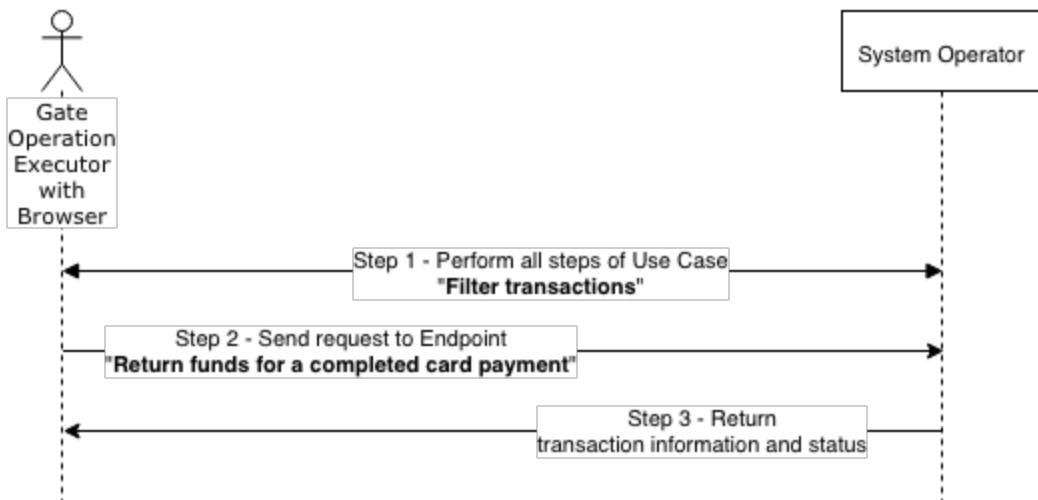
Return funds for a completed card payment scheme

Use case: Return funds for a completed card payment

Basic FLow



Optional Web UI Flow



Hold funds on a payment card

_Release card hold description

Use Case Name

Release card hold

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_OWNER" will go through all steps of "View list of card hold items" Use Case (to obtain card hold item ID), and then send a request to Endpoint "Release card hold".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_OWNER", e.g. a merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View list of card hold items".
2. External Entity sends a request to Endpoint "Release card hold".

Endpoint URL: POST /invoice-card-holds/{id}/release

Parameters: TOKEN - identifies authenticated user

1. System Operator returns updated card information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case View list of card hold items".
2. A user sends a request to Endpoint "Release card hold".

Endpoint URL: POST /invoice-card-holds/{id}/release

Parameters: TOKEN - identifies authenticated user

1. System Operator returns updated card information to User (See Result example below).

Post Conditions

Card hold item is available.

Result example

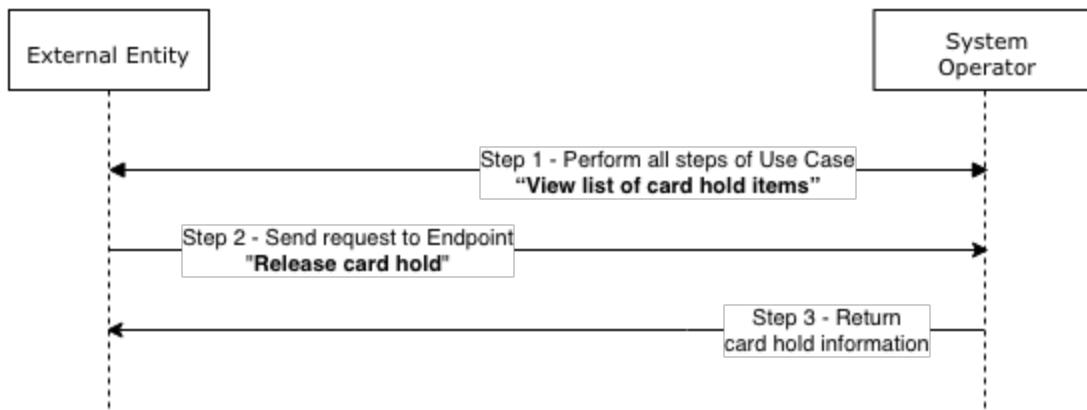
```
{  
  "status": "ok",  
  "message": "string",  
  "cardHold": {  
    "id": "string",  
    "date": "2018-08-31T14:37:03.254Z",  
    "status": "INITIAL",  
    "invoice": {  
      "identifier": "string",  
      "name": "string",  
      "createdAt": "2018-08-31T14:37:03.254Z",  
      "createdByUser": {  
        "id": "string",  
        "name": "string"  
      },  
      "status": "initiated",  
      "payer": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
      },  
      "payerContact": "string",  
      "merchantName": "string",  
      "totalPrice": 0,  
      "expiresAt": "2018-08-31T14:37:03.254Z",  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "data": {  
        "productCode": "string",  
        "productPrice": 0,  
        "description": "string",  
        "count": 0,  
        "terms": "string"  
      },  
      "paymentCode": "string"  
    },  
  },  
}
```

```
"cardTx": {  
    "id": "string",  
    "phase": "INIT",  
    "createdAt": "2018-08-31T14:37:03.254Z",  
    "paymentCardId": "string"  
},  
"heldAmount": 0  
}  
}
```

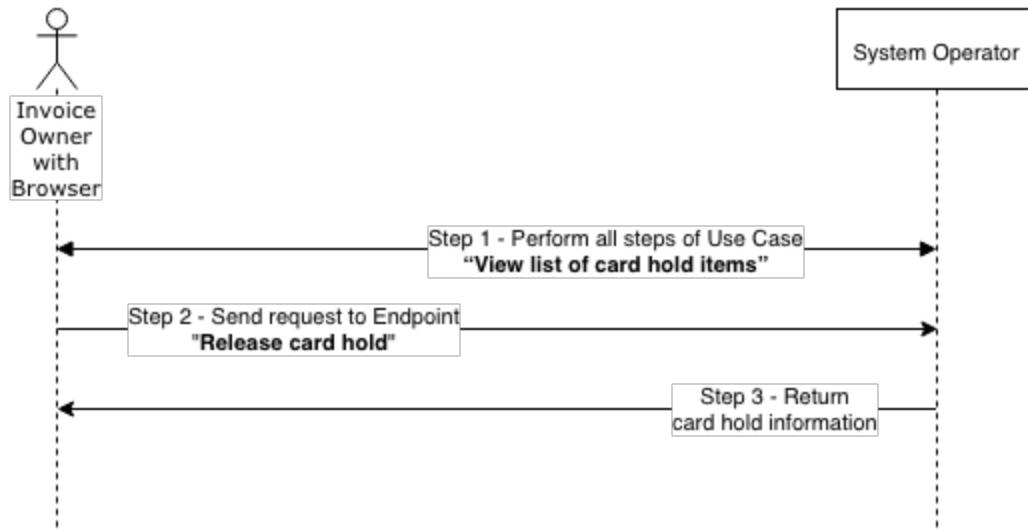
_Release card hold scheme

Use case: Release card hold

Basic FFlow



Optional Web UI Flow



Add invoice hold payment settings description

Use Case Name

Add invoice hold payment settings

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_OWNER" will go through all steps of "View invoices" Use Case, and then send a request to Endpoint "Add invoice hold payment settings". A merchant invokes this Use Case to change the payment settings for an invoice. The invoice identifier is retrieved using the "View invoices" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_OWNER", e.g. a merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View invoices".
2. External Entity sends a request to Endpoint "Add invoice hold payment settings".

Endpoint URL: POST /invoice-card-holds/invoice/{invoice}/hold-settings

Parameters: TOKEN - identifies authenticated user

1. System Operator returns hold payment settings to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View invoices".
2. A user sends a request to Endpoint "Add invoice hold payment settings".

Endpoint URL: POST /invoice-card-holds/invoice/{invoice}/hold-settings

Parameters: TOKEN - identifies authenticated user

1. System Operator returns hold payment settings to User (See Result example below).

Post Conditions

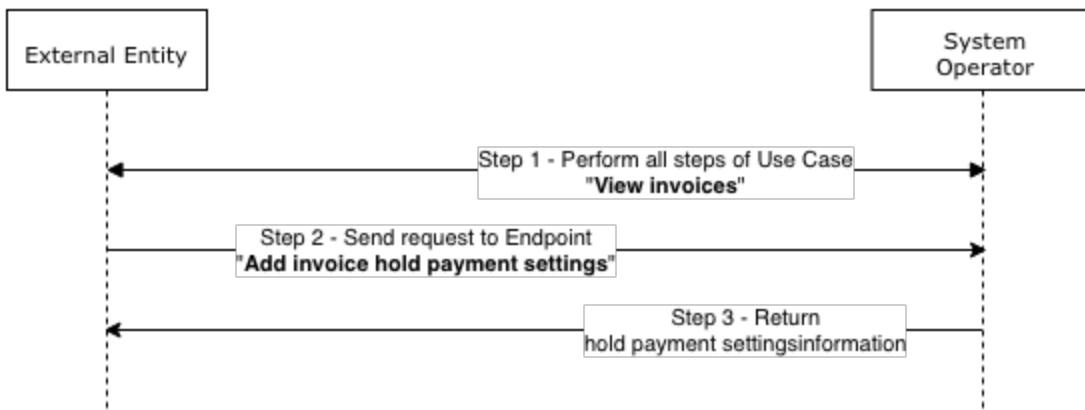
Invoice is available.

Result example

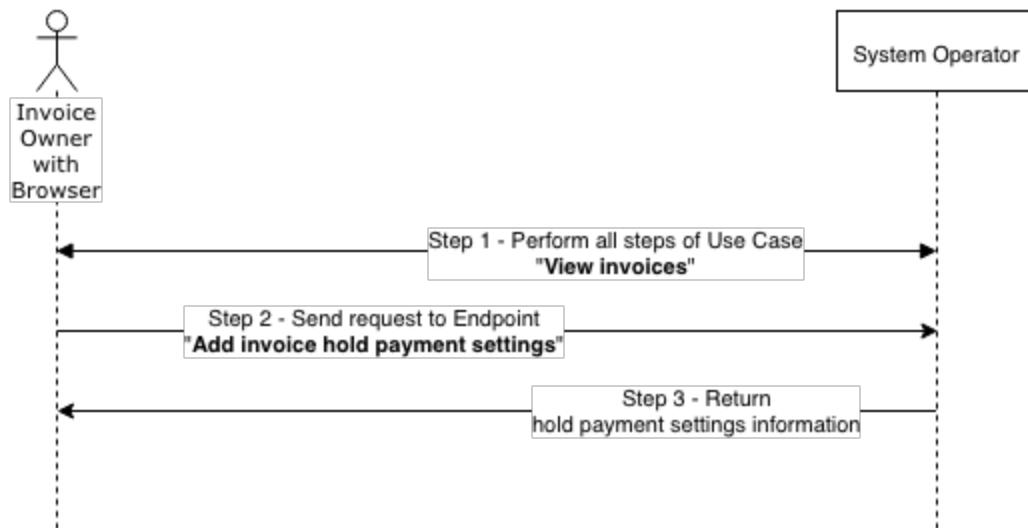
```
{  
  "holdPaymentSettings": {  
    "invoiceSerial": "string",  
    "paymentSettings": "CARD_HOLD"  
  },  
  "status": "ok",  
  "message": "string"  
}  
Add invoice hold payment settings scheme
```

Use case: Add invoice hold payment settings

Basic FFlow



Optional Web UI Flow



Capture card hold description

Use Case Name**Capture card hold****Brief Description**

A User or External Entity on behalf of a User with role permission "INVOICE_OWNER" will go through all steps of "View invoices" Use Case (in which invoiceID is obtained), and then send a request to Endpoint "Capture card hold".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_OWNER", e.g. a merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View invoices".
2. External Entity sends a request to Endpoint "Capture card hold".

Endpoint URL: POST /invoice-card-holds/{id}/capture

Parameters: TOKEN - identifies authenticated user

1. System Operator returns card information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View invoices".
2. A user sends a request to Endpoint "Capture card hold".

Endpoint URL: POST /invoice-card-holds/{id}/capture

Parameters: TOKEN - identifies authenticated user

1. System Operator returns card information to User (See Result example below).

Post Conditions

Invoice is available.

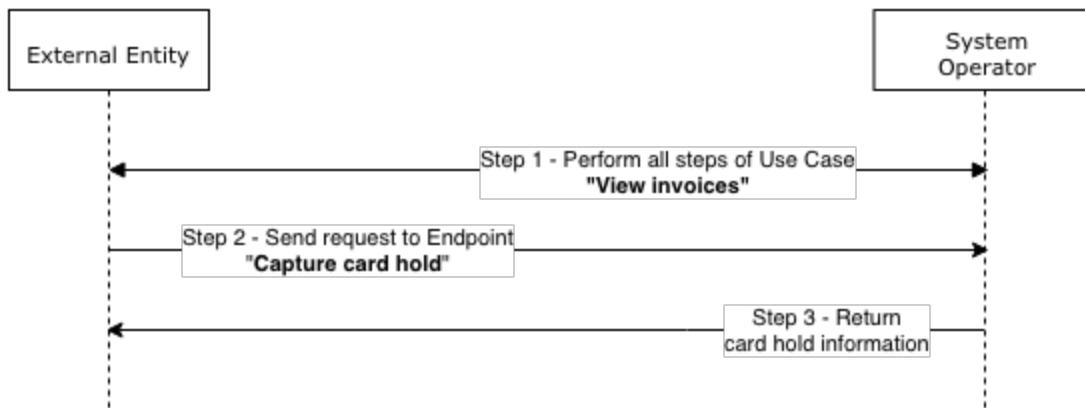
Result example

```
{  
  "status": "ok",  
  "message": "string",  
  "cardHold": {  
    "id": "string",  
    "date": "2018-08-31T14:37:03.235Z",  
    "status": "INITIAL",  
    "invoice": {  
      "identifier": "string",  
      "name": "string",  
      "createdAt": "2018-08-31T14:37:03.235Z",  
      "createdByUser": {  
        "id": "string",  
        "name": "string"  
      },  
      "status": "initiated",  
      "payer": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
      },  
      "payerContact": "string",  
      "merchantName": "string",  
      "totalPrice": 0,  
      "expiresAt": "2018-08-31T14:37:03.235Z",  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "data": {  
        "productCode": "string",  
        "productPrice": 0,  
        "description": "string",  
        "count": 0,  
        "terms": "string"  
      },  
      "paymentCode": "string"  
    },  
  },  
}
```

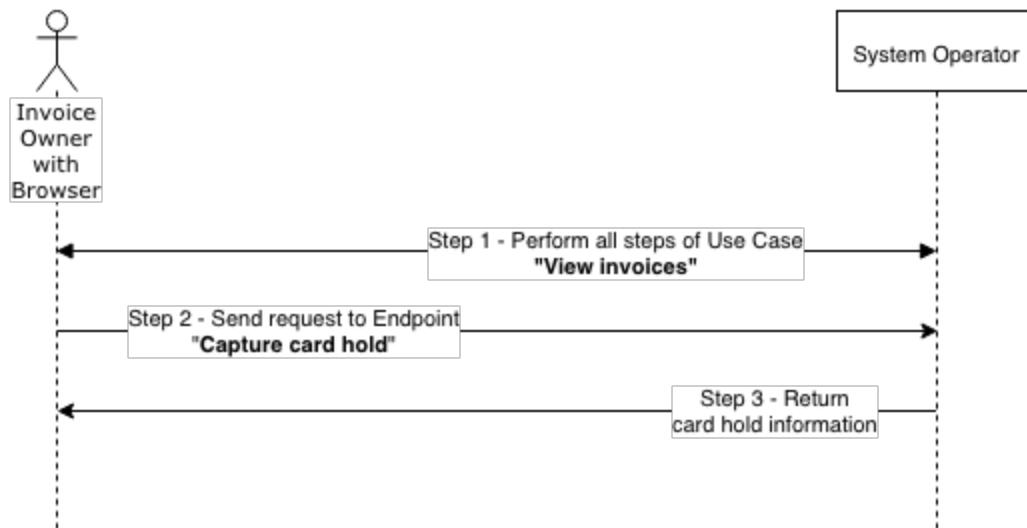
```
"cardTx": {  
    "id": "string",  
    "phase": "INIT",  
    "createdAt": "2018-08-31T14:37:03.235Z",  
    "paymentCardId": "string"  
},  
    "heldAmount": 0  
}  
}  
Capture card hold scheme
```

Use case: Capture card hold

Basic FFlow



Optional Web UI Flow



Create card hold for an invoice description

Use Case Name

Create card hold for an invoice

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_PAYER" will go through all steps of "View invoices" and "Get cards" Use Cases, and then send a request to Endpoint "Create card hold for an invoice".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_PAYER", e.g. a merchant, exchange manager or individual..
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View invoices".
2. Perform all steps of Use Case "Get cards".
3. External Entity sends a request to Endpoint "Create card hold for an invoice".

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice

Parameters: {

```
"invoiceIdentifier": "string",
"paymentCardId": "string",
"providerAccountId": "string"
}
```

1. System Operator returns card hold information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View invoices".
2. Perform all steps of Use Case "Get cards".
3. A user sends a request to Endpoint "Create card hold for an invoice".

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice

Parameters: {

```
"invoiceIdentifier": "string",
"paymentCardId": "string",
"providerAccountId": "string"
```

```
}
```

1. System Operator returns card hold information to User (See Result example below).

Post Conditions

Payment card and invoice are available.

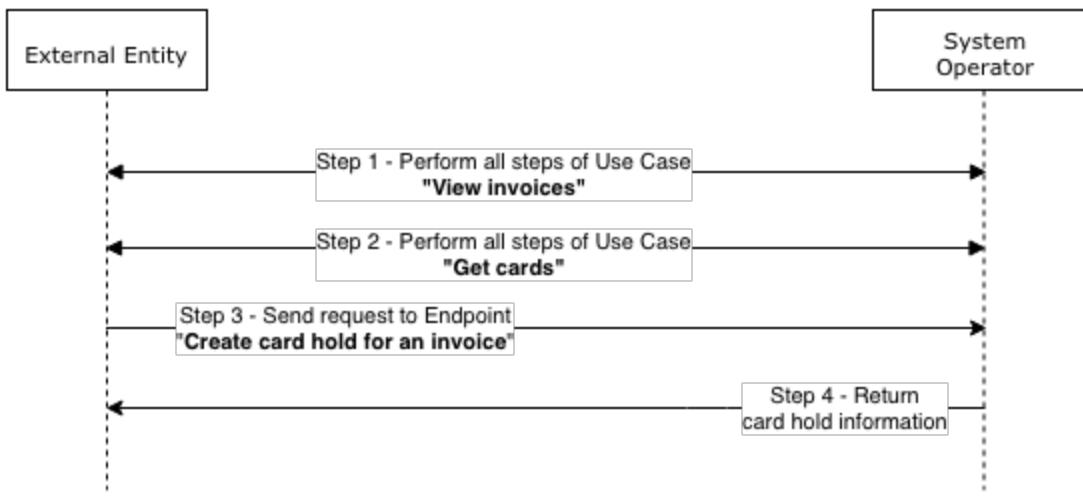
Result example

```
{
  "status": "ok",
  "message": "string",
  "cardHold": {
    "id": "string",
    "date": "2018-08-31T14:37:03.048Z",
    "status": "INITIAL",
    "invoice": {
      "identifier": "string",
      "name": "string",
      "createdAt": "2018-08-31T14:37:03.048Z",
      "createdByUser": {
        "id": "string",
        "name": "string"
      },
      "status": "initiated",
      "payer": {
        "id": "string",
        "type": "string",
        "name": "string"
      },
      "payerContact": "string",
      "merchantName": "string",
      "totalPrice": 0,
      "expiresAt": "2018-08-31T14:37:03.048Z",
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "card": {
        "id": "string",
        "holderName": "string",
        "cardNumber": "string",
        "cvv": "string",
        "expMonth": 12,
        "expYear": 2020,
        "type": "string"
      }
    }
  }
}
```

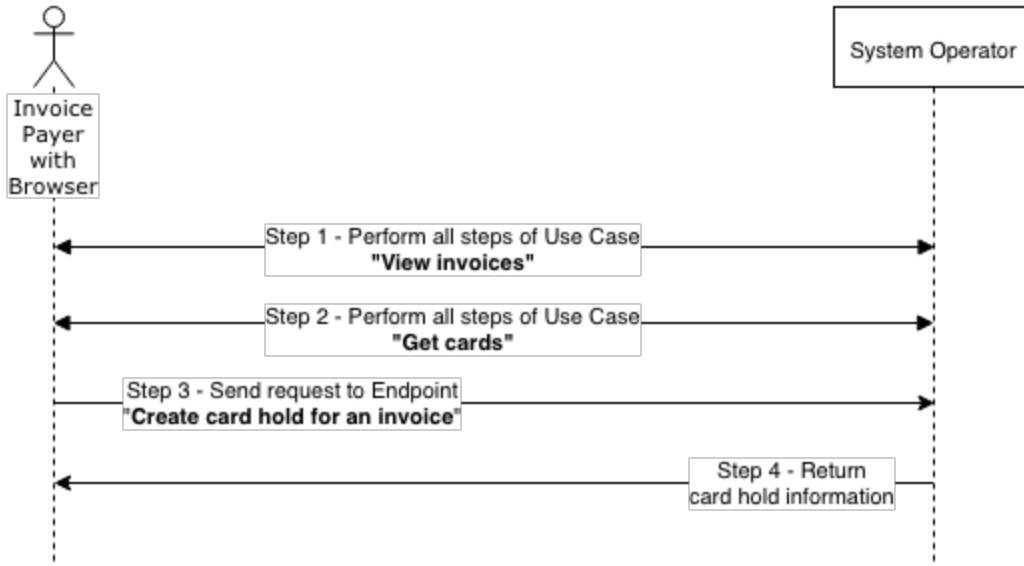
```
"data": {  
    "productCode": "string",  
    "productPrice": 0,  
    "description": "string",  
    "count": 0,  
    "terms": "string"  
},  
    "paymentCode": "string"  
},  
"cardTx": {  
    "id": "string",  
    "phase": "INIT",  
    "createdAt": "2018-08-31T14:37:03.048Z",  
    "paymentCardId": "string"  
},  
    "heldAmount": 0  
}  
}  
}  
Create card hold for an invoice scheme
```

Use case: Create card hold for an invoice

Basic FLow



Optional Web UI Flow



Create card hold for an invoice with new card description

Use Case Name

Create card hold for an invoice with new card

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_PAYER" will go through all steps of "View invoices" Use Case, and then send a request to Endpoint "Create card hold for an invoice with new card". The Invoice Payer will need to have details of the new card (which is not yet in the system) and an invoiceIdentifier from the "View invoices" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_PAYER", e.g. an individual, merchant or exchange manager. .
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View invoices".
2. External Entity sends a request to Endpoint "Create card hold for an invoice with new card".

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice-with-new-card

Parameters: {

```
"invoiceIdentifier": "string",
"paymentCardData": {
  "pan": "string",
  "expirationMonth": 0,
  "expirationYear": 0,
  "cardholderName": "string",
  "cvv": "string"
},
"providerAccountId": "string"
}
```

1. System Operator returns card hold information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.
2. A user sends a request to Endpoint “Create card hold for an invoice with new card”.

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice-with-new-card

```
Parameters: {
  "invoiceIdentifier": "string",
  "paymentCardData": {
    "pan": "string",
    "expirationMonth": 0,
    "expirationYear": 0,
    "cardholderName": "string",
    "cvv": "string"
  },
  "providerAccountId": "string"
}
```

1. System Operator returns card hold information to User (See Result example below).

Post Conditions

Invoice and new card details are available.

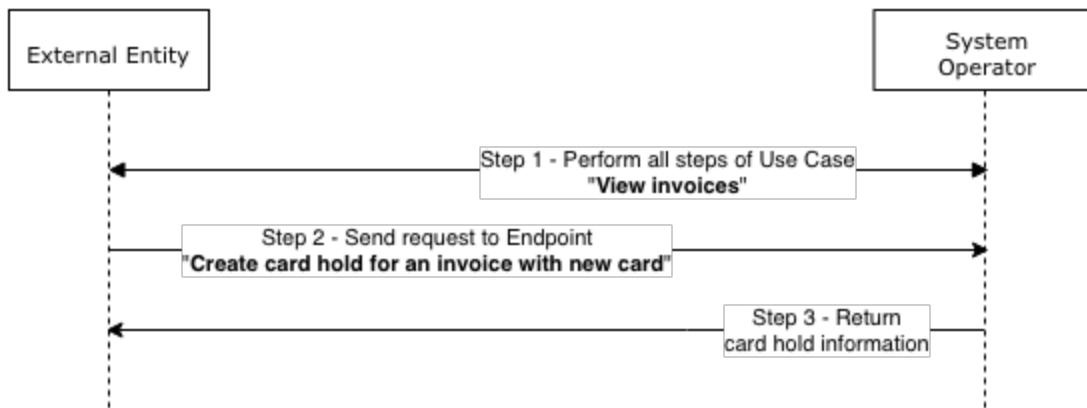
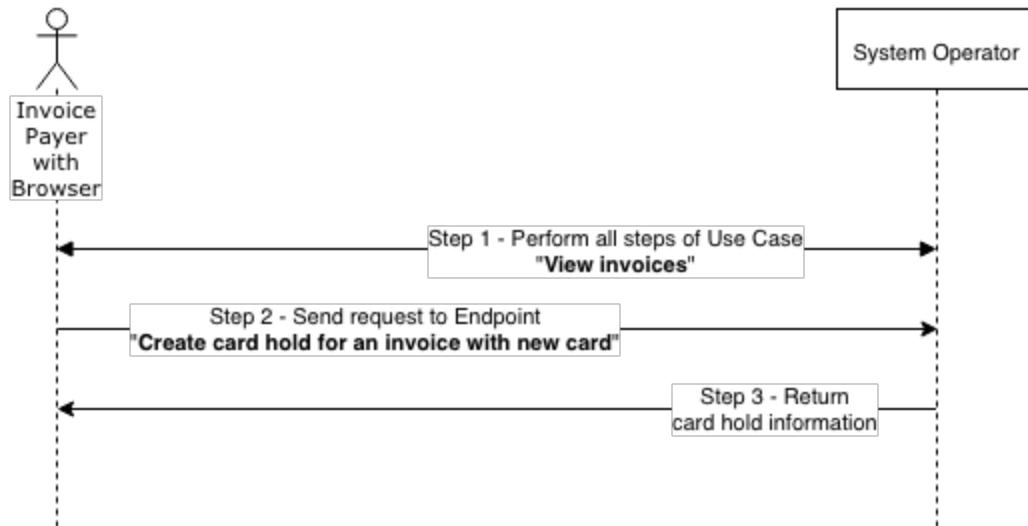
Result example

```
{
  "status": "ok",
  "message": "string",
  "cardHold": {
    "id": "string",
    "date": "2018-08-31T14:37:03.173Z",
    "status": "INITIAL",
    "invoice": {
      "identifier": "string",
      "name": "string",
      "createdAt": "2018-08-31T14:37:03.173Z",
      "createdByUser": {
        "id": "string",
        "name": "string"
      },
      "status": "initiated",
      "payer": {
        "id": "string",
        "name": "string"
      }
    }
  }
}
```

```
"type": "string",
"name": "string"
},
"payerContact": "string",
"merchantName": "string",
"totalPrice": 0,
"expiresAt": "2018-08-31T14:37:03.173Z",
"issuer": {
  "id": "string",
  "sn": "string",
  "currency": "string"
},
"data": {
  "productCode": "string",
  "productPrice": 0,
  "description": "string",
  "count": 0,
  "terms": "string"
},
"paymentCode": "string"
},
"cardTx": {
  "id": "string",
  "phase": "INIT",
  "createdAt": "2018-08-31T14:37:03.174Z",
  "paymentCardId": "string"
},
"heldAmount": 0
}
}
```

Create card hold for an invoice with new card scheme

Use case: Create card hold for an invoice with new card

Basic FFlow**Optional Web UI Flow**

View invoice hold payment settings description

Use Case Name

Create card hold for an invoice with new card

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_PAYER" will go through all steps of "View invoices" Use Case, and then send a request to Endpoint "Create card hold for an invoice with new card". The Invoice Payer will need to have details of the new card (which is not yet in the system) and an invoiceIdentifier from the "View invoices" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_PAYER", e.g. an individual, merchant or exchange manager. .
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View invoices".
2. External Entity sends a request to Endpoint "Create card hold for an invoice with new card".

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice-with-new-card

Parameters: {

```
"invoiceIdentifier": "string",
"paymentCardData": {
  "pan": "string",
  "expirationMonth": 0,
  "expirationYear": 0,
  "cardholderName": "string",
  "cvv": "string"
},
"providerAccountId": "string"
}
```

1. System Operator returns card hold information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.
2. A user sends a request to Endpoint “Create card hold for an invoice with new card”.

Endpoint URL: POST /invoice-card-holds/create-hold-for-invoice-with-new-card

```
Parameters: {
  "invoiceIdentifier": "string",
  "paymentCardData": {
    "pan": "string",
    "expirationMonth": 0,
    "expirationYear": 0,
    "cardholderName": "string",
    "cvv": "string"
  },
  "providerAccountId": "string"
}
```

1. System Operator returns card hold information to User (See Result example below).

Post Conditions

Invoice and new card details are available.

Result example

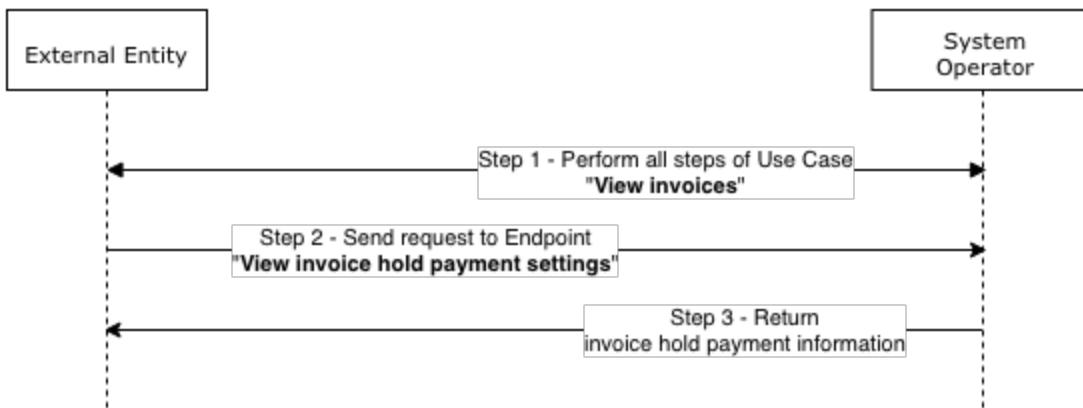
```
{
  "status": "ok",
  "message": "string",
  "cardHold": {
    "id": "string",
    "date": "2018-08-31T14:37:03.173Z",
    "status": "INITIAL",
    "invoice": {
      "identifier": "string",
      "name": "string",
      "createdAt": "2018-08-31T14:37:03.173Z",
      "createdByUser": {
        "id": "string",
        "name": "string"
      },
      "status": "initiated",
      "payer": {
        "id": "string",
        "name": "string"
      }
    }
  }
}
```

```
"type": "string",
"name": "string"
},
"payerContact": "string",
"merchantName": "string",
"totalPrice": 0,
"expiresAt": "2018-08-31T14:37:03.173Z",
"issuer": {
  "id": "string",
  "sn": "string",
  "currency": "string"
},
"data": {
  "productCode": "string",
  "productPrice": 0,
  "description": "string",
  "count": 0,
  "terms": "string"
},
"paymentCode": "string"
},
"cardTx": {
  "id": "string",
  "phase": "INIT",
  "createdAt": "2018-08-31T14:37:03.174Z",
  "paymentCardId": "string"
},
"heldAmount": 0
}
}
```

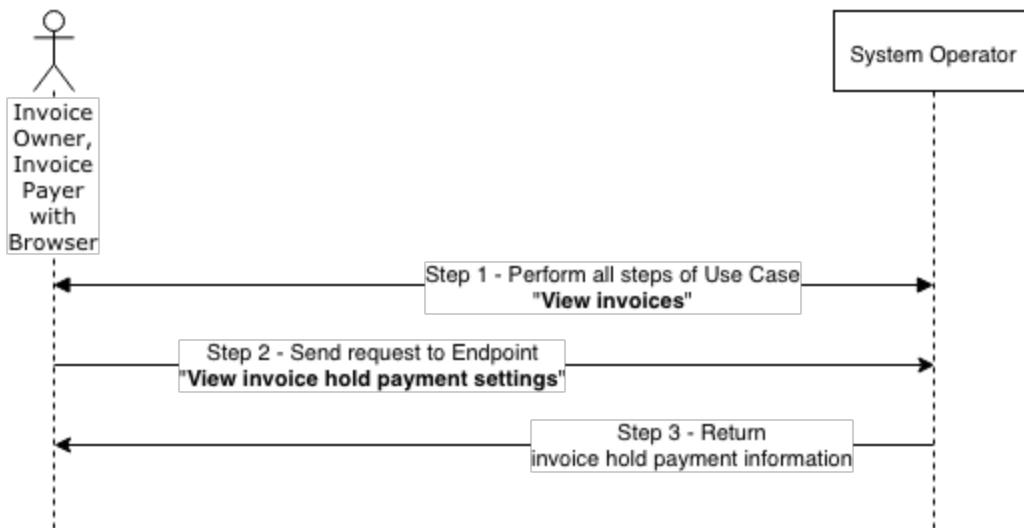
View invoice hold payment settings scheme

Use case: View invoice hold payment settings

Basic FFlow



Optional Web UI Flow



View list of card hold items description

Use Case Name

[View list of card hold items](#)

Brief Description

A User or External Entity on behalf of a User with role permission "INVOICE_OWNER" and "INVOICE_PAYER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View list of card hold items". The results list may be filtered by card hold ID or status. If the filter is omitted, all card hold items are returned.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "INVOICE_OWNER" and "INVOICE_PAYER", e.g. merchant, individual or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View list of card hold items".

Endpoint URL: POST /invoice-card-holds/view

Parameters: {

 "pageNumber": 0,

 "pageSize": 0,

 "filter": {

 "ids": [

 "string"

],

 "statuses": [

 "INITIAL"

]

 },

 "sort": {

 "date": "asc",

 "status": "asc"

 }

}

1. System Operator returns list of card hold items to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View list of card hold items”.

Endpoint URL: POST /invoice-card-holds/view

Parameters: {

 "pageNumber": 0,

 "pageSize": 0,

 "filter": {

 "ids": [

 "string"

],

 "statuses": [

 "INITIAL"

]

 },

 "sort": {

 "date": "asc",

 "status": "asc"

 }

}

1. System Operator returns list of card hold items information to User (See Result example below).

Post Conditions

One or more card hold items are available.

Result example

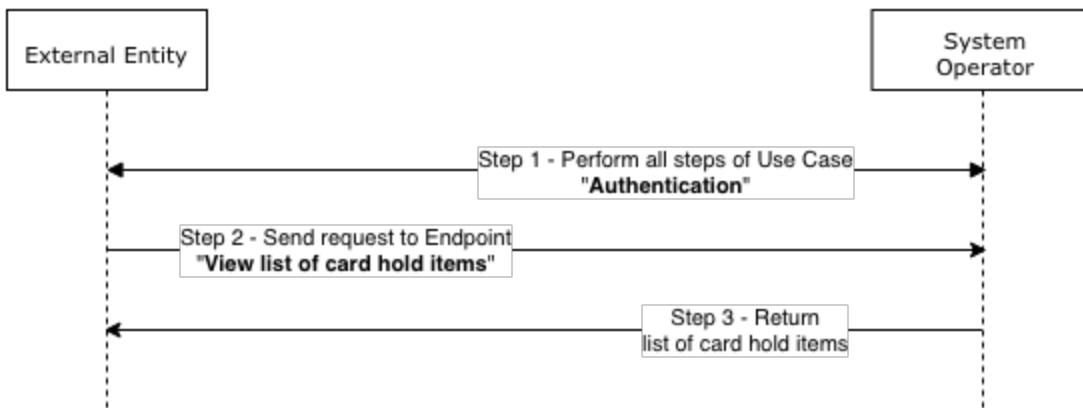
```
{
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "records": [
    {
      "id": 1,
      "cardHoldItem": "string"
    }
  ]
}
```

```
"id": "string",
"date": "2018-08-31T14:37:03.226Z",
"status": "INITIAL",
"invoice": {
  "identifier": "string",
  "name": "string",
  "createdAt": "2018-08-31T14:37:03.226Z",
  "createdByUser": {
    "id": "string",
    "name": "string"
  },
  "status": "initiated",
  "payer": {
    "id": "string",
    "type": "string",
    "name": "string"
  },
  "payerContact": "string",
  "merchantName": "string",
  "totalPrice": 0,
  "expiresAt": "2018-08-31T14:37:03.227Z",
  "issuer": {
    "id": "string",
    "sn": "string",
    "currency": "string"
  },
  "data": {
    "productCode": "string",
    "productPrice": 0,
    "description": "string",
    "count": 0,
    "terms": "string"
  },
  "paymentCode": "string"
},
"cardTx": {
  "id": "string",
  "phase": "INIT",
  "createdAt": "2018-08-31T14:37:03.228Z",
  "paymentCardId": "string"
},
```

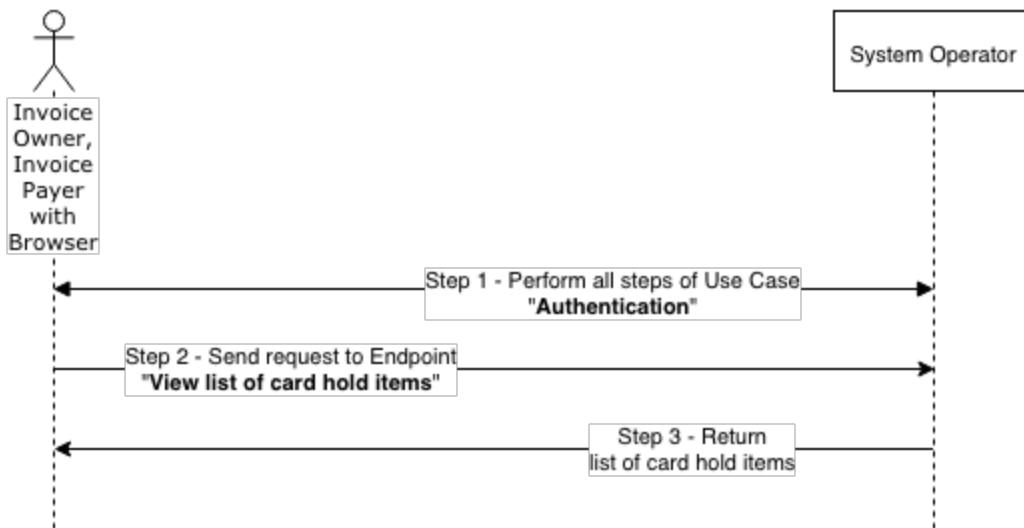
```
        "heldAmount": 0
    }
]
}
View list of card hold items scheme
```

Use case: View list of card hold items

Basic FFlow



Optional Web UI Flow



Smart cards management - as supervisor

Create a new smart card description

Use Case Name

Create a new smart card

Brief Description

A User or External Entity on behalf of a User with role permission "SMART_CARD_ISSUER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Create a new smart card".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SMART_CARD_ISSUER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create a new smart card".

Endpoint URL: POST /smart-cards

```
Parameters: {
  "name": "string",
  "number": "string",
  "organizationId": "string"
}
```

1. System Operator returns new smart card information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Create a new smart card".

Endpoint URL: POST /smart-cards

```
Parameters: {
  "name": "string",
  "number": "string",
  "organizationId": "string"
}
```

```
}
```

1. System Operator returns new smart card information to User (See Result example below).

Post Conditions

Parameter details are available, e.g. name of user and organizationId.

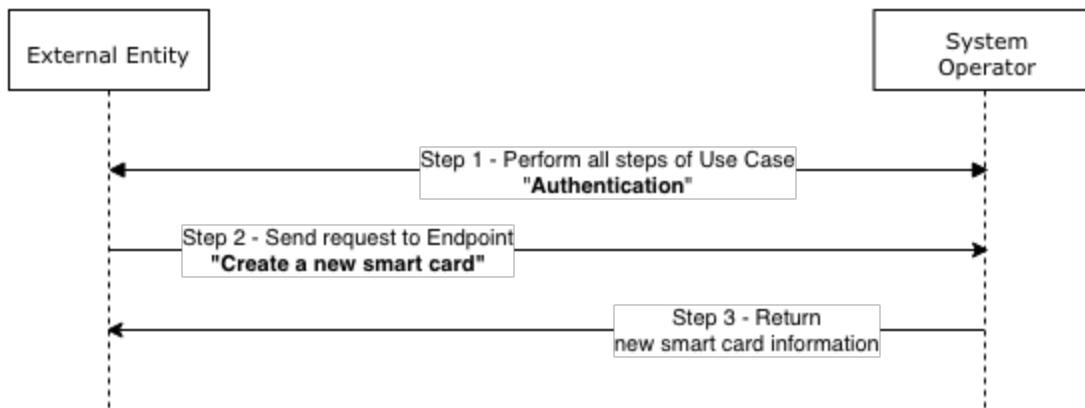
Result example

```
{  
  "smartCard": {  
    "cardNumber": "string",  
    "name": "string",  
    "organizationId": "string",  
    "active": false,  
    "createdAt": "2018-08-22T09:12:26.530Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

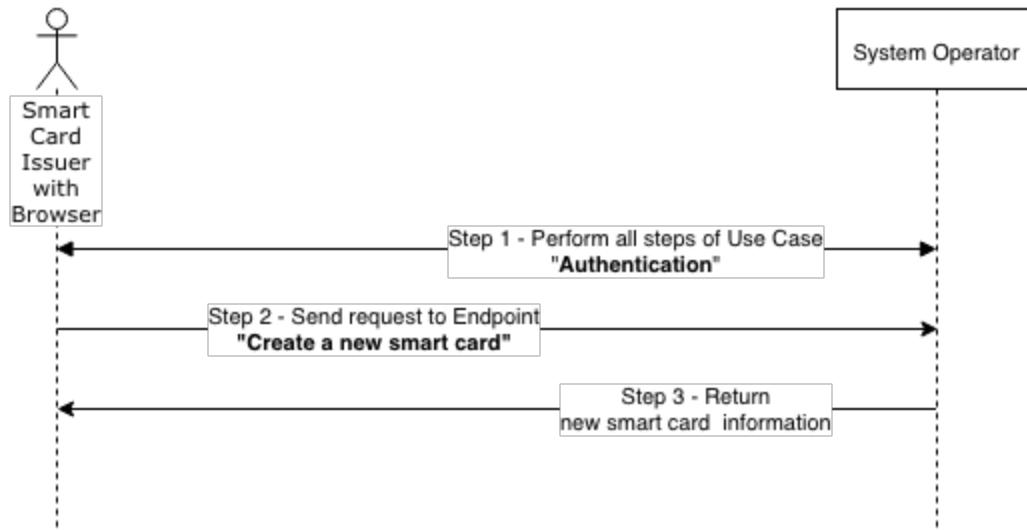
Create a new smart card scheme

Use case: Create a new smart card

Basic FFlow



Optional Web UI Flow



Delete an existing smart card description

Use Case Name

Delete an existing smart card

Brief Description

A User or External Entity on behalf of a User with role permission "SMART_CARD_ISSUER" will go through all steps of "Get smart cards by specified filter" (obtains smart card number) Use Case, and then send a request to Endpoint "Delete an existing smart card".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SMART_CARD_ISSUER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get smart cards by specified filter".
2. External Entity sends a request to Endpoint "Delete an existing smart card".

Endpoint URL: DELETE /smart-cards/{number}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get smart cards by specified filter".
2. A user sends a request to Endpoint "Delete an existing smart card".

Endpoint URL: DELETE /smart-cards/{number}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Smart card is available.

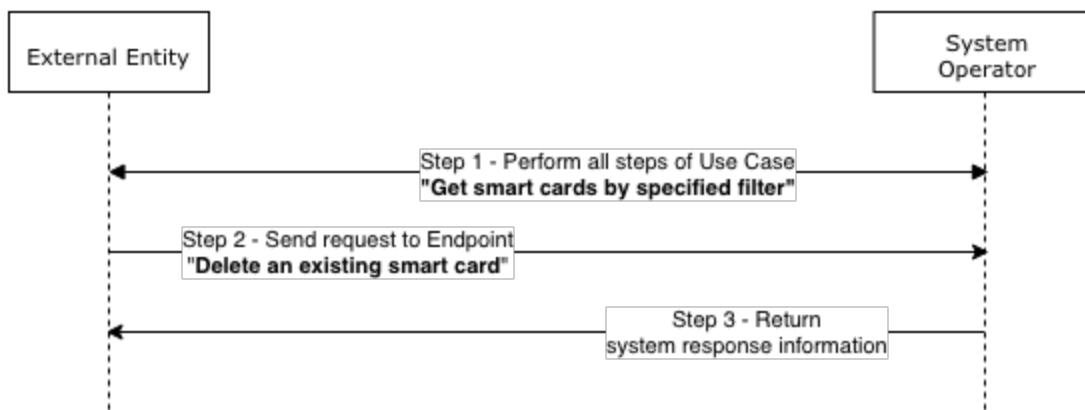
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

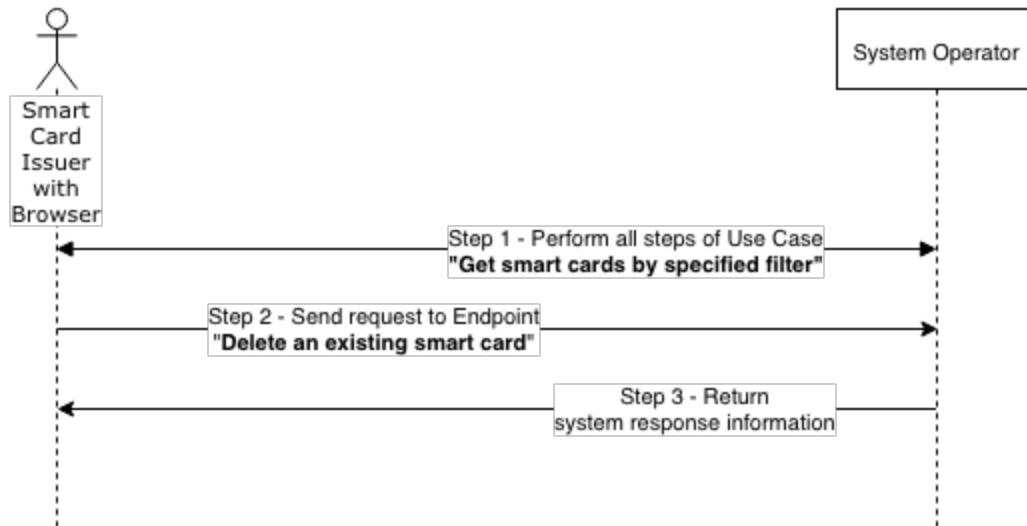
Delete an existing smart card scheme

Use case: Delete an existing smart card

Basic FFlow



Optional Web UI Flow



Get smart cards by specified filter description

Use Case Name

Get smart cards by specified filter

Brief Description

A User or External Entity on behalf of a User with role permission "SMART_CARD_ISSUER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get smart cards by specified filter". When there are no filtering parameter available, the filter could be omitted to get all smart cards in the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SMART_CARD_ISSUER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get smart cards by specified filter".

Endpoint URL: POST /smart-cards/view

Parameters: {

```
"filter": { -optional
  "cardNumber": "string",
  "organizationIds": [
    "string"
  ],
  "active": false
},
"sort": {
  "date": "asc"
},
"pageNumber": 0,
"pageSize": 0
}
```

1. System Operator returns list of smart cards to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get smart cards by specified filter”.

Endpoint URL: POST /smart-cards/view

Parameters: {

```
"filter": {
  "cardNumber": "string",
  "organizationIds": [
    "string"
  ],
  "active": false
},
"sort": {
  "date": "asc"
},
"pageNumber": 0,
"pageSize": 0
}
```

1. System Operator returns list of smart cards to User (See Result example below).

Post Conditions

Smart cards are available.

Result example

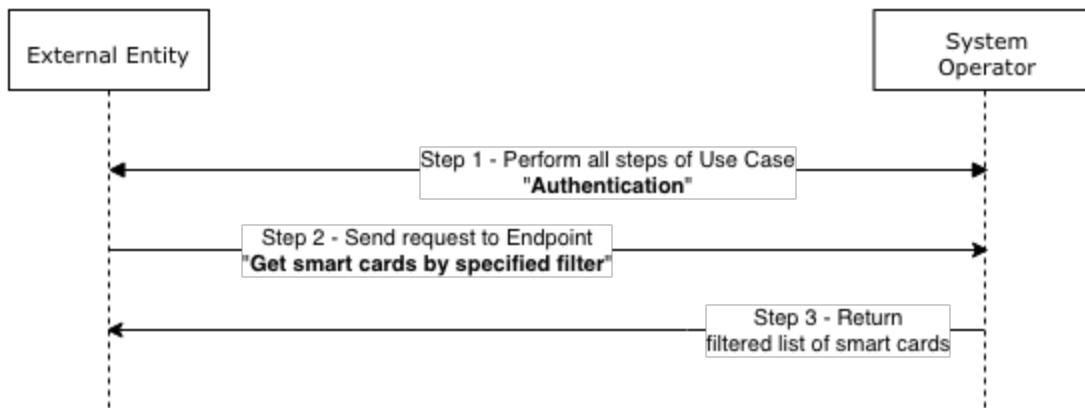
```
{
  "records": [
    {
      "cardNumber": "string",
      "name": "string",
      "organizationId": "string",
      "active": false,
      "createdAt": "2018-08-22T09:12:26.531Z"
    }
  ],
  "status": "ok",
  "message": "string",
}
```

```
"pageNumber": 0,  
"pageSize": 0,  
"totalRecords": 0,  
"totalPages": 0  
}
```

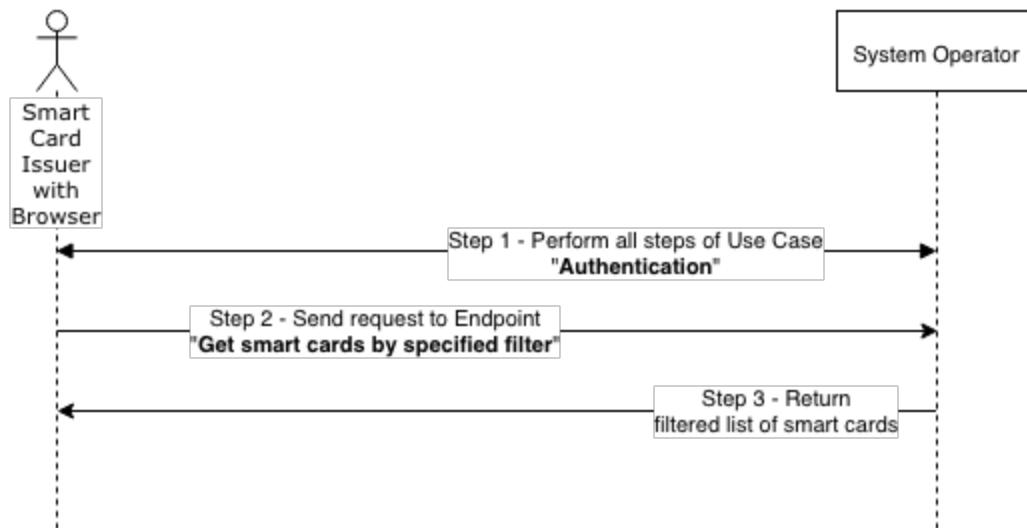
Get smart cards by specified filter scheme

Use case: Get smart cards by specified filter

Basic FFlow



Optional Web UI Flow



UPDATE AN EXISTING SMART CARD description

Use Case Name

UPDATE AN EXISTING SMART CARD

Brief Description

A User or External Entity on behalf of a User with role permission "SMART_CARD_ISSUER" will go through all steps of "GET SMART CARDS BY SPECIFIED FILTER" (to obtain smart card number) Use Case, and then send a request to Endpoint "UPDATE AN EXISTING SMART CARD" to activate or deactivate the card.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SMART_CARD_ISSUER", e.g. administrator, CFO or financial specialist.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "GET SMART CARDS BY SPECIFIED FILTER".
2. External Entity sends a request to Endpoint "UPDATE AN EXISTING SMART CARD".

Endpoint URL: PATCH /smart-cards/{number}

```
Parameters: {
  "name": "string",
  "active": false
}
```

1. System Operator returns updated smart card information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "GET SMART CARDS BY SPECIFIED FILTER".
2. A user sends a request to Endpoint "UPDATE AN EXISTING SMART CARD".

Endpoint URL: PATCH /smart-cards/{number}

```
Parameters: {
  "name": "string",
  "active": false
}
```

1. System Operator returns updated smart card information to User (See Result example below).

Post Conditions

Smart card is available.

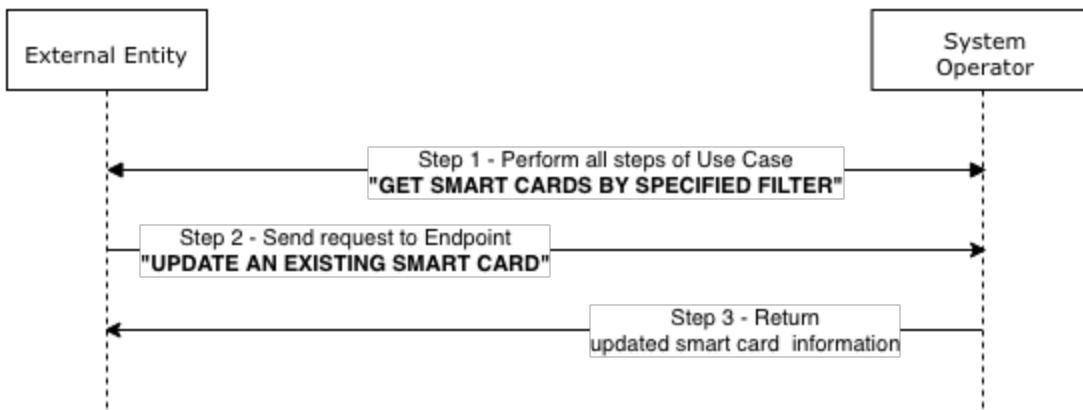
Result example

```
{  
  "smartCard": {  
    "cardNumber": "string",  
    "name": "string",  
    "organizationId": "string",  
    "active": false,  
    "createdAt": "2018-08-22T09:12:26.537Z"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

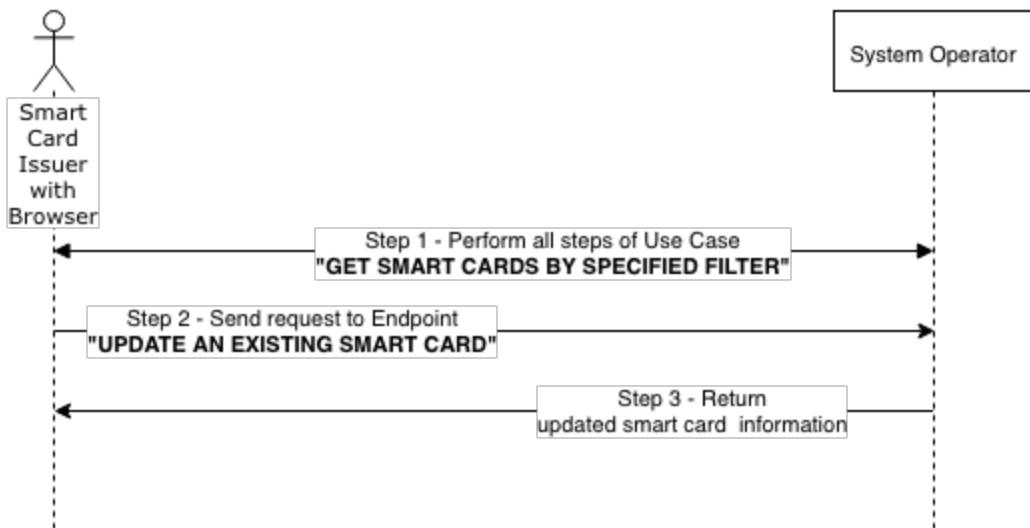
UPDATE AN EXISTING SMART CARD scheme

Use case: UPDATE AN EXISTING SMART CARD

Basic FFlow



Optional Web UI Flow



Smart cards viewing - as client

GET SMART CARDS OWNED BY CURRENT USER description

Use Case Name**GET SMART CARDS OWNED BY CURRENT USER****Brief Description**

A User or External Entity on behalf of a User with role permission SMART_CARD_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “GET SMART CARDS OWNED BY CURRENT USER”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: SMART_CARD_OWNER, e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “GET SMART CARDS OWNED BY CURRENT USER”.

Endpoint URL: GET /smart-cards

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of smart cards to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “GET SMART CARDS OWNED BY CURRENT USER”.

Endpoint URL: GET /smart-cards

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of smart cards User (See Result example below).

Post Conditions

Smart card is available.

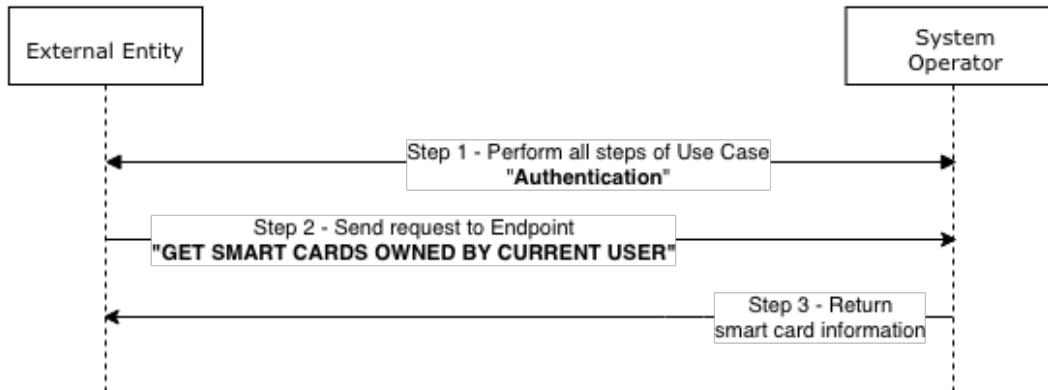
Result example

```
{  
  "records": [  
    {  
      "cardNumber": "string",  
      "name": "string",  
      "organizationId": "string",  
      "active": false,  
      "createdAt": "2018-08-22T09:12:26.527Z"  
    }  
,  
    {"status": "ok",  
     "message": "string"  
   }  
]
```

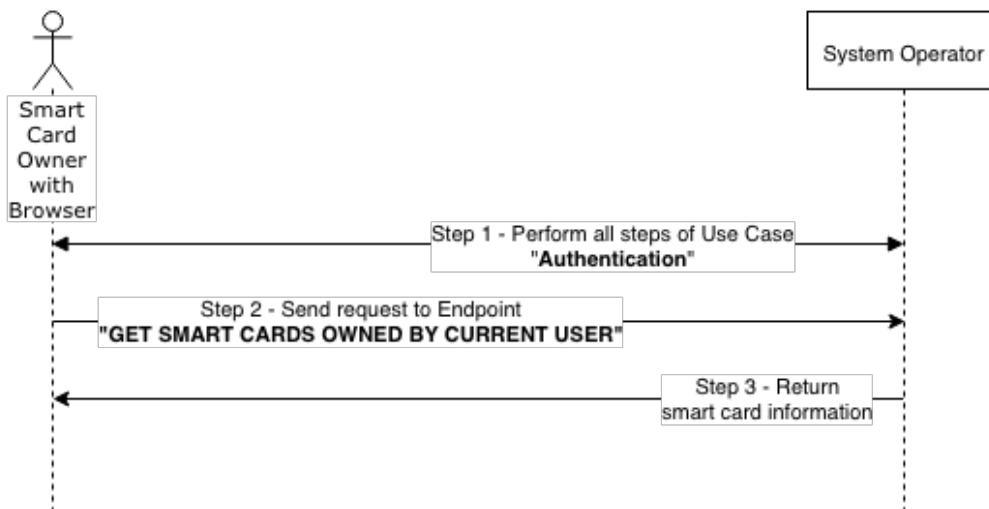
GET SMART CARDS OWNED BY CURRENT USER scheme

Use case: GET SMART CARDS OWNED BY CURRENT USER

Basic FLow



Optional Web UI Flow



Validation for smart card

Validate a smart card description

Use Case Name

Validate a smart card

Brief Description

A User or External Entity on behalf of a User with role permission "SMART_CARD_COIN_VALIDATOR" and "SMART_CARD_VALIDATOR", will go through all steps of "GET SMART CARDS OWNED BY CURRENT USER" (to obtain cardNumber) Use Case, and then send a request to Endpoint "Validate a smart card".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SMART_CARD_COIN_VALIDATOR" and "SMART_CARD_VALIDATOR", e.g. administrator, individual, financial specialist, merchant, payroll specialist, CFO, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "GET SMART CARDS OWNED BY CURRENT USER".
2. External Entity sends a request to Endpoint "Validate a smart card".

Endpoint URL: POST /smart-cards/validate

```
Parameters: {
  "cardNumber": "string"
}
```

1. System Operator returns status of card validation to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "GET SMART CARDS OWNED BY CURRENT USER".
2. A user sends a request to Endpoint "Validate a smart card".

Endpoint URL: POST /smart-cards/validate

```
Parameters: {
  "cardNumber": "string"
}
```

1. System Operator returns status of card validation to User (See Result example below).

Post Conditions

Smart card exists.

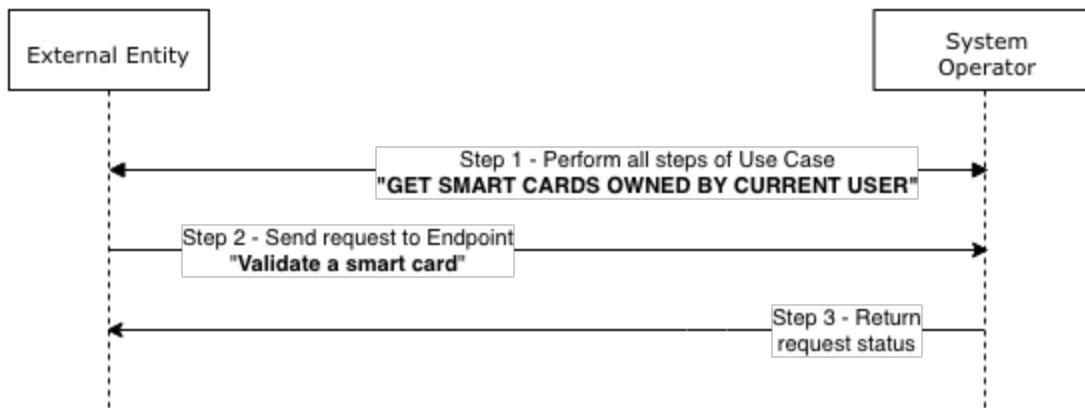
Result example

```
{  
  "owner": {  
    "id": "string",  
    "type": "string",  
    "name": "string"  
  },  
  "status": "ok",  
  "message": "string"  
}
```

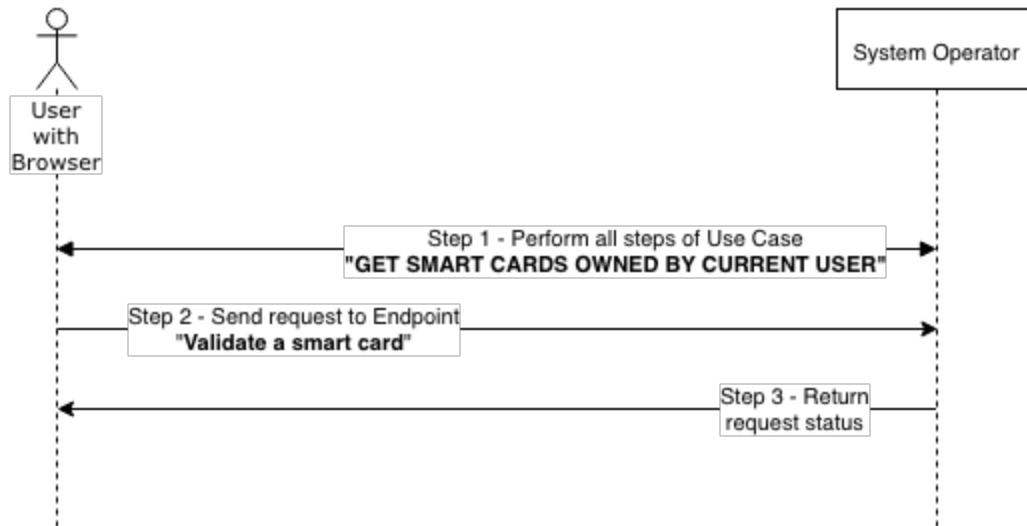
Validate a smart card scheme

Use case: Validate a smart card

Basic FFlow



Optional Web UI Flow



Paysafe cards

Retrieve all cards available for user description

Use Case Name

Retrieve all cards available for user

Brief Description

A User or External Entity on behalf of a User with role permission GATE_OPERATION_EXECUTOR will go through all steps of “Get users” Use Case, and then send a request to Endpoint “Retrieve all cards available for user”. “

Note: “View provider accounts” returns all provider account IDs.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: GATE_OPERATION_EXECUTOR, e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”. (View provider accounts XX)
2. External Entity sends a request to Endpoint “Retrieve all cards available for user”.

Endpoint URL: GET /gate-provider-accounts/{gate-provider-id}/paysafe-cards

Parameters: TOKEN - identifies authenticated user.

1. System Operator returns a List of cards to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users””.
2. A user sends a request to Endpoint “Retrieve all cards available for user”.

Endpoint URL: GET /gate-provider-accounts/{gate-provider-id}/paysafe-cards

Parameters: TOKEN

1. System Operator returns a List of cards to User (See Result example below).

Post Conditions

Card list is available.

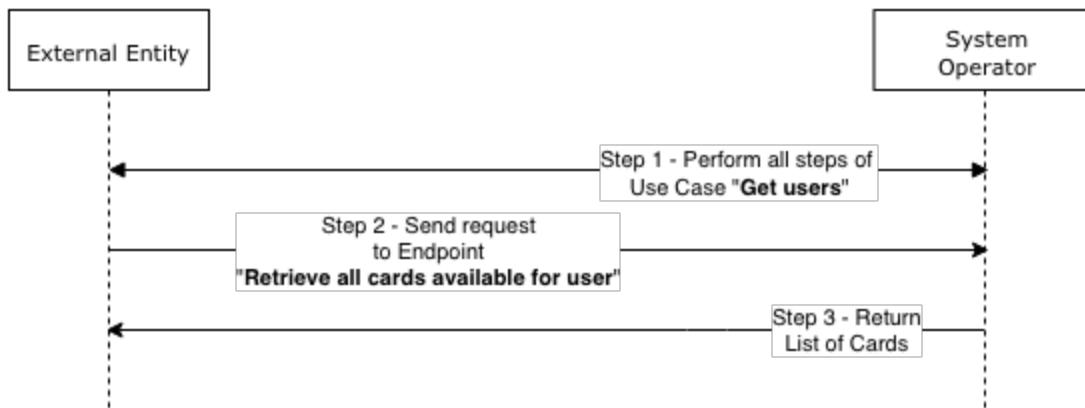
Result example

```
{  
  "cards": [  
    {  
      "id": "string",  
      "bin": "string",  
      "last4Digits": "string",  
      "mask": "string",  
      "verified": false,  
      "active": false  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

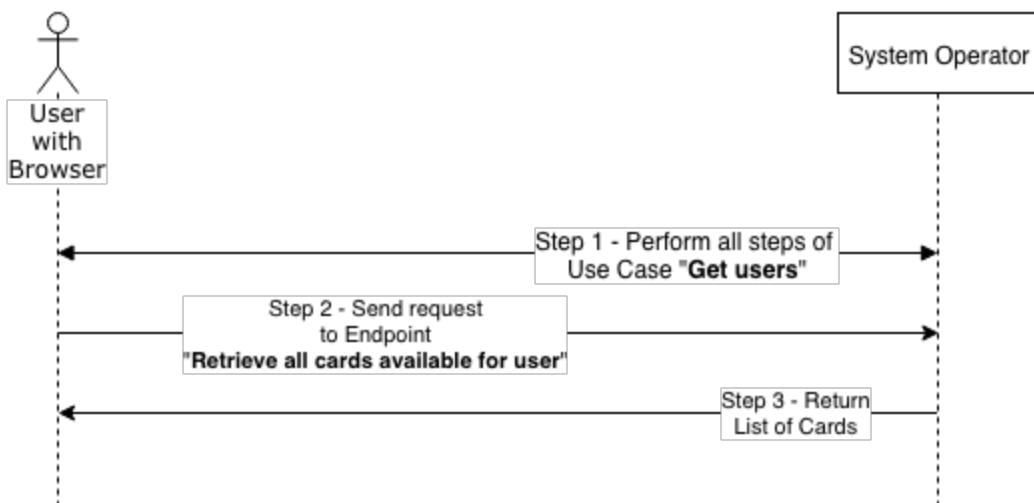
Retrieve all cards available for user scheme

Use case: Retrieve all cards available for user

Basic FFlow



Optional Web UI Flow



Subscriptions

Activate subscription description

Use Case Name

Activate subscription

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" (to get subscription ID) or "Get subscriptions for current user with Filter And Pagination" (to get subscription ID), and then send a request to Endpoint "Activate subscription".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. External Entity sends a request to Endpoint "Activate subscription".

Endpoint URL: PATCH /subscriptions/{id}/activate

Parameters: TOKEN - identifies authenticated user

1. System Operator returns activated subscription information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. A user sends a request to Endpoint "Activate subscription".

Endpoint URL: PATCH /subscriptions/{id}/activate

Parameters: TOKEN - identifies authenticated user

1. System Operator returns activated subscription information to User (See Result example below).

Post Conditions

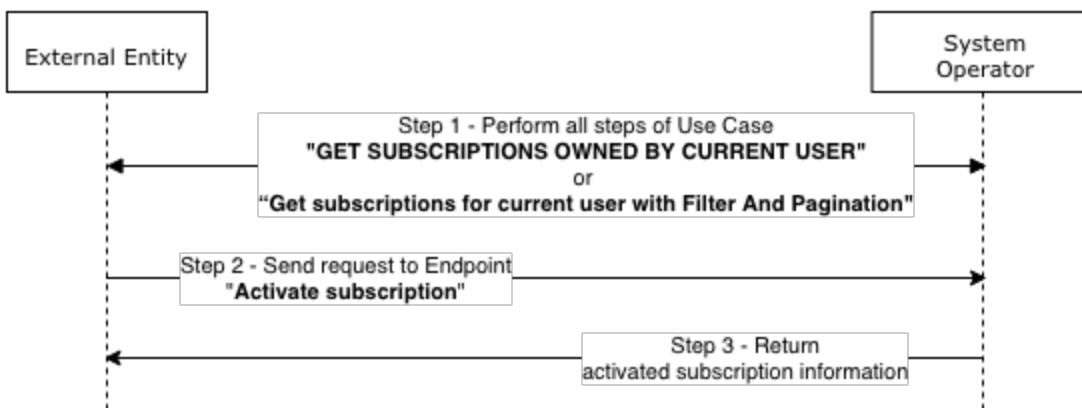
Subscription is available.

Result example

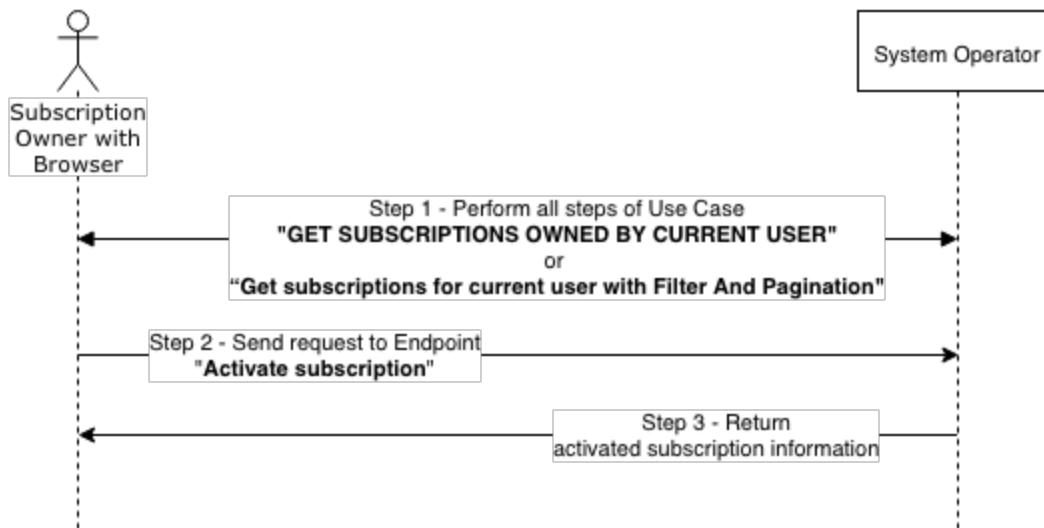
```
{  
  "id": "string",  
  "name": "string",  
  "expirationConfig": {  
    "type": "DATE",  
    "amount": 0,  
    "count": 0,  
    "expirationDate": "2018-08-24T05:06:13.660Z"  
  },  
  "recurringStartDate": "2018-08-24T05:06:13.660Z",  
  "type": "DAILY",  
  "frequency": 0,  
  "templateDto": {  
    "id": "string",  
    "type": "TRANSFER",  
    "name": "string",  
    "amount": 0,  
    "description": "string"  
  },  
  "subscriptionStatus": "ACTIVE",  
  "status": "ok",  
  "message": "string"  
}  
Activate subscription scheme
```

Use case: Activate subscription

Basic FFlow



Optional Web UI Flow



Create subscription description

Use Case Name

Create subscription

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Create subscription". The "SUBSCRIPTION_OWNER" should have the required input parameters.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create subscription".

Endpoint URL: POST /subscriptions

Parameters: {

```
"name": "string",
"expirationType": "DATE",
"amount": 0,
"count": 0,
"endDate": "string",
"recurringStartDate": "string",
"type": "DAILY",
"frequency": 0,
"templateId": "string"
}
```

1. System Operator returns new subscription information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Create subscription".

Endpoint URL: POST /subscriptions

Parameters: {
 "name": "string",
 "expirationType": "DATE",
 "amount": 0,
 "count": 0,
 "endDate": "string",
 "recurringStartDate": "string",
 "type": "DAILY",
 "frequency": 0,
 "templateId": "string"
 }

1. System Operator returns new subscription information to User (See Result example below).

Post Conditions

User authenticated and subscription parameters are available.

Result example

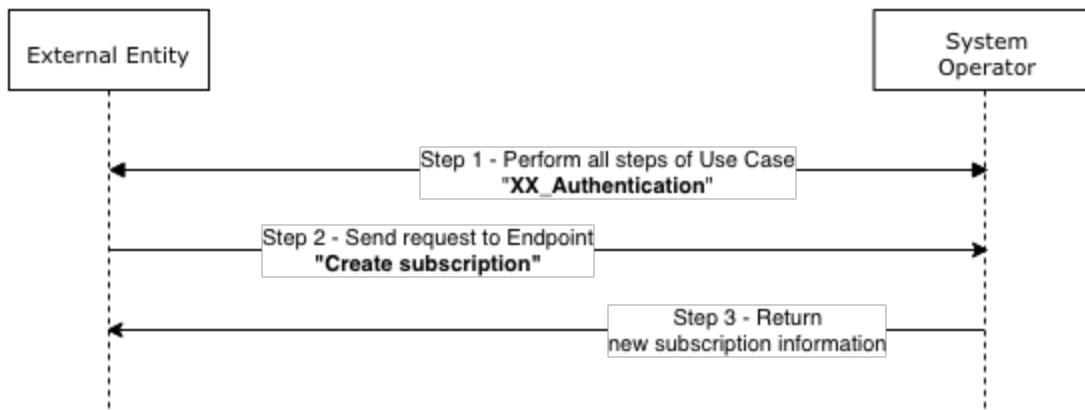
```
{
  "id": "string",
  "name": "string",
  "expirationConfig": {
    "type": "DATE",
    "amount": 0,
    "count": 0,
    "expirationDate": "2018-08-24T05:06:13.560Z"
  },
  "recurringStartDate": "2018-08-24T05:06:13.560Z",
  "type": "DAILY",
  "frequency": 0,
  "templateDto": {
    "id": "string",
    "type": "TRANSFER",
    "name": "string",
    "amount": 0,
    "description": "string"
  },
  "subscriptionStatus": "ACTIVE",
```

```
"status": "ok",  
"message": "string"  
}
```

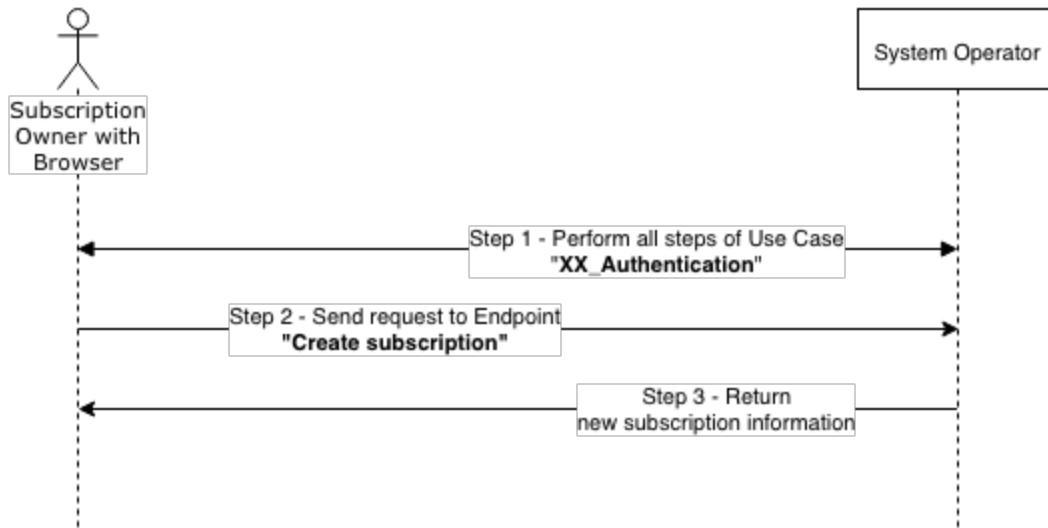
Create subscription scheme

Use case: Create subscription

Basic FFlow



Optional Web UI Flow



DELETE SUBSCRIPTION description

Use Case Name

Delete subscription

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" (to get subscription ID) or "Get subscriptions for current user with Filter And Pagination" (to get subscription ID), and then send a request to Endpoint "Delete subscription".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. External Entity sends a request to Endpoint "Delete subscription".

Endpoint URL: DELETE /subscriptions/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system status message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. A user sends a request to Endpoint "Delete subscription".

Endpoint URL: DELETE /subscriptions/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system status message to User (See Result example below).

Post Conditions

Subscription is available.

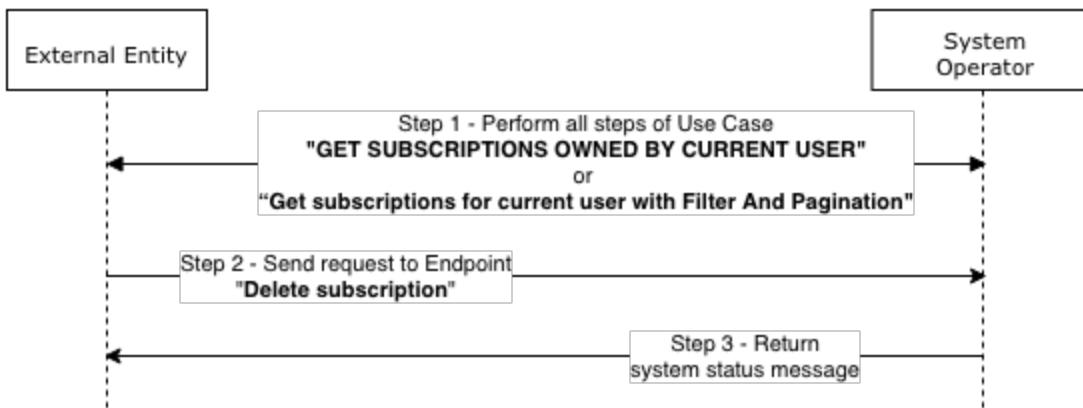
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

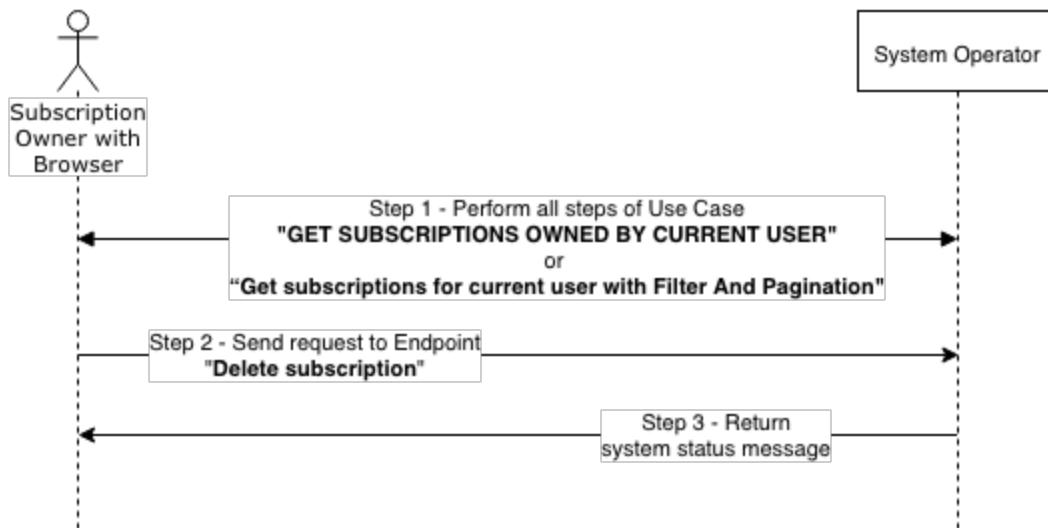
DELETE SUBSCRIPTION scheme

Use case: Delete subscription

Basic FFlow



Optional Web UI Flow



Get scheduled payments for the period description

Use Case Name

Get scheduled payments for the period

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get scheduled payments for the period".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get scheduled payments for the period".

Endpoint URL: POST /subscriptions/scheduled-payments

Parameters: {

```
"periodFrom": "2018-08-24T05:06:04.399Z",
"periodTo": "2018-08-24T05:06:04.399Z"
}
```

1. System Operator returns list of scheduled payments to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get scheduled payments for the period".

Endpoint URL: POST /subscriptions/scheduled-payments

Parameters: {

```
"periodFrom": "2018-08-24T05:06:04.399Z",
"periodTo": "2018-08-24T05:06:04.399Z"
}
```

1. System Operator returns list of scheduled payments to User (See Result example below).

Post Conditions

Scheduled payments are available.

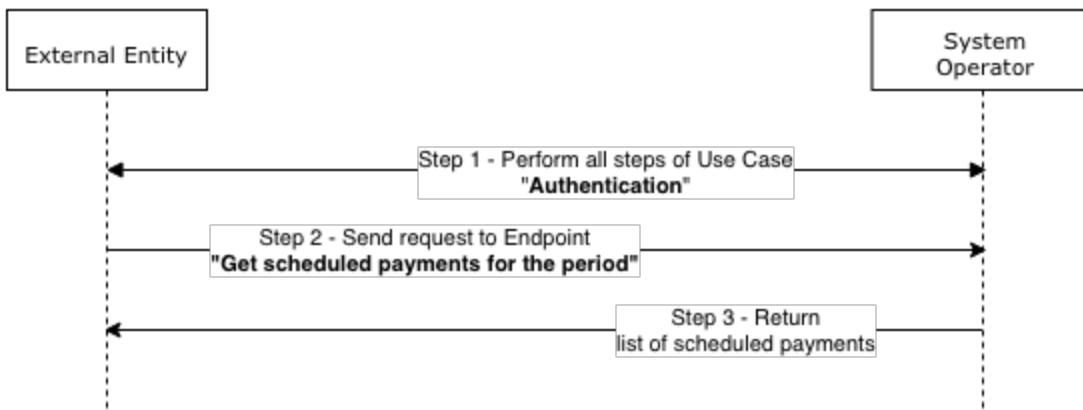
Result example

```
{  
  "scheduledPayments": [  
    {  
      "templateDto": {  
        "id": "string",  
        "type": "TRANSFER",  
        "name": "string",  
        "amount": 0,  
        "description": "string"  
      },  
      "dates": [  
        "2018-08-24T05:06:13.569Z"  
      ]  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

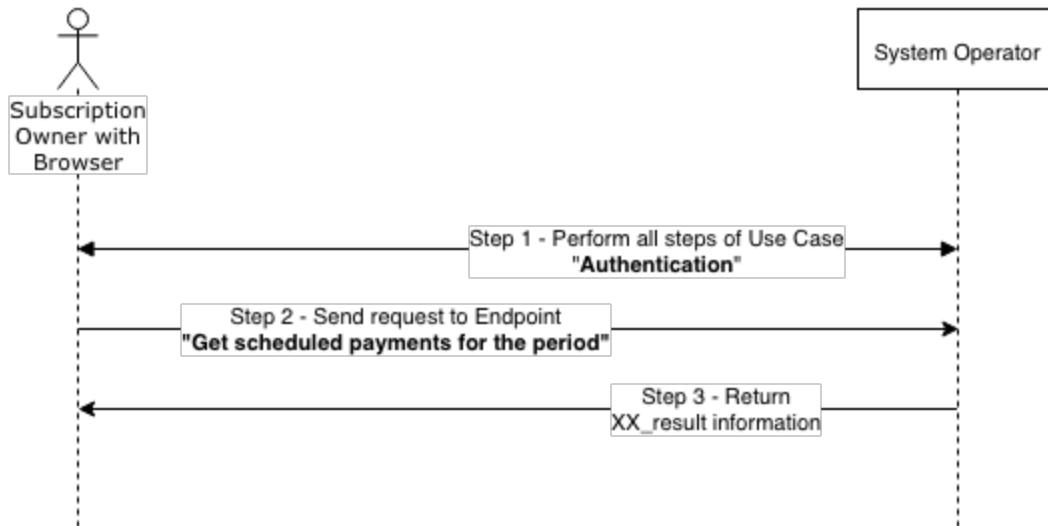
Get scheduled payments for the period scheme

Use case: Get scheduled payments for the period

Basic FFlow



Optional Web UI Flow



Get subscriptions for current user with Filter And Pagination description

Use Case Name

Get subscriptions for current user with Filter And Pagination

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get subscriptions for current user with Filter And Pagination". When there are no filtering parameter available, the filter could be omitted to get all subscriptions in the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get subscriptions for current user with Filter And Pagination".

Endpoint URL: POST /subscriptions/view

```
Parameters: {
  "filter": { - optional
    "id": "string",
    "startDateFrom": "2018-08-24T05:04:401Z",
    "name": "string",
    "status": "ACTIVE",
    "senderCoinSerial": "string",
    "recipientCoinSerial": "string",
    "recipientPhoneNumber": "string"
  },
  "sort": {
    "date": "asc",
    "name": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

1. System Operator returns list of subscriptions to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get subscriptions for current user with Filter And Pagination”.

Endpoint URL: POST /subscriptions/view

```
Parameters: {
  "filter": {
    "id": "string",
    "startDateFrom": "2018-08-24T05:06:04.401Z",
    "name": "string",
    "status": "ACTIVE",
    "senderCoinSerial": "string",
    "recipientCoinSerial": "string",
    "recipientPhoneNumber": "string"
  },
  "sort": {
    "date": "asc",
    "name": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

1. System Operator returns list of subscriptions to User (See Result example below).

Post Conditions

Subscriptions are available.

Result example

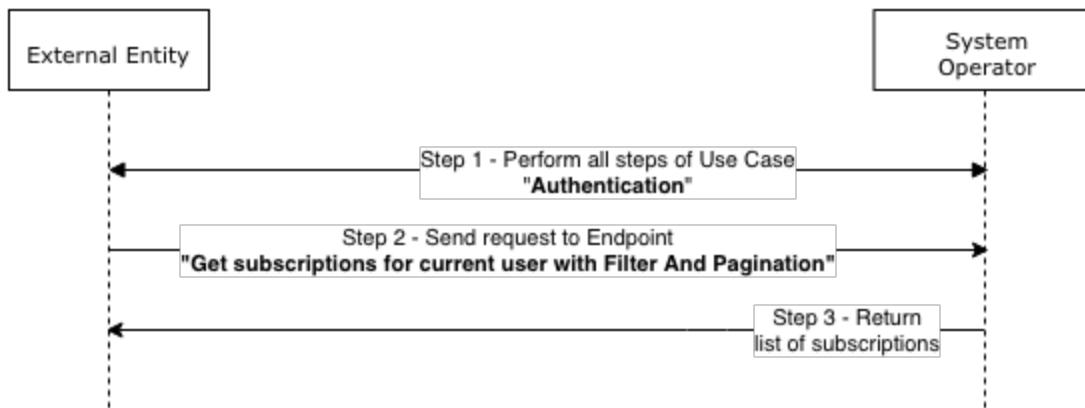
```
{
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
```

```
"records": [  
    {  
        "id": "string",  
        "name": "string",  
        "expirationConfig": {  
            "type": "DATE",  
            "amount": 0,  
            "count": 0,  
            "expirationDate": "2018-08-24T05:06:13.575Z"  
        },  
        "recurringStartDate": "2018-08-24T05:06:13.575Z",  
        "type": "DAILY",  
        "frequency": 0,  
        "templateDto": {  
            "id": "string",  
            "type": "TRANSFER",  
            "name": "string",  
            "amount": 0,  
            "description": "string"  
        },  
        "subscriptionStatus": "ACTIVE"  
    }  
]
```

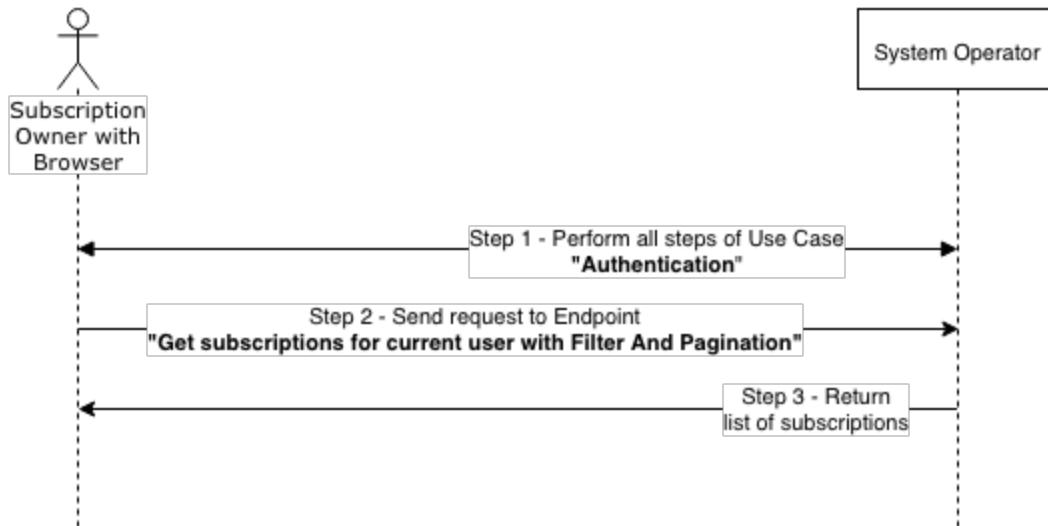
Get subscriptions for current user with Filter And Pagination scheme

Use case: Get subscriptions for current user with Filter And Pagination

Basic FFlow



Optional Web UI Flow



GET SUBSCRIPTIONS OWNED BY CURRENT USER description

Use Case Name

GET SUBSCRIPTIONS OWNED BY CURRENT USER

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "GET SUBSCRIPTIONS OWNED BY CURRENT USER".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "GET SUBSCRIPTIONS OWNED BY CURRENT USER".

Endpoint URL: GET /subscriptions

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of subscriptions to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "GET SUBSCRIPTIONS OWNED BY CURRENT USER".

Endpoint URL: GET /subscriptions

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of subscriptions to User (See Result example below).

Post Conditions

Subscriptions are available.

Result example

```
{  
  "subscriptions": [  
    {  
      "id": "string",  
      "name": "string",  
      "expirationConfig": {  
        "type": "DATE",  
        "amount": 0,  
        "count": 0,  
        "expirationDate": "2018-08-24T05:06:13.554Z"  
      },  
      "recurringStartDate": "2018-08-24T05:06:13.554Z",  
      "type": "DAILY",  
      "frequency": 0,  
      "templateDto": {  
        "id": "string",  
        "type": "TRANSFER",  
        "name": "string",  
        "amount": 0,  
        "description": "string"  
      },  
      "subscriptionStatus": "ACTIVE"  
    },  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

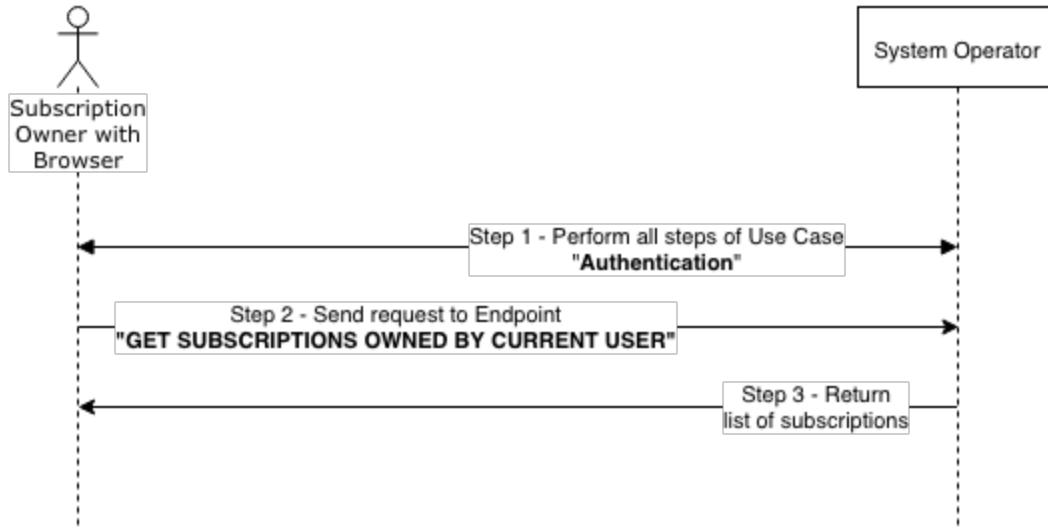
GET SUBSCRIPTIONS OWNED BY CURRENT USER scheme

Use case: GET SUBSCRIPTIONS OWNED BY CURRENT USER

Basic FFlow



Optional Web UI Flow



Stop subscription description

Use Case Name

Stop subscription

Brief Description

A User or External Entity on behalf of a User with role permission "SUBSCRIPTION_OWNER" will go through all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" (to obtain subscription ID) or "Get subscriptions for current user with Filter And Pagination" (to obtain subscription ID), and then send a request to Endpoint "Stop subscription".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "SUBSCRIPTION_OWNER", e.g. individual.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. External Entity sends a request to Endpoint "Stop subscription".

Endpoint URL: PATCH /subscriptions/{id}/stop

Parameters: TOKEN - identifies authenticated user

1. System Operator returns stopped subscription information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "GET SUBSCRIPTIONS OWNED BY CURRENT USER" or "Get subscriptions for current user with Filter And Pagination".
2. A user sends a request to Endpoint "Stop subscription".

Endpoint URL: PATCH /subscriptions/{id}/stop

Parameters: TOKEN - identifies authenticated user

1. System Operator returns stopped subscription information to User (See Result example below).

Post Conditions

Subscription is available.

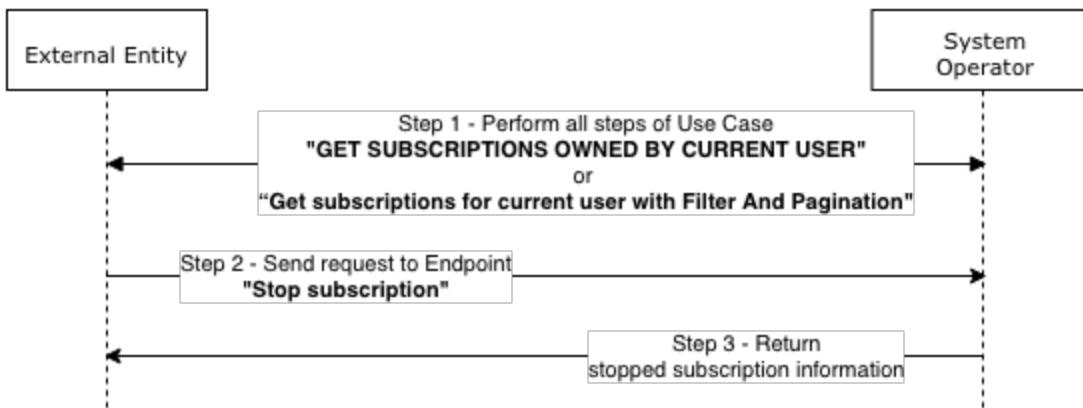
Result example

```
{  
  "id": "string",  
  "name": "string",  
  "expirationConfig": {  
    "type": "DATE",  
    "amount": 0,  
    "count": 0,  
    "expirationDate": "2018-08-24T05:06:13.660Z"  
  },  
  "recurringStartDate": "2018-08-24T05:06:13.660Z",  
  "type": "DAILY",  
  "frequency": 0,  
  "templateDto": {  
    "id": "string",  
    "type": "TRANSFER",  
    "name": "string",  
    "amount": 0,  
    "description": "string"  
  },  
  "subscriptionStatus": "ACTIVE",  
  "status": "ok",  
  "message": "string"  
}
```

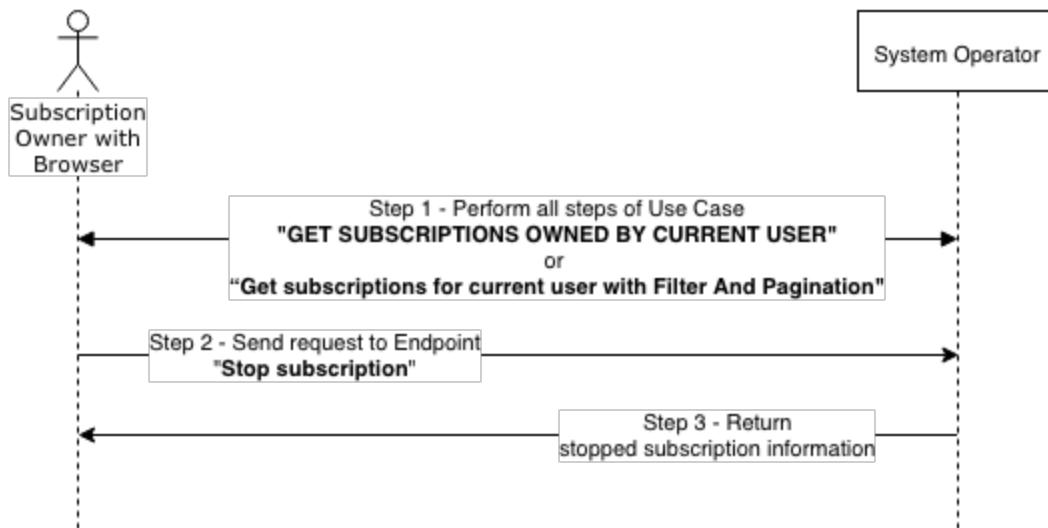
Stop subscription scheme

Use case: Stop subscription

Basic FFlow



Optional Web UI Flow



Mass payment management description

Use Case Name

Mass payment management

Brief Description

A User or External Entity on behalf of a User with role permission "MASSPAY_REQUEST_CREATION_EXECUTOR" or "MASSPAY_REQUEST_VALIDATION_EXECUTOR" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Mass payment management".

Note:

The filter is optional. When there are no filtering parameter available, the filter could be omitted to get all mass payments in the system.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MASSPAY_REQUEST_CREATION_EXECUTOR" or "MASSPAY_REQUEST_VALIDATION_EXECUTOR", e.g. payroll specialist or payroll manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Mass payment management".

Endpoint URL: POST /mass-payments/view

Parameters: {

```
"pageNumber": 0,
"pageSize": 0,
"filter": {
  "identifier": "string",
  "startDate": "2018-09-17T11:07:12.650Z",
  "endDate": "2018-09-17T11:07:12.650Z"
},
"sort": {
  "date": "asc"
}
```

1. System Operator returns list of mass payment items to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Mass payment management”.

Endpoint URL: POST /mass-payments/view

```
Parameters: {
  "pageNumber": 0,
  "pageSize": 0,
  "filter": {
    "identifier": "string",
    "startDate": "2018-09-17T11:07:12.650Z",
    "endDate": "2018-09-17T11:07:12.650Z"
  },
  "sort": {
    "date": "asc"
  }
}
```

1. System Operator returns list of mass payment items to User (See Result example below).

Post Conditions

Token is available.

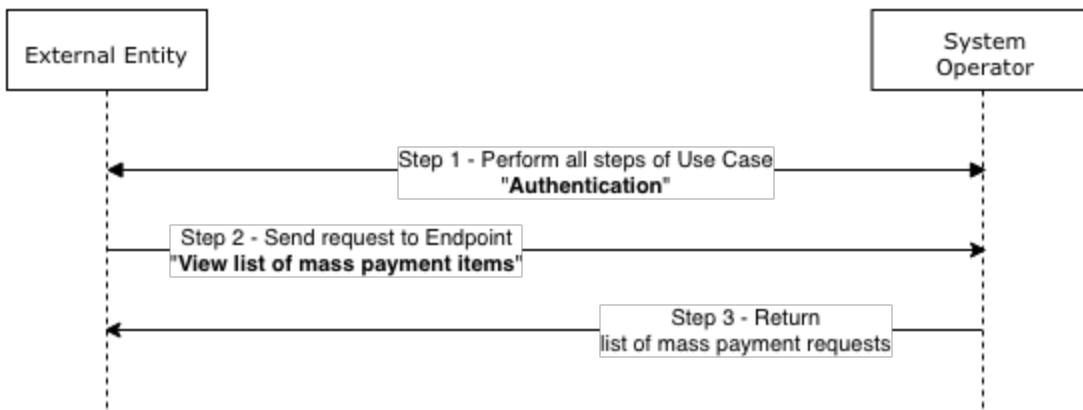
Result example

```
{
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "status": "ok",
  "message": "string",
  "records": [
    {}
  ]
}
```

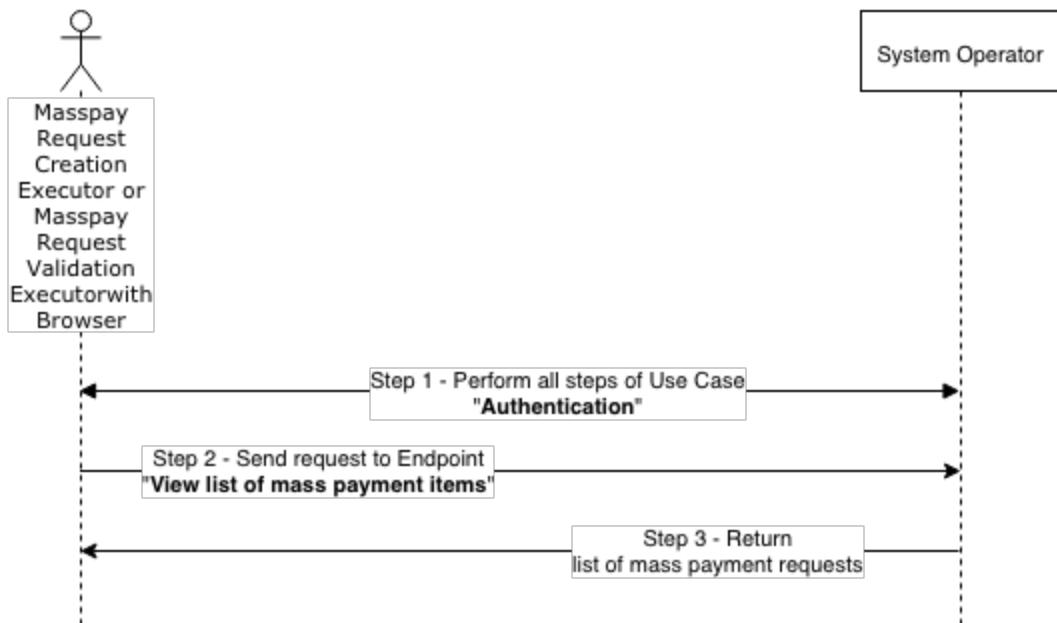
Mass payment management scheme

Use case: View list of mass payment items

Basic FFlow



Optional Web UI Flow



Point of sale

Create point of sale description

Use Case Name

Create a point of sale.

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Create a point of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Create a point of sale”.

Endpoint URL: POST /points-of-sale

Parameters:

```
{
  "name": "string",
  "description": "string",
  "website": "string",
  "resultUrl": "string",
  "serverUrl": "string",
  "trustAllCertificates": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Create a point of sale”.

Endpoint URL: POST /points-of-sale

Parameters:

```
{  
    "name": "string",  
    "description": "string",  
    "website": "string",  
    "resultUrl": "string",  
    "serverUrl": "string",  
    "trustAllCertificates": false  
}
```

3. System Operator returns result information to User (See Result example below).

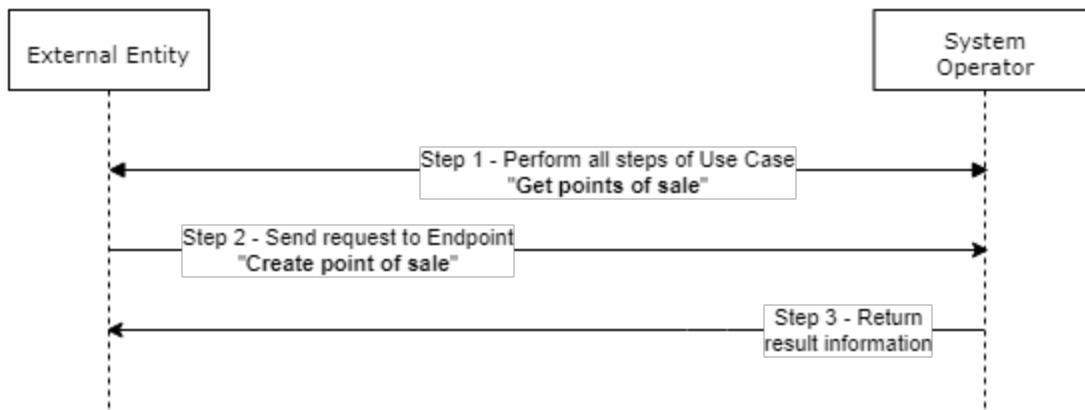
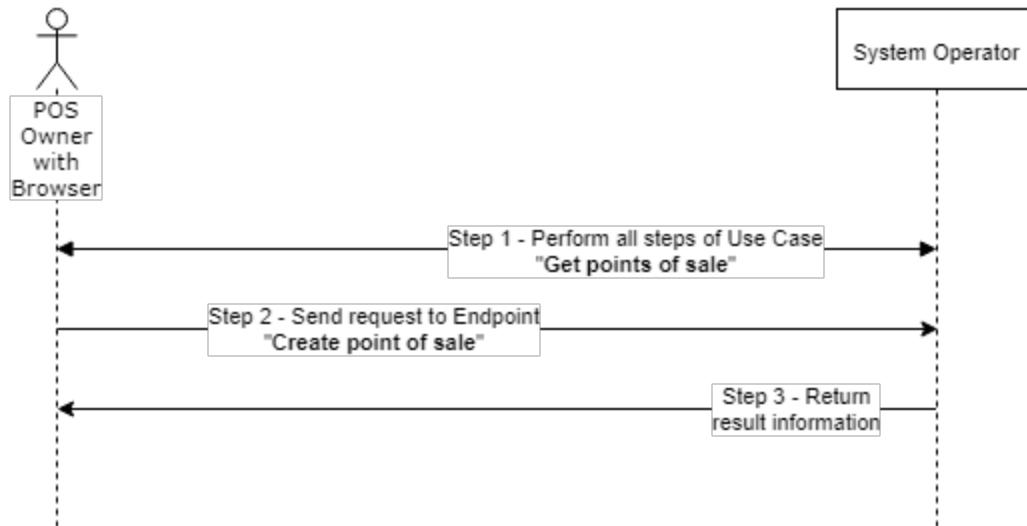
Post Conditions

New Point of Sale is available.

Result example

```
{  
    "pos": {  
        "secretCode": "string",  
        "serverUrl": "string",  
        "resultUrl": "string",  
        "trustAllCertificates": false,  
        "id": "string",  
        "name": "string",  
        "description": "string",  
        "website": "string",  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Create point of sale scheme

Use case: Create point of sale**Basic FFlow****Optional Web UI Flow**

Delete point of sale description

Use Case Name**Delete point of sale****Brief Description**

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Delete point of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Delete point of sale”.

Endpoint URL: DELETE /points-of-sale/{posId}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Delete point of sale”.

Endpoint URL: DELETE /points-of-sale/{posId}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

Point of Sale is deleted.

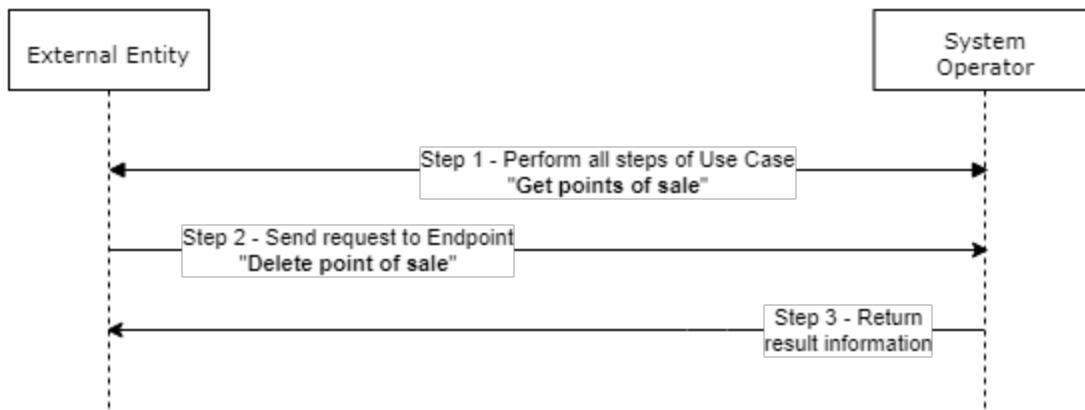
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

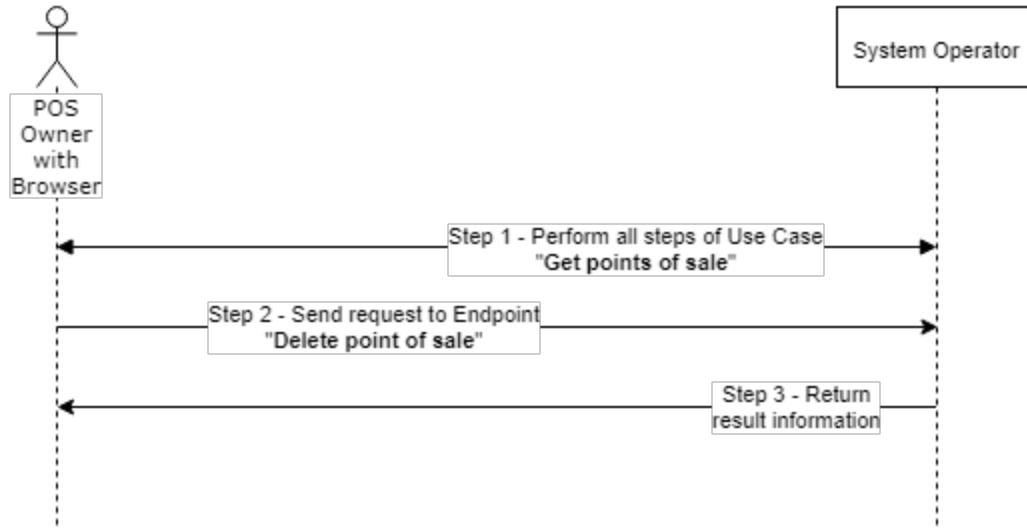
Delete point of sale scheme

Use case: Delete point of sale

Basic FFlow



Optional Web UI Flow



Generate new secret code for point of sale description

Use Case Name

Generate new secret code for point of sale

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Generate new secret code for point of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Generate new secret code for point of sale”.

Endpoint URL: POST /points-of-sale/{posId}/generate-secret

Parameters: TOKEN

3. System Operator returns New POS secret code to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Generate new secret code for point of sale”.

Endpoint URL: POST /points-of-sale/{posId}/generate-secret

Parameters: TOKEN

3. System Operator returns New POS secret code to User (See Result example below).

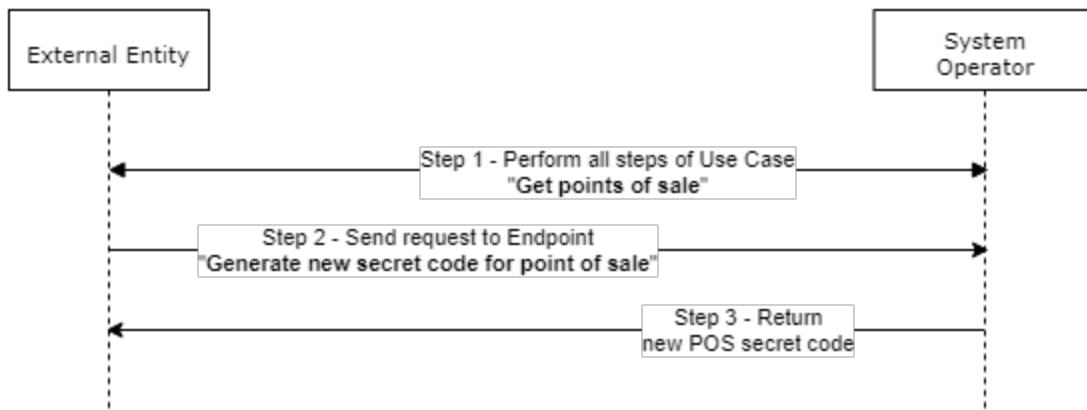
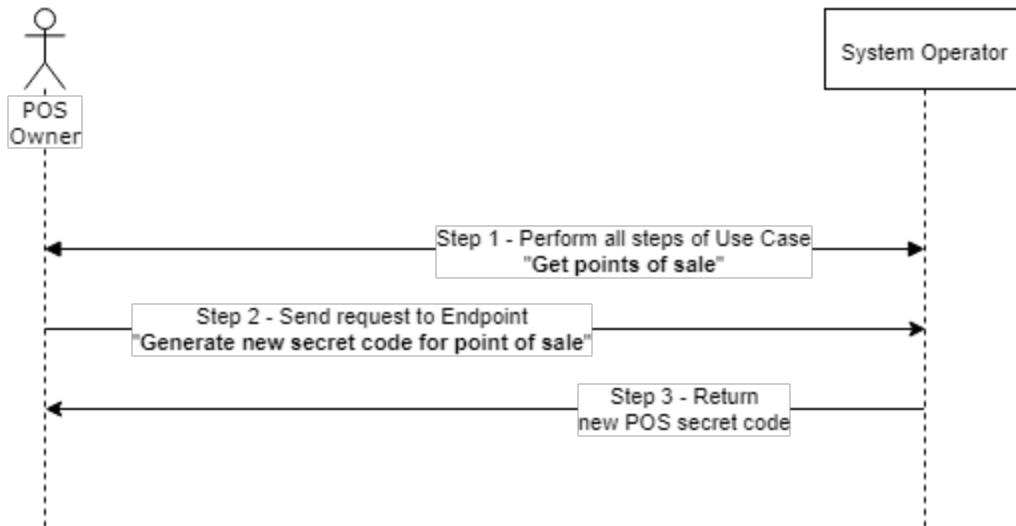
Post Conditions

New POS secret code is available.

Result example

```
{  
    "pos": {  
        "secretCode": "string",  
        "serverUrl": "string",  
        "resultUrl": "string",  
        "trustAllCertificates": false,  
        "id": "string",  
        "name": "string",  
        "description": "string",  
        "website": "string",  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Generate new secret code for point of sale scheme

Use case: Generate new secret code for point of sale**Basic FFlow****Optional Web UI Flow**

Get information about single point of sale description

Use Case Name

Get information about a single point of sale

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Get information about the single point of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Get information about a single point of sale”.

Endpoint URL: GET /points-of-sale/{posId}

Parameters: TOKEN

3. System Operator returns POS information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Get information about a single point of sale”.

Endpoint URL: GET /points-of-sale/{posId}

Parameters: TOKEN

3. System Operator returns POS information to User (See Result example below).

Post Conditions

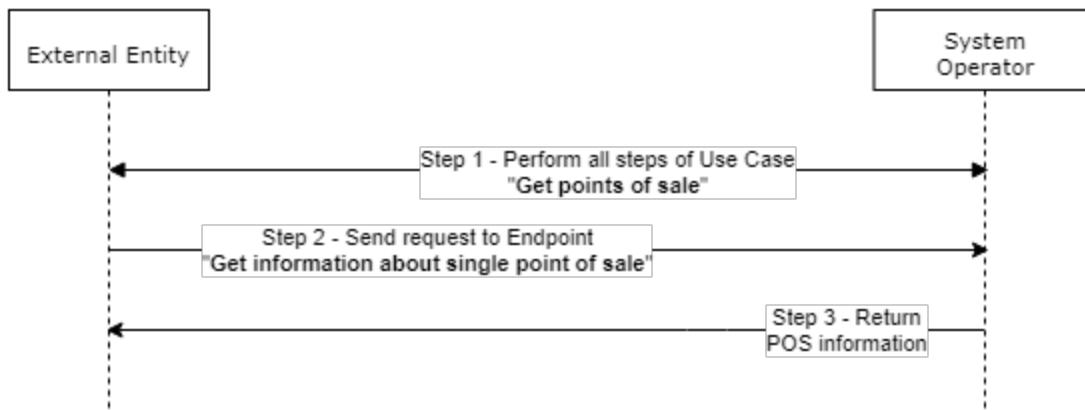
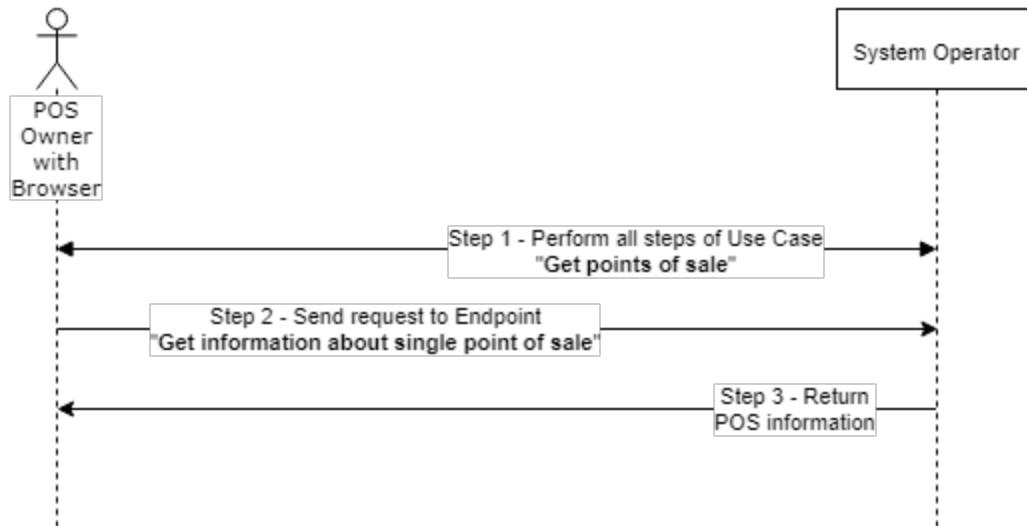
POS information is available.

Result example

```
{
  "pos": {
    "id": "string",
    "name": "string",
    "description": "string",
    "website": "string",
    "active": false
  },
  "status": "ok",
  "message": "string"
}
```

Get information about single point of sale scheme

Use case: Get information about single point of sale

Basic FFlow**Optional Web UI Flow**

Get merchant payment transactions for specified POS description

Use Case Name

Get merchant payment transactions for specified POS

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Get merchant payment transactions for specified POS”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Get merchant payment transactions for specified POS”.

Endpoint URL: POST /points-of-sale/{posId}/transactions/view

Parameters:

```
{
  "filter": {
    "ids": [
      "string"
    ],
    "statuses": [
      "limited"
    ],
    "dateFrom": "2018-08-21T09:45:12.092Z",
    "dateTo": "2018-08-21T09:45:12.092Z",
    "coinSerials": [
      "string"
    ],
    "issuerIds": [
      "string"
    ],
    "currencyCodes": [
      "string"
    ],
    "paymentIdentifiers": [
      0
    ]
  },
  "sort": {
    "date": "asc",
    "status": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns a List of merchant payment transactions to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Get merchant payment transactions for specified POS”.

Endpoint URL: POST /points-of-sale/{posId}/transactions/view

Parameters:

```
{
  "sort": {
    "date": "asc",
    "status": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns a List of merchant payment transactions to User (See Result example below).

Post Conditions

List of merchant payment transactions is available.

Result example

```
{
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0,
  "records": [
    {
      "posId": "string",
      "invoiceAmount": 0,
      "paymentIdentifier": 0,
      "description": "string",
      "payerId": "string",
      "merchantCoinSerial": "string",
      "id": "string",
      "createdAt": "2018-08-21T09:45:18.060Z",
      "updatedAt": "2018-08-21T09:45:18.060Z",
      "type": "string",
      "status": "limited",
      "requestIdentifier": 0,
      "requestStatus": "limited",
      "transactions": [
        {
          "id": "string",
          "parentId": "string",
          "type": "transfer",
          "from": {
            "serial": "string",
            "organizationId": "string",
            "amount": 0
          }
        }
      ]
    }
  ]
}
```

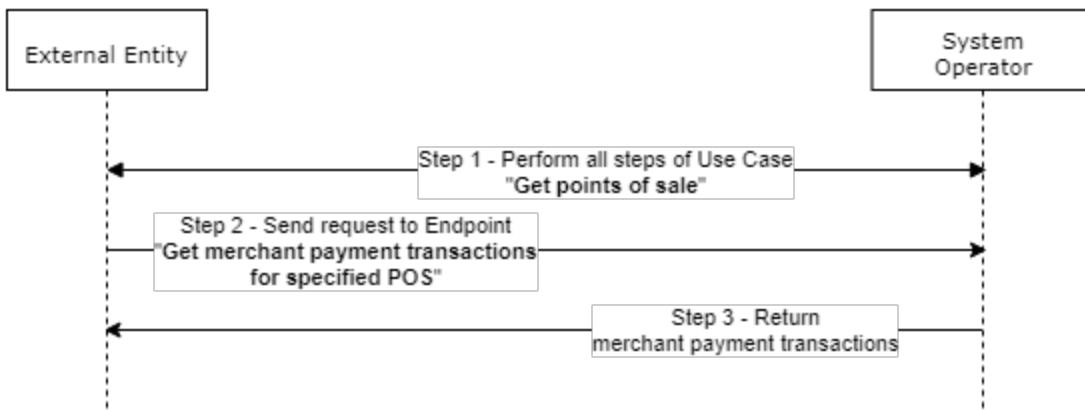
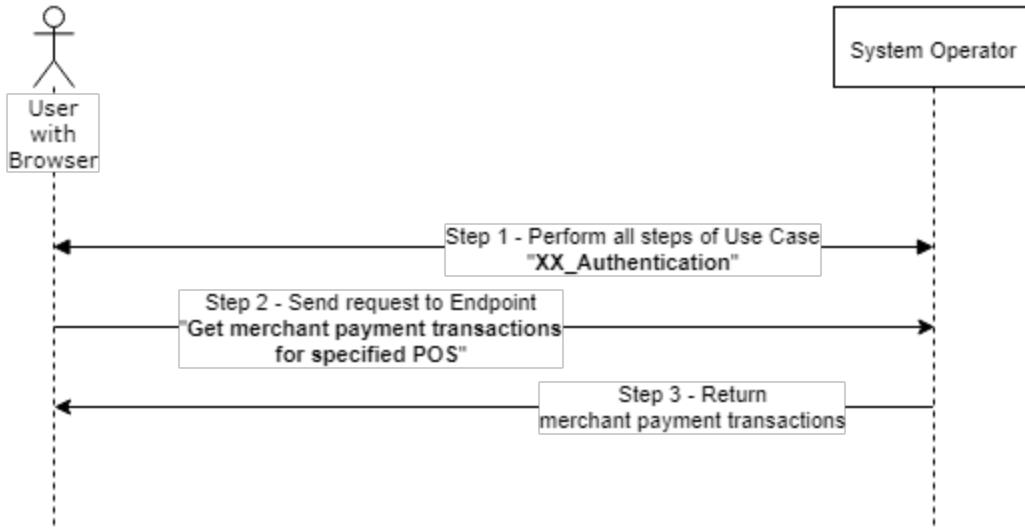
```
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "to": {
        "serial": "string",
        "organizationId": "string",
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "amount": 0,
    "performedAt": "2018-08-21T09:45:18.060Z",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
],
"children": [
{
    "id": "string",
    "createdAt": "2018-08-21T09:45:18.060Z",
    "updatedAt": "2018-08-21T09:45:18.060Z",
    "type": "string",
    "status": "limited",
    "requestIdentifier": 0,
    "requestStatus": "limited",
    "transactions": [
    {
        "id": "string",
        "parentId": "string",
        "type": "transfer",
        "from": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        },
        "to": {
            "serial": "string",
            "organizationId": "string",
            "organizationName": "string",
            "technical": false,
            "type": "regular_commission",
            "issuer": {
                "id": "string",
                "sn": "string",
                "currency": "string"
            }
        }
    }
]
}
```

```
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "to": {
        "serial": "string",
        "organizationId": "string",
        "organizationName": "string",
        "technical": false,
        "type": "regular_commission",
        "issuer": {
            "id": "string",
            "sn": "string",
            "currency": "string"
        }
    },
    "amount": 0,
    "performedAt": "2018-08-21T09:45:18.060Z",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    }
}
],
"children": [
    {
        ...
    }
],
"errorMessage": "string"
},
],
"errorMessage": "string"
```

```
        }  
    ]  
}
```

Get merchant payment transactions for specified POS scheme

Use case: Get merchant payment transactions for specified POS

Basic FFlow**Optional Web UI Flow**

Get points of sale description

Use Case Name

Get points of sale

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get points of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get points of sale”.

Endpoint URL: GET /points-of-sale

Parameters: TOKEN

3. System Operator returns Points of Sale to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get points of sale”.

Endpoint URL: GET /points-of-sale

Parameters: TOKEN

3. System Operator returns Points of Sale to User (See Result example below).

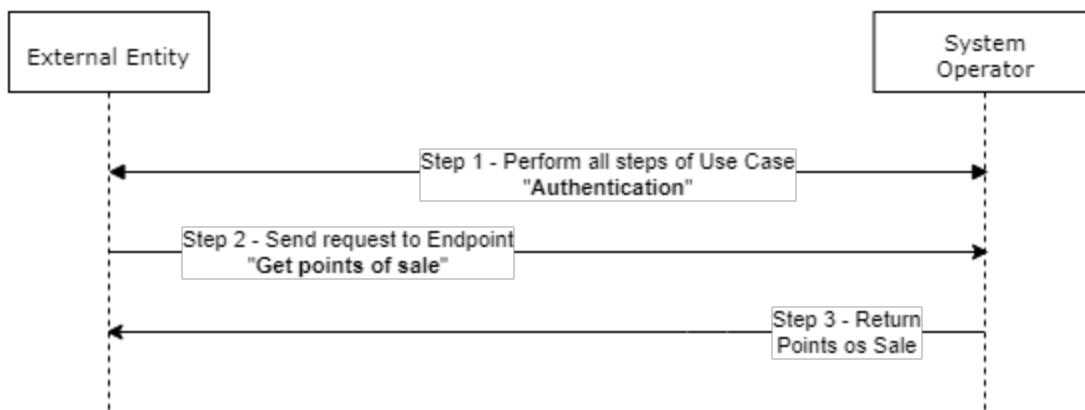
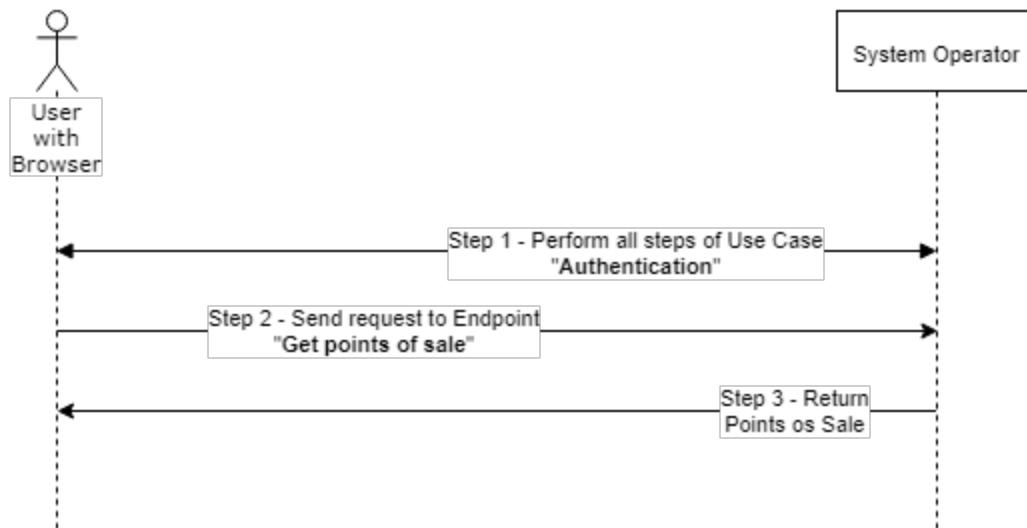
Post Conditions

List of Point of Sale is available.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "records": [  
        {  
            "secretCode": "string",  
            "serverUrl": "string",  
            "resultUrl": "string",  
            "trustAllCertificates": false,  
            "id": "string",  
            "name": "string",  
            "description": "string",  
            "website": "string",  
            "active": false  
        }  
    ]  
}
```

Get points of sale scheme

Use case: Get points of sale**Basic FFlow****Optional Web UI Flow**

Update point of sale description

Use Case Name

Update point of sale

Brief Description

A User or External Entity on behalf of a User with role permission POINT_OF_SALE_OWNER will go through all steps of “Get points of sale” Use Case, and then send a request to Endpoint “Update point of sale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: POINT_OF_SALE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale”.
2. External Entity sends a request to Endpoint “Update point of sale”.

Endpoint URL: PATCH /points-of-sale/{posId}

Parameters:

```
{
  "name": "string",
  "description": "string",
  "website": "string",
  "resultUrl": "string",
  "serverUrl": "string",
  "trustAllCertificates": false
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale”.
2. A user sends a request to Endpoint “Update point of sale”.

Endpoint URL: PATCH /points-of-sale/{posId}

Parameters:

```
{  
    "name": "string",  
    "description": "string",  
    "website": "string",  
    "resultUrl": "string",  
    "serverUrl": "string",  
    "trustAllCertificates": false  
}
```

3. System Operator returns result information to User (See Result example below).

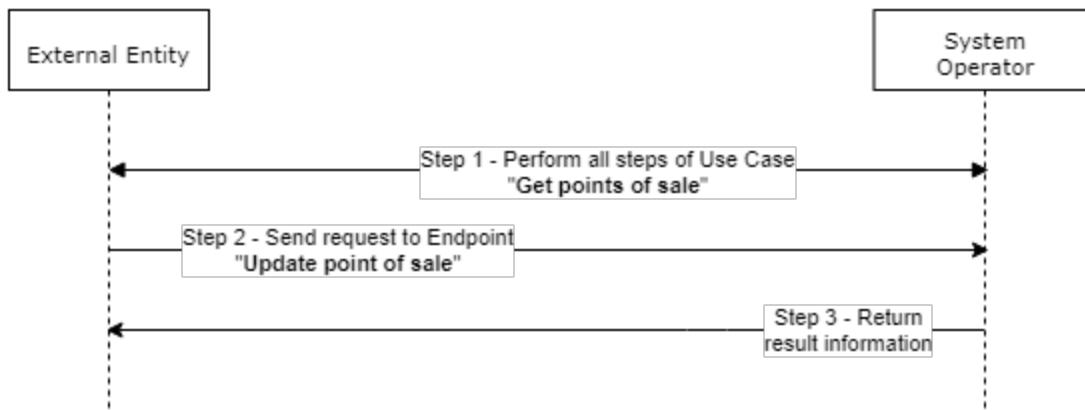
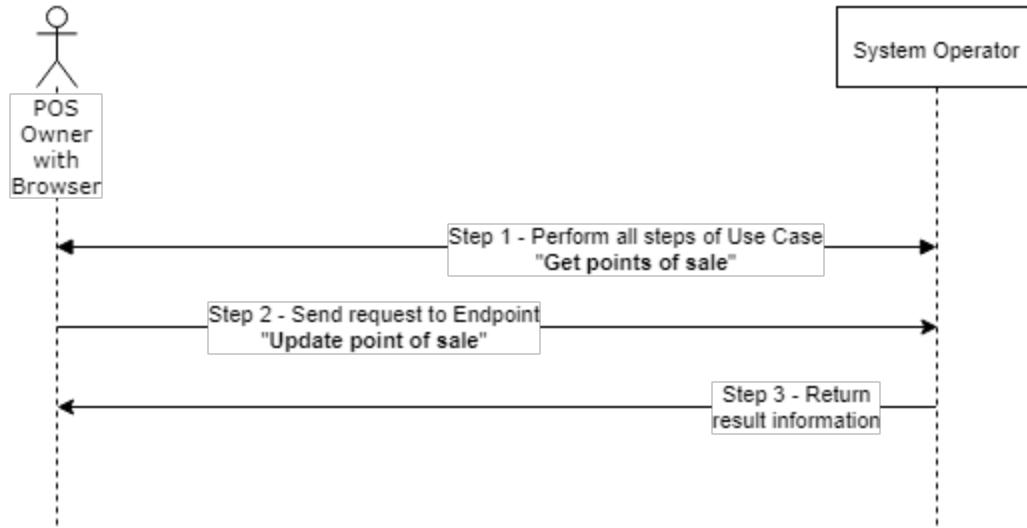
Post Conditions

Updated POS is available.

Result example

```
{  
    "pos": {  
        "secretCode": "string",  
        "serverUrl": "string",  
        "resultUrl": "string",  
        "trustAllCertificates": false,  
        "id": "string",  
        "name": "string",  
        "description": "string",  
        "website": "string",  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Update point of sale scheme

Use case: Update point of sale**Basic FFlow****Optional Web UI Flow**

Invoice

Calculate commission for invoice as merchant description

Use Case Name

Calculate commission for invoice as a merchant

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Calculate commission for invoice as merchant”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Calculate commission for invoice as merchant”.

Endpoint URL: POST /invoices/calculate

Parameter:

```
{
  "payerContact": "string",
  "recipientCoin": "string",
  "amount": 0
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Calculate commission for invoice as merchant”.

Endpoint URL: POST /invoices/calculate

Parameter:

```
{
  "payerContact": "string",
  "recipientCoin": "string",
  "amount": 0
}
```

3. System Operator returns result information to User (See Result example below).

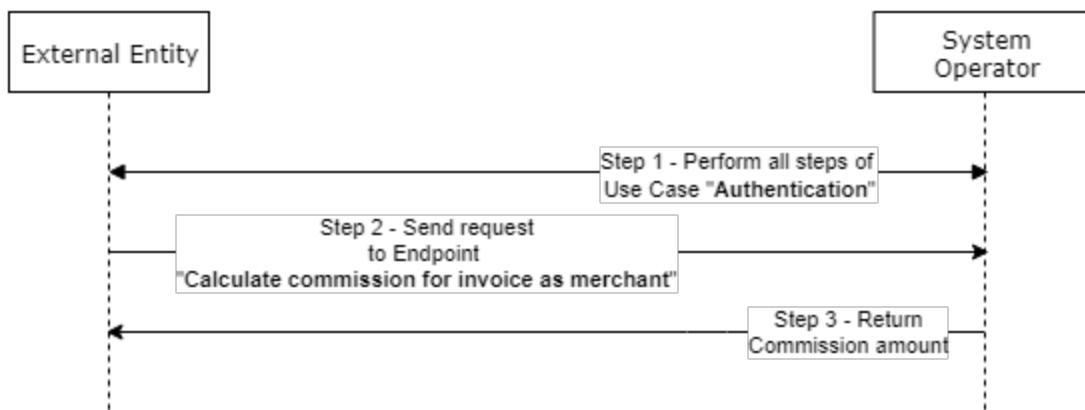
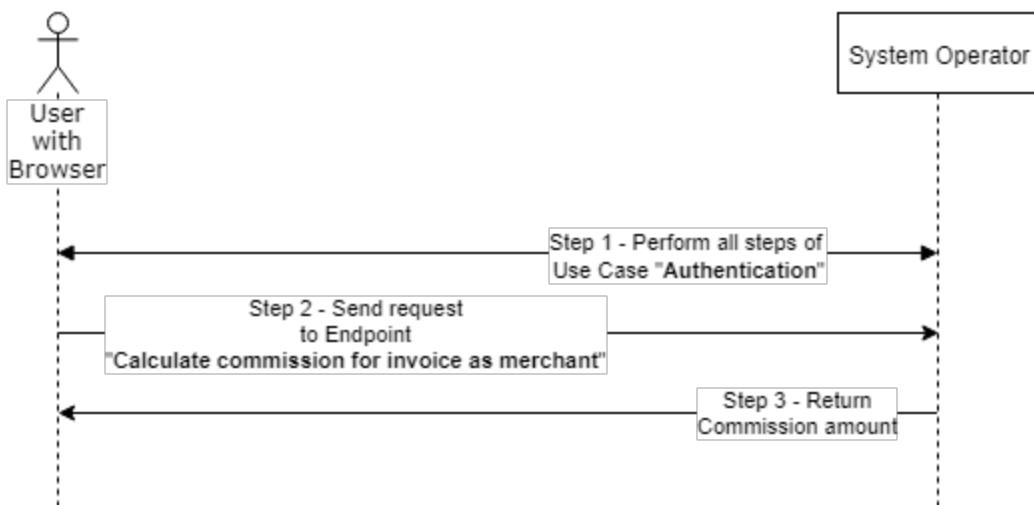
Post Conditions

Commission is available

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for invoice as merchant scheme

Use case: Calculate commission for invoice as merchant**Basic FFlow****Optional Web UI Flow**

Calculate commission for specified invoice as payer description

Use Case Name

Calculate commission for specified invoice as payer

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_PAYER will go through all steps of “View invoices” Use Case, and then send a request to Endpoint “Calculate commission for specified invoice as payer”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_PAYER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View invoices”.
2. External Entity sends a request to Endpoint “Calculate commission for a specified invoice as payer”.

Endpoint URL: POST /invoices/{identifier}/calculate

Parameter:

```
{
  "payerCoin": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.
2. A user sends a request to Endpoint “Calculate commission for a specified invoice as payer”.

Endpoint URL: POST /invoices/{identifier}/calculate

Parameter:

```
{
  "payerCoin": "string"
}
```

3. System Operator returns result information to User (See Result example below).

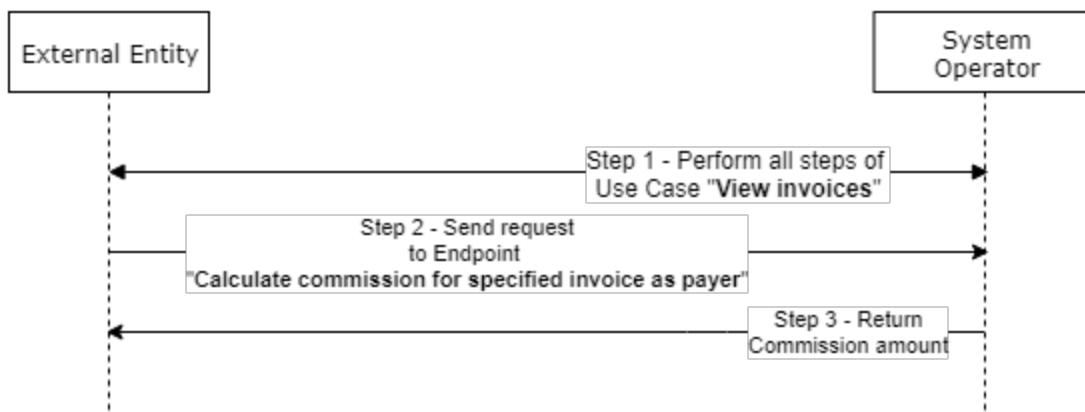
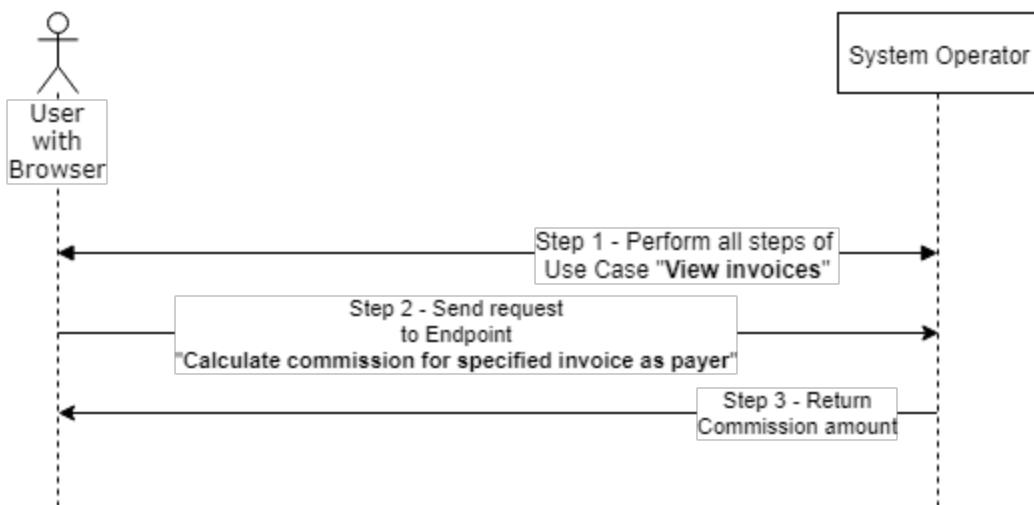
Post Conditions

Commission is available

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission for specified invoice as payer scheme

Use case: Calculate commission for specified invoice as payer**Basic FFlow****Optional Web UI Flow**

Create an invoice description

Use Case Name

Create an invoice

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER

will go through all steps of “View invoices” Use Case, and then send a request to Endpoint “Create an invoice”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View invoices”.

2. External Entity sends a request to Endpoint “Create an invoice”.

Endpoint URL: POST /invoices

Parameter:

```
{
  "name": "string",
  "payerContact": "string",
  "recipientCoin": "string",
  "data": {
    "productCode": "string",
    "productPrice": 0,
    "description": "string",
    "count": 0,
    "terms": "string"
  },
  "amount": 0,
  "expiresAt": "2018-08-06T13:20:27.576Z"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.

2. A user sends a request to Endpoint “Create an invoice”.

Endpoint URL: POST /invoices

Parameter:

```
{  
    "name": "string",  
    "payerContact": "string",  
    "recipientCoin": "string",  
    "data": {  
        "productCode": "string",  
        "productPrice": 0,  
        "description": "string",  
        "count": 0,  
        "terms": "string"  
    },  
    "amount": 0,  
    "expiresAt": "2018-08-06T13:20:27.576Z"  
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

Invoice created

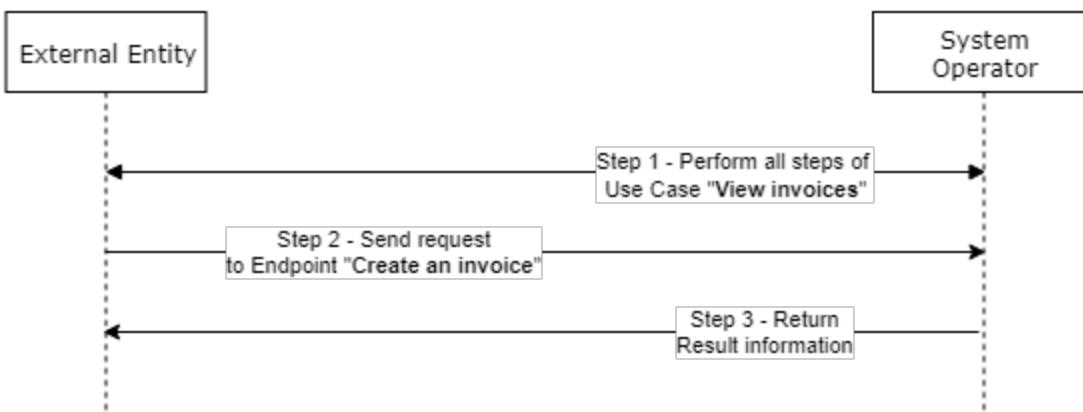
Result example

```
{  
    "invoice": {  
        "identifier": "string",  
        "name": "string",  
        "createdAt": "2018-08-06T13:20:28.919Z",  
        "createdByUser": {  
            "id": "string",  
            "name": "string"  
        },  
        "status": "initiated",  
        "payer": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "payerContact": "string",  
        "merchantName": "string",  
        "totalPrice": 0,  
        "expiresAt": "2018-08-06T13:20:28.919Z",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "data": {  
            "productCode": "string",  
            "productPrice": 0,  
            "description": "string",  
            "count": 0,  
            "terms": "string"  
        },  
        "paymentCode": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

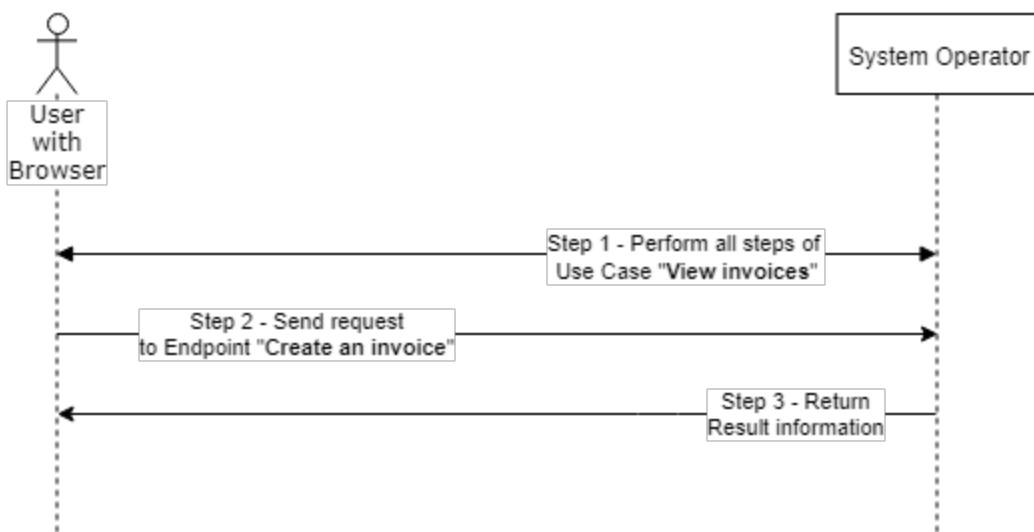
Create an invoice scheme

Use case: Create an invoice

Basic FLow



Optional Web UI Flow



Delete invoice description

Use Case Name

Delete invoice

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER, INVOICE_PAYER will go through all steps of “View invoices” Use Case, and then send a request to Endpoint “Delete invoice”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, INVOICE_PAYER.

2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View invoices”.

2. External Entity sends a request to Endpoint “Delete invoice”.

Endpoint URL: DELETE /invoices/{identifier}

Parameter: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.

2. A user sends a request to Endpoint “Delete invoice”.

Endpoint URL: DELETE /invoices/{identifier}

Parameter: TOKEN

3. System Operator returns result information to User (See Result example below).

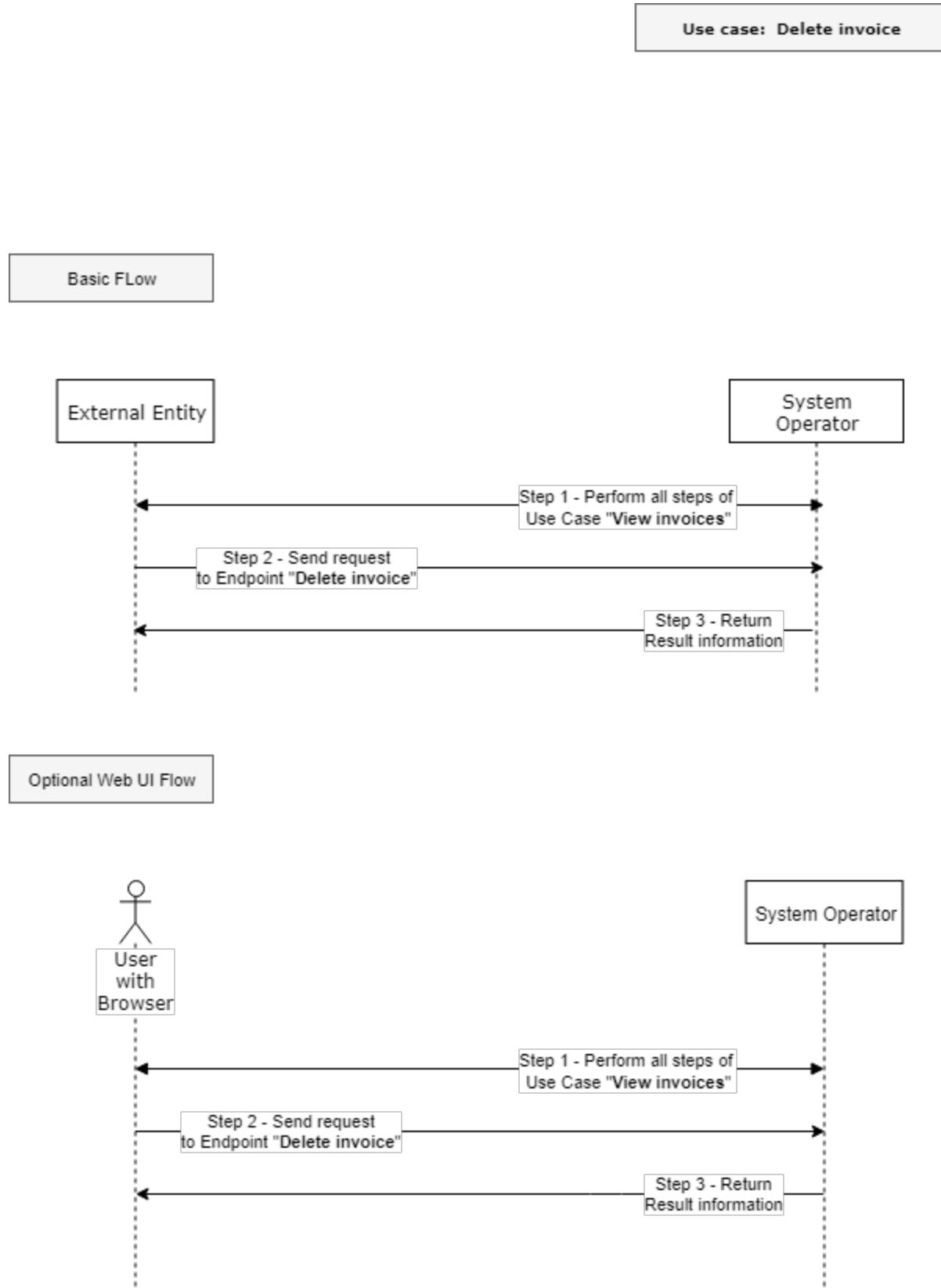
Post Conditions

Invoice deleted

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete invoice scheme



Get file attached to the invoice description

Use Case Name

Get file attached to the invoice

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER, INVOICE_PAYER will go through all steps of “Get information about invoice by identifier” Use Case, and then send a request to Endpoint “Get file attached to the invoice”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, INVOICE_PAYER.

2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get information about invoice by the identifier”.

Return: "terms": "string" - will contain attached media file Id.

2. External Entity sends a request to Endpoint “Get file attached to the invoice”.

Endpoint URL: GET /invoices/{identifier}/files/{fileId}

Parameter: Security TOKEN

3. System Operator returns requested information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get information about invoice by the identifier”.

Return: "terms": "string" - will contain attached media file Id.

2. A user sends a request to Endpoint “Get file attached to the invoice”.

Endpoint URL: GET /invoices/{identifier}/files/{fileId}

Parameter: Security TOKEN

3. System Operator returns requested information to User (See Result example below).

Post Conditions

File information is available for review

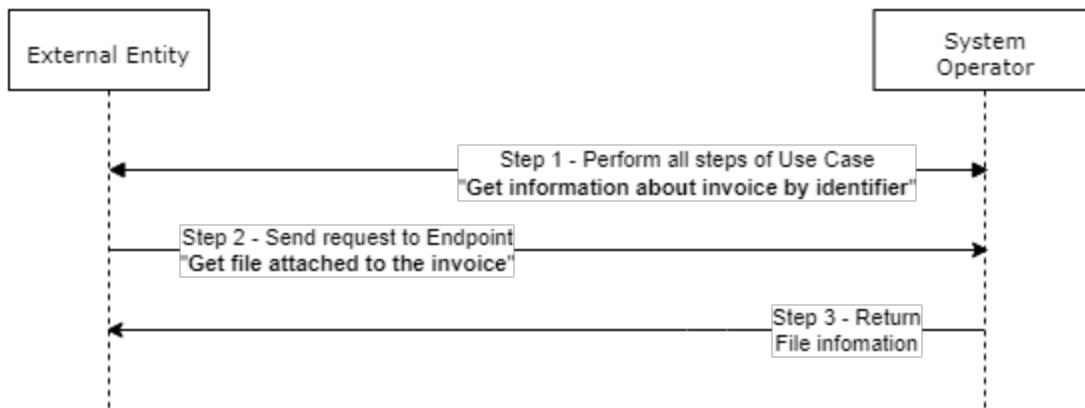
Result example

```
{  
    "file": {  
        "id": "string",  
        "ownerId": "string",  
        "mediaType": "string",  
        "name": "string",  
        "url": "string",  
        "md5": "string",  
        "sha1": "string",  
        "size": 0,  
        "used": false,  
        "createdAt": "2018-08-06T10:53:35.887Z",  
        "expiresAt": "2018-08-06T10:53:35.887Z",  
        "tag": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

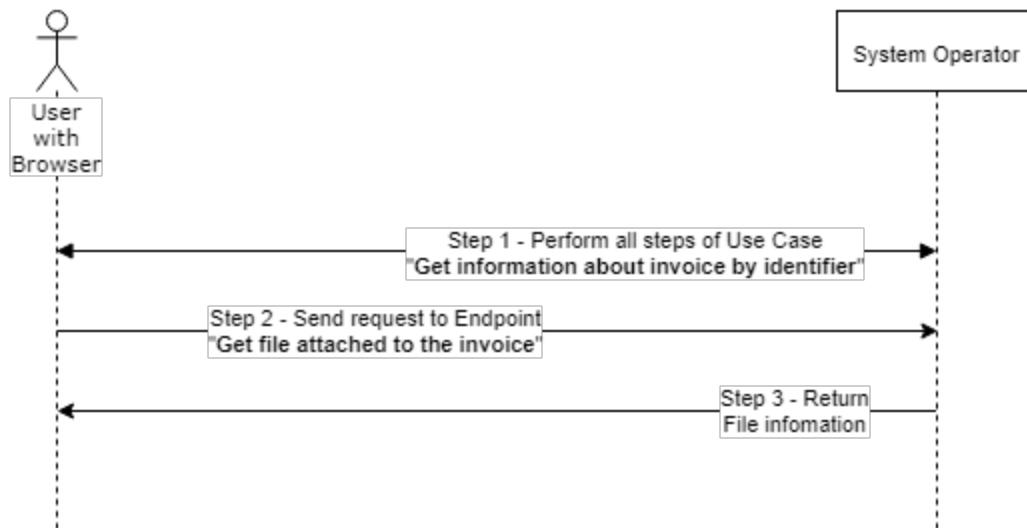
Get file attached to the invoice scheme

Use case: Get file attached to the invoice

Basic FFlow



Optional Web UI Flow



Get information about invoice by identifier description

Use Case Name

Get information about invoice by the identifier

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER, INVOICE_PAYER

will go through all steps of “View invoices” Use Case, and then send a request to Endpoint “Get information about invoice by the identifier”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, INVOICE_PAYER.

2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View invoices”.
2. External Entity sends a request to Endpoint “Get information about invoice by the identifier”.

Endpoint URL: GET /invoices/{identifier}

Parameter: Security TOKEN

3. System Operator returns Invoice information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.
2. A user sends a request to Endpoint “Get information about invoice by the identifier”.

Endpoint URL: GET /invoices/{identifier}

Parameter: Security TOKEN

3. System Operator returns Invoice information to User (See Result example below).

Post Conditions

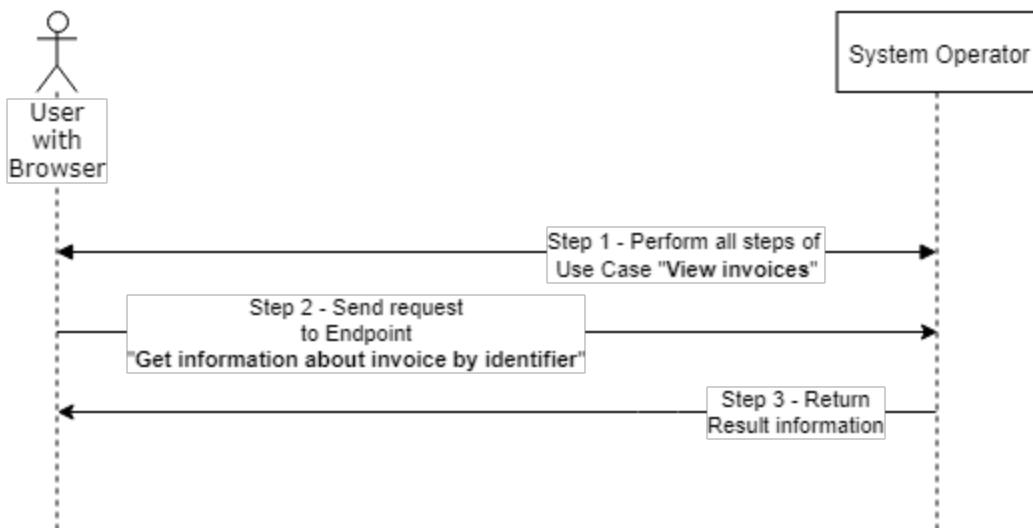
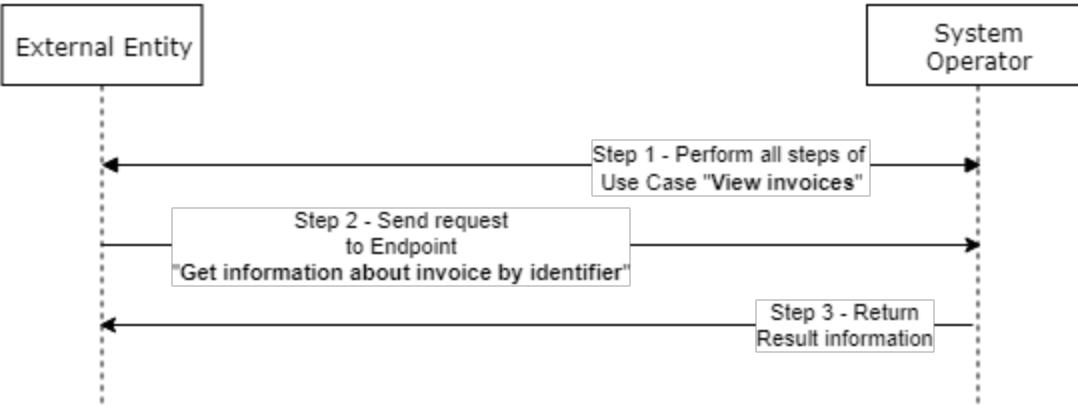
Invoice is available for review

Result example

```
{  
    "invoice": {  
        "identifier": "string",  
        "name": "string",  
        "createdAt": "2018-08-06T10:53:35.877Z",  
        "createdByUser": {  
            "id": "string",  
            "name": "string"  
        },  
        "status": "initiated",  
        "payer": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "payerContact": "string",  
        "merchantName": "string",  
        "totalPrice": 0,  
        "expiresAt": "2018-08-06T10:53:35.877Z",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "data": {  
            "productCode": "string",  
            "productPrice": 0,  
            "description": "string",  
            "count": 0,  
            "terms": "string"  
        },  
        "paymentCode": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Get information about invoice by identifier scheme

Use case: Get information about invoice by identifier



Pay for the invoice description

Use Case Name

Pay for the invoice

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER, INVOICE_PAYER will go through all steps of “View invoices” Use Case, and then send a request to Endpoint “Pay for the invoice”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, INVOICE_PAYER.

2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View invoices”.
2. External Entity sends a request to Endpoint “Pay for the invoice”.

Endpoint URL: POST /invoices/{identifier}/pay

Parameter:

```
{
  "payerCoin": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View invoices”.
2. A user sends a request to Endpoint “Pay for the invoice”.

Endpoint URL: POST /invoices/{identifier}/pay

Parameter:

```
{
  "payerCoin": "string"
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

Invoice is paid.

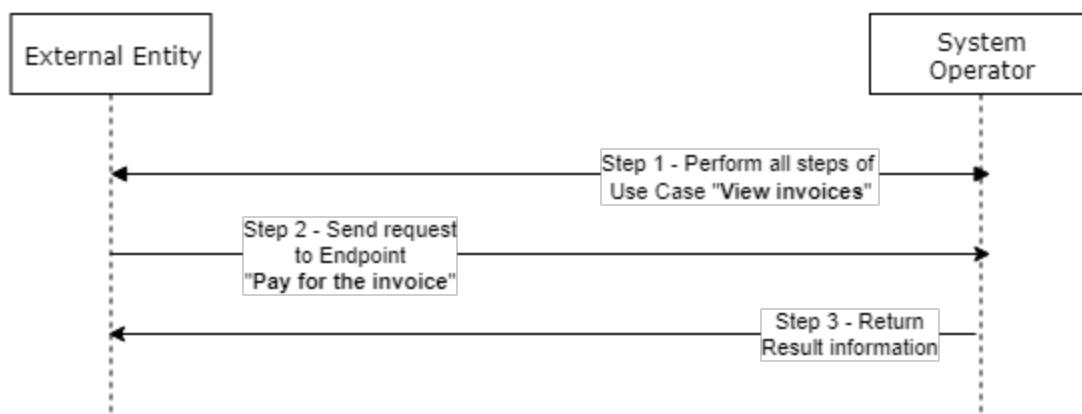
Result example

```
{  
    "invoice": {  
        "identifier": "string",  
        "name": "string",  
        "createdAt": "2018-08-06T12:12:39.970Z",  
        "createdByUser": {  
            "id": "string",  
            "name": "string"  
        },  
        "status": "initiated",  
        "payer": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "payerContact": "string",  
        "merchantName": "string",  
        "totalPrice": 0,  
        "expiresAt": "2018-08-06T12:12:39.970Z",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "data": {  
            "productCode": "string",  
            "productPrice": 0,  
            "description": "string",  
            "count": 0,  
            "terms": "string"  
        },  
        "paymentCode": "string"  
    },  
    "status": "ok",  
    "message": "string"  
}
```

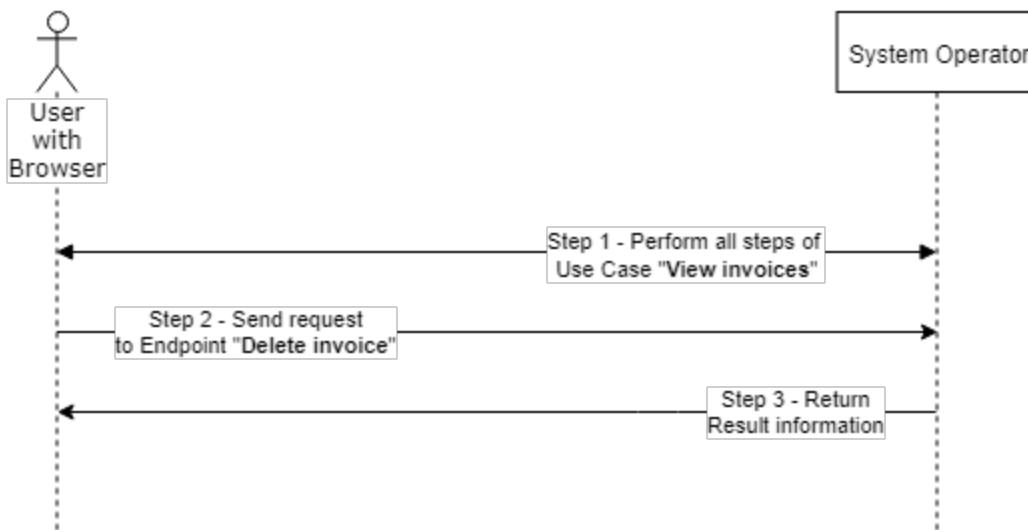
Pay for the invoice scheme

Use case: Pay for the invoice

Basic FLow



Optional Web UI Flow



View invoices description

Use Case Name

View invoices

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER, INVOICE_PAYER

will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View invoices”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, INVOICE_PAYER.

2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to Endpoint “View invoices”.

Endpoint URL: POST /invoices/view

Parameter:

```
{
  "pageNumber": 0,
  "pageSize": 0,
  "filter": {
    "statuses": [
      "initiated"
    ],
    "identifier": "string",
    "totalPrice": 0,
    "direction": "INCOMING",
    "createdByUserId": "string"
  },
  "sort": {
    "date": "asc",
    "totalPrice": "asc"
  }
}
```

3. System Operator returns a List of Invoices to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “View invoices”.

Endpoint URL: POST /invoices/view

Parameter:

```
{  
    "pageNumber": 0,  
    "pageSize": 0,  
    "filter": {  
        "statuses": [  
            "initiated"  
        ],  
        "identifier": "string",  
        "totalPrice": 0,  
        "direction": "INCOMING",  
        "createdByUserId": "string"  
    },  
    "sort": {  
        "date": "asc",  
        "totalPrice": "asc"  
    }  
}
```

3. System Operator returns a List of Invoices to User (See Result example below).

Post Conditions

List of invoices is available

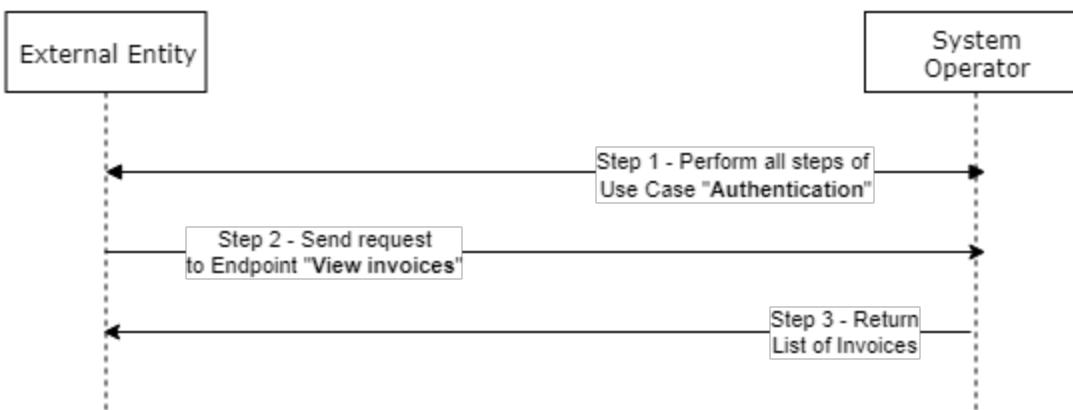
Result example

```
{  
  "records": [  
    {  
      "identifier": "string",  
      "name": "string",  
      "createdAt": "2018-08-06T10:53:35.867Z",  
      "createdByUser": {  
        "id": "string",  
        "name": "string"  
      },  
      "status": "initiated",  
      "payer": {  
        "id": "string",  
        "type": "string",  
        "name": "string"  
      },  
      "payerContact": "string",  
      "merchantName": "string",  
      "totalPrice": 0,  
      "expiresAt": "2018-08-06T10:53:35.867Z",  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "data": {  
        "productCode": "string",  
        "productPrice": 0,  
        "description": "string",  
        "count": 0,  
        "terms": "string"  
      },  
      "paymentCode": "string"  
    }  
  ]  
}
```

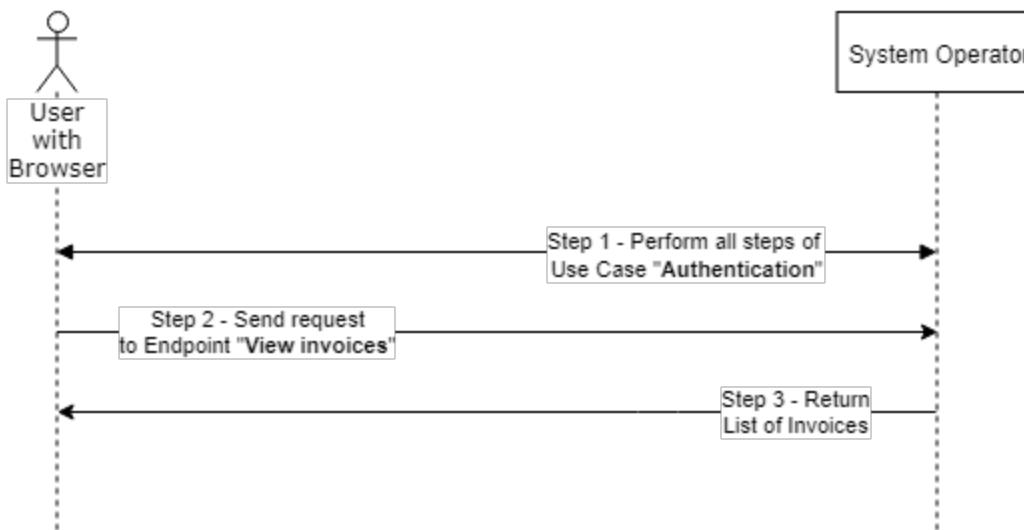
View invoices scheme

Use case: View invoices

Basic Flow



Optional Web UI Flow



Merchant Payment

Calculate commission fee description

Use Case Name

Calculate commission fee

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER, PAYMENT_PAYER will go through all steps of “Get payments” Use Case, and then send a request to Endpoint “Calculate commission fee”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER, PAYMENT_PAYER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get payments”.
2. External Entity sends a request to Endpoint “Calculate commission fee”.

Endpoint URL: POST /merchant-payments/{identifier}/calculate

Parameters: TOKEN

3. System Operator returns commission amount to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get payments”.
2. A user sends a request to Endpoint “Calculate commission fee”.

Endpoint URL: POST /merchant-payments/{identifier}/calculate

Parameters: TOKEN

3. System Operator returns commission amount to User (See Result example below).

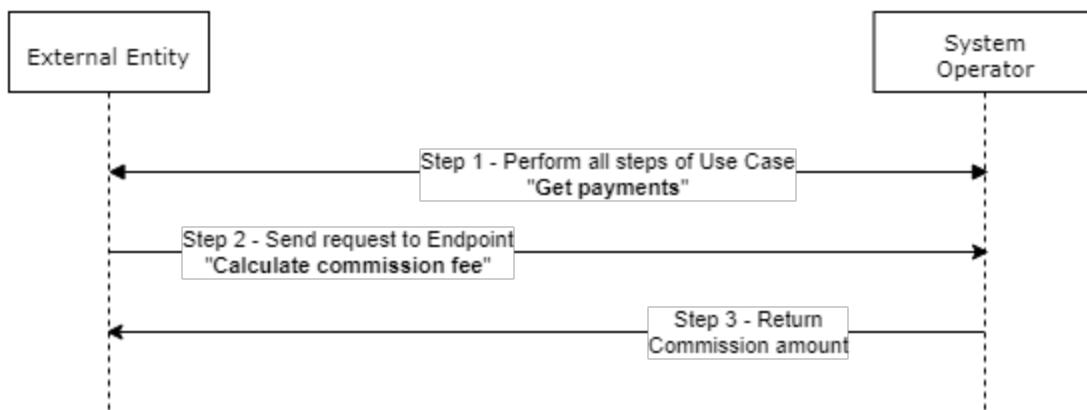
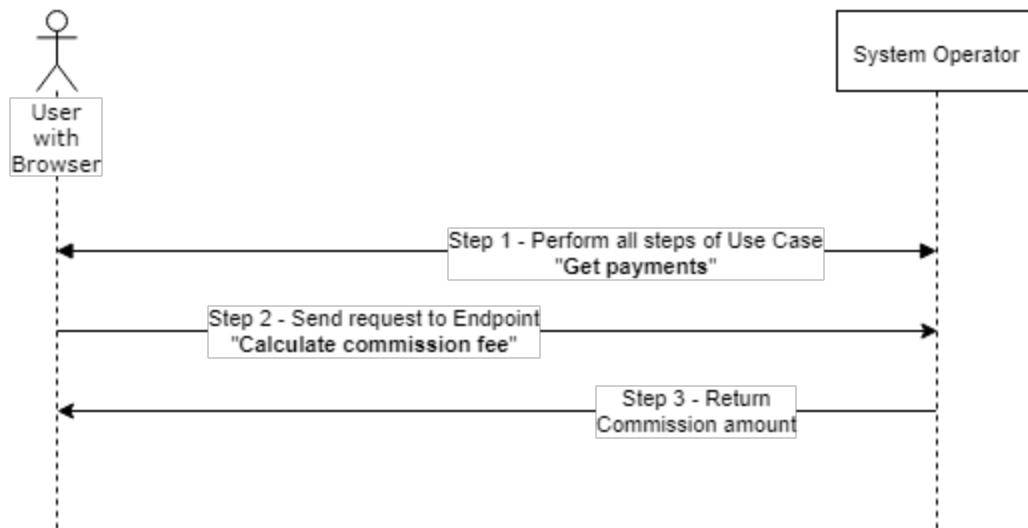
Post Conditions

Commission amount is available.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "transactionAmount": 0,  
    "senderAmountPush": 0,  
    "recipientAmountPush": 0,  
    "commissionAmountPush": 0,  
    "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
    }  
}
```

Calculate commission fee scheme

Use case: Calculate commission fee**Basic FFlow****Optional Web UI Flow**

Create a payment description

Use Case Name

Create a payment

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Create a payment”.

A payment is different from an invoice. After creation, an invoice shows up in Coin owner cabinet.

The payment is created by the merchant at the Point of Sale terminal. All information needed to submit a request to the Endpoint “Create a payment” is available for the merchant, therefore there are no additional Use cases to execute except “Authentication”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Create a payment”.

Endpoint URL: POST /merchant-payments

Parameters:

```
{
    "pointOfSaleId": "string",
    "amount": 0,
    "issuerId": "string",
    "paymentInstrument": "COIN",
    "description": "string",
    "orderId": "string",
    "payer": {
        "method": "none",
        "value": "string",
        "confirmation": "NONE"
    },
    "productsPolicy": "NO_PRODUCTS",
    "purchases": [
        {
            "product": {
                "identifier": 0,
                "externalCode": "string"
            },
            "measureUnit": {
                "identifier": 0,
                "externalCode": "string"
            },
            "quantity": 0,
            "price": 0
        }
    ],
    "urls": {
        "callbackUrl": "string",
        "successRedirectUrl": "string"
    }
}
```

3. System Operator returns newly created payment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a payment”.

Endpoint URL: POST /merchant-payments

Parameters:

```
{  
    "pointOfSaleId": "string",  
    "amount": 0,  
    "issuerId": "string",  
    "paymentInstrument": "COIN",  
    "description": "string",  
    "orderId": "string",  
    "payer": {  
        "method": "none",  
        "value": "string",  
        "confirmation": "NONE"  
    },  
    "productsPolicy": "NO_PRODUCTS",  
    "purchases": [  
        {  
            "product": {  
                "identifier": 0,  
                "externalCode": "string"  
            },  
            "measureUnit": {  
                "identifier": 0,  
                "externalCode": "string"  
            },  
            "quantity": 0,  
            "price": 0  
        }  
    ],  
    "urls": {  
        "callbackUrl": "string",  
        "successRedirectUrl": "string"  
    }  
}
```

3. System Operator returns newly created payment information to User (See Result example below).

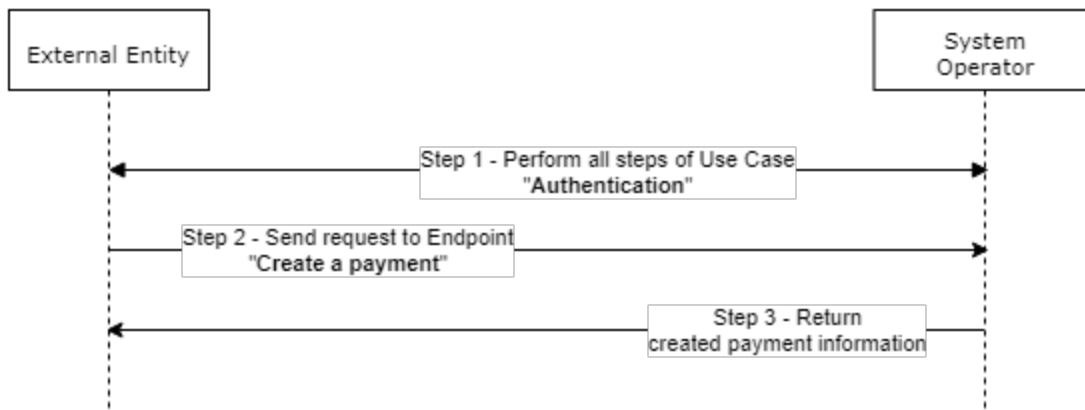
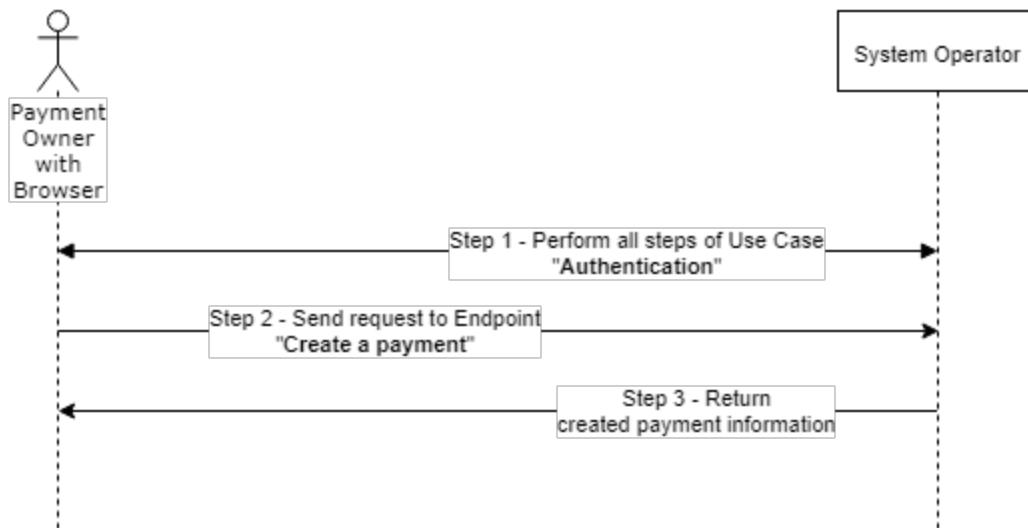
Post Conditions

New payment is available.

Result example

```
{  
    "payment": {  
        "identifier": 0,  
        "pos": {  
            "id": "string",  
            "name": "string",  
            "description": "string",  
            "website": "string",  
            "active": false  
        },  
        "parameters": {  
            "apiType": "v1",  
            "invoiceAmount": 0,  
            "orderId": "string",  
            "orderDescription": "string",  
            "paymentInstrument": "COIN"  
        },  
        "date": "2018-08-23T07:21:20.733Z",  
        "createdByUser": {  
            "id": "string",  
            "name": "string"  
        },  
        "status": "pending",  
        "payer": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "urls": {  
            "callbackUrl": "string",  
            "successRedirectUrl": "string"  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Create a payment scheme

Use case: Create a payment**Basic FFlow****Optional Web UI Flow**

Delete a pending payment description

Use Case Name

Delete a pending payment

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER will go through all steps of “Get payments” Use Case, and then send a request to Endpoint “Delete a pending payment”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get payments”.
2. External Entity sends a request to Endpoint “Delete a pending payment”.

Endpoint URL: DELETE /merchant-payments/{identifier}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get payments”.
2. A user sends a request to Endpoint “Delete a pending payment”.

Endpoint URL: DELETE /merchant-payments/{identifier}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

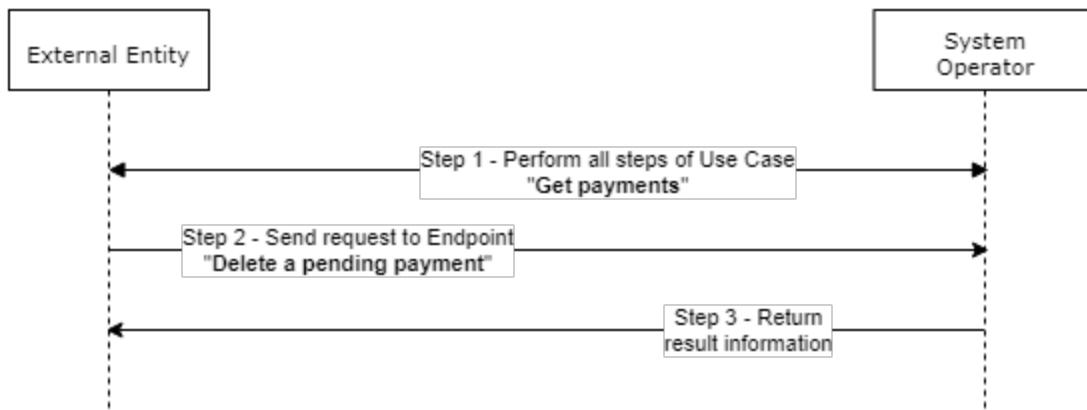
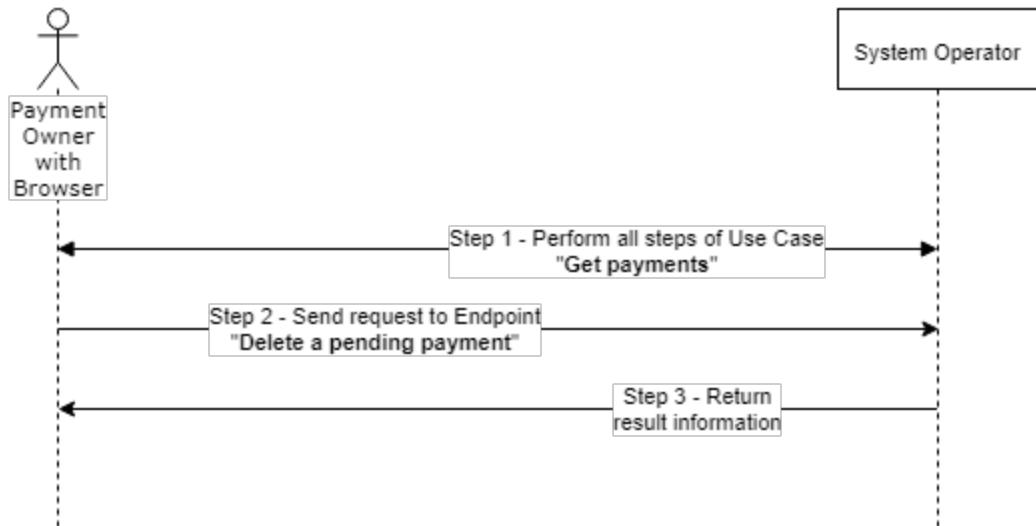
Post Conditions

Payment is deleted.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete a pending payment scheme

Use case: Delete a pending payment**Basic FFlow****Optional Web UI Flow**

Get payments description

Use Case Name

Get payments

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get payments”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to Endpoint “Get payments”.

Endpoint URL: POST /merchant-payments/view

Parameters:

```
{
    "pageNumber": 0,
    "pageSize": 0,
    "filter": {
        "posId": "string",
        "statuses": [
            "pending"
        ],
        "identifier": 0,
        "invoiceAmount": 0,
        "orderId": "string",
        "createdById": "string"
    },
    "sort": {
        "date": "asc",
        "invoiceAmount": "asc"
    }
}
```

3. System Operator returns List of payments to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get payments”.

Endpoint URL: POST /merchant-payments/view

Parameters:

```
{  
    "pageNumber": 0,  
    "pageSize": 0,  
    "filter": {  
        "posId": "string",  
        "statuses": [  
            "pending"  
        ],  
        "identifier": 0,  
        "invoiceAmount": 0,  
        "orderId": "string",  
        "createdByUserId": "string"  
    },  
    "sort": {  
        "date": "asc",  
        "invoiceAmount": "asc"  
    }  
}
```

3. System Operator returns List of payments to User (See Result example below).

Post Conditions

List of payments is available.

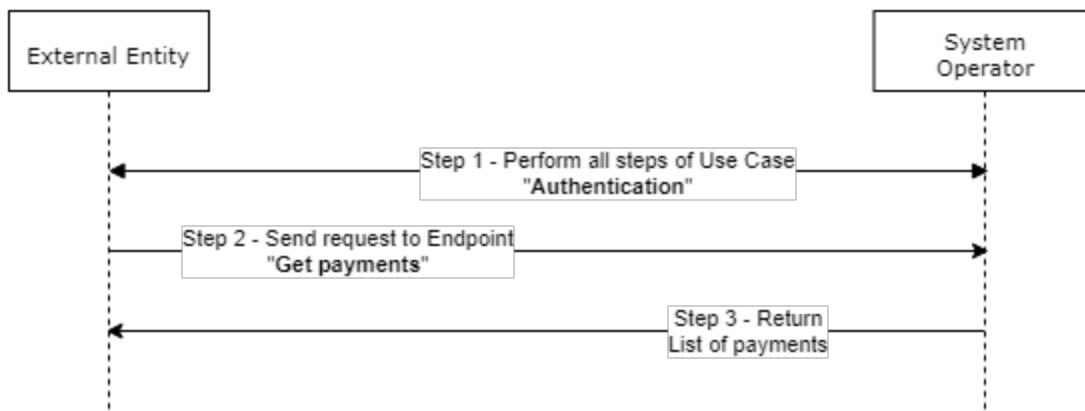
Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0,  
    "records": [  
        {  
            "identifier": 0,  
            "pos": {  
                "id": "string",  
                "name": "string",  
                "description": "string",  
                "website": "string",  
                "active": false  
            },  
            "parameters": {  
                "apiType": "v1",  
                "invoiceAmount": 0,  
                "orderId": "string",  
                "orderDescription": "string",  
                "paymentInstrument": "COIN"  
            },  
            "date": "2018-08-23T07:21:20.739Z",  
            "createdByUser": {  
                "id": "string",  
                "name": "string"  
            },  
            "status": "pending",  
            "payer": {  
                "id": "string",  
                "type": "string",  
                "name": "string"  
            },  
            "issuer": {  
                "id": "string",  
                "sn": "string",  
                "currency": "string"  
            },  
            "urls": {  
                "callbackUrl": "string",  
                "successRedirectUrl": "string"  
            }  
        }  
    ]  
}
```

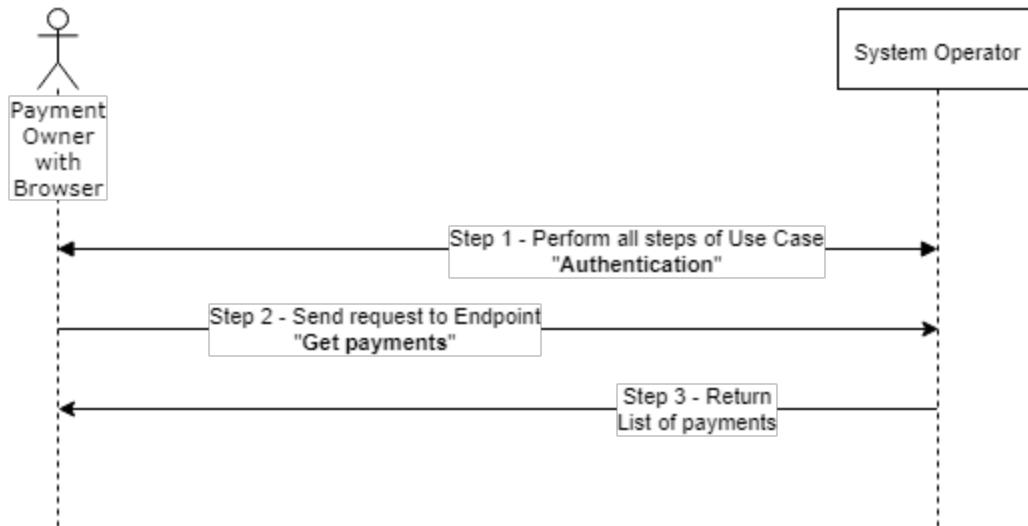
Get payments scheme

Use case: Get payments

Basic FFlow



Optional Web UI Flow



Get purchases completed within payment description

Use Case Name

Get purchases completed within the payment

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER, PAYMENT_PAYER

will go through all steps of “Get payments” Use Case, and then send a request to Endpoint “Get purchases completed within payment”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER, PAYMENT_PAYER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get payments”.
2. External Entity sends a request to Endpoint “Get purchases completed within payment”.

Endpoint URL: GET /merchant-payments/{identifier}/merchant-purchases

Parameters: TOKEN

3. System Operator returns List of purchases to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get payments”.
2. A user sends a request to Endpoint “Get purchases completed within payment”.

Endpoint URL: GET /merchant-payments/{identifier}/merchant-purchases

Parameters: TOKEN

3. System Operator returns List of purchases to User (See Result example below).

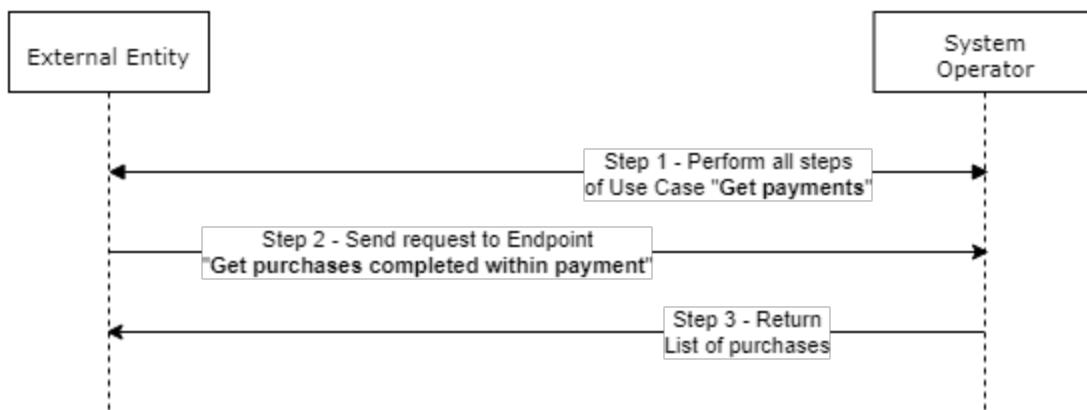
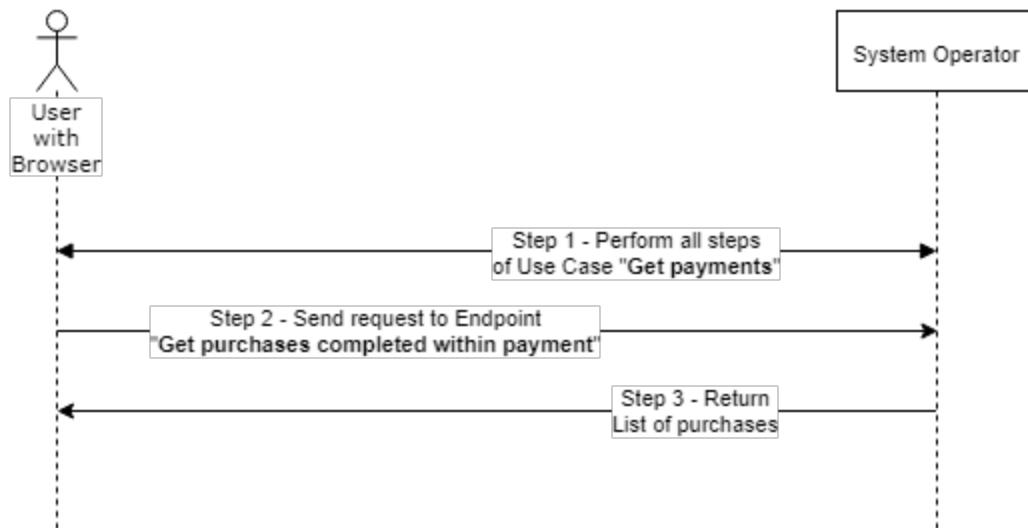
Post Conditions

List of purchases is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "product": {  
                "id": 0,  
                "externalCode": "string",  
                "name": "string",  
                "description": "string",  
                "measureUnit": {  
                    "id": 0,  
                    "externalCode": "string",  
                    "code": "string",  
                    "description": "string"  
                }  
            },  
            "quantity": 0,  
            "measureUnit": {  
                "id": 0,  
                "externalCode": "string",  
                "code": "string",  
                "description": "string"  
            },  
            "price": 0,  
            "totalAmount": 0  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Get purchases completed within payment scheme

Use case: Get purchases completed within payment**Basic FFlow****Optional Web UI Flow**

Pay for payment description

Use Case Name

Pay for payment

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER, PAYMENT_PAYER will go through all steps of “Get payments” Use Case, and then send a request to Endpoint “Pay for payment”.

The request parameters in this Endpoint are described as follows.

OTP - One-time password sent to user's contact via email or SMS message.

Mandatory for payments with OTP_PHONE confirmation.

PIN - Payer's master PIN. Mandatory for payments with MASTER_PIN confirmation.

OTP and pin are mutually exclusive.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER, PAYMENT_PAYER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get payments”.
2. External Entity sends a request to Endpoint “Pay for payment”.

Endpoint URL: POST /merchant-payments/{identifier}/execute

Parameters:

```
{
  "otp": "string",
  "pin": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get payments”.
2. A user sends a request to Endpoint “Pay for payment”.

Endpoint URL: POST /merchant-payments/{identifier}/execute

Parameters:

```
{
  "otp": "string",
  "pin": "string"
}
```

3. System Operator returns result information to User (See Result example below).

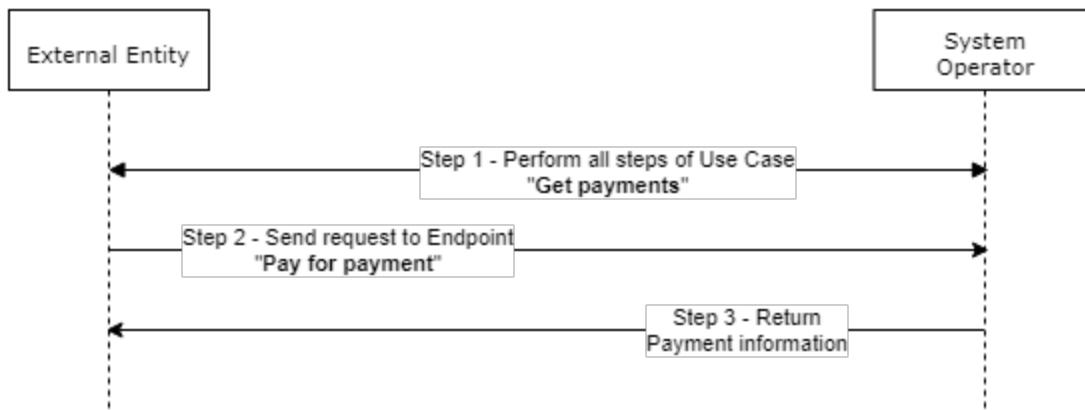
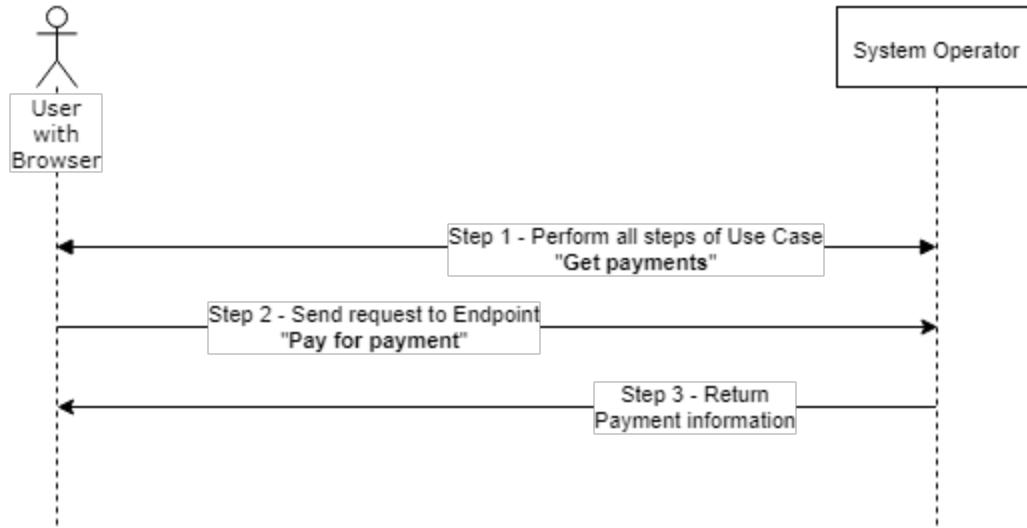
Post Conditions

The payment processed.

Result example

```
{  
    "payment": {  
        "identifier": 0,  
        "pos": {  
            "id": "string",  
            "name": "string",  
            "description": "string",  
            "website": "string",  
            "active": false  
        },  
        "parameters": {  
            "apiType": "v1",  
            "invoiceAmount": 0,  
            "orderId": "string",  
            "orderDescription": "string",  
            "paymentInstrument": "COIN"  
        },  
        "date": "2018-08-27T13:53:17.122Z",  
        "createdByUser": {  
            "id": "string",  
            "name": "string"  
        },  
        "status": "pending",  
        "payer": {  
            "id": "string",  
            "type": "string",  
            "name": "string"  
        },  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "urls": {  
            "callbackUrl": "string",  
            "successRedirectUrl": "string"  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Pay for payment scheme

Use case: Pay for payment**Basic FFlow****Optional Web UI Flow**

Resend one time password description

Use Case Name

Resend one-time password

Brief Description

A User or External Entity on behalf of a User with role permission PAYMENT_OWNER will go through all steps of “Get payments” Use Case, and then send a request to Endpoint “Resend one-time password”.

This Use Case is executed when there was a problem during “Pay for payment”. As a result of this “Get payments” Use Case a person who is trying to make a payment gets a new one time password via email to the address or SMS to the phone number specified in the profile.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: PAYMENT_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get payments”.
2. External Entity sends a request to Endpoint “Resend one-time password”.

Endpoint URL: POST /merchant-payments/{identifier}/resend-otp

Parameters: TOKEN

3. System Operator creates new OTP, sends it to email or SMS and returns result information to User (See Result example below).

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get payments”.
2. A user sends a request to Endpoint “Resend one-time password”.

Endpoint URL: POST /merchant-payments/{identifier}/resend-otp

Parameters: TOKEN

3. System Operator creates new OTP, sends it to email or SMS and returns result information to User (See Result example below).

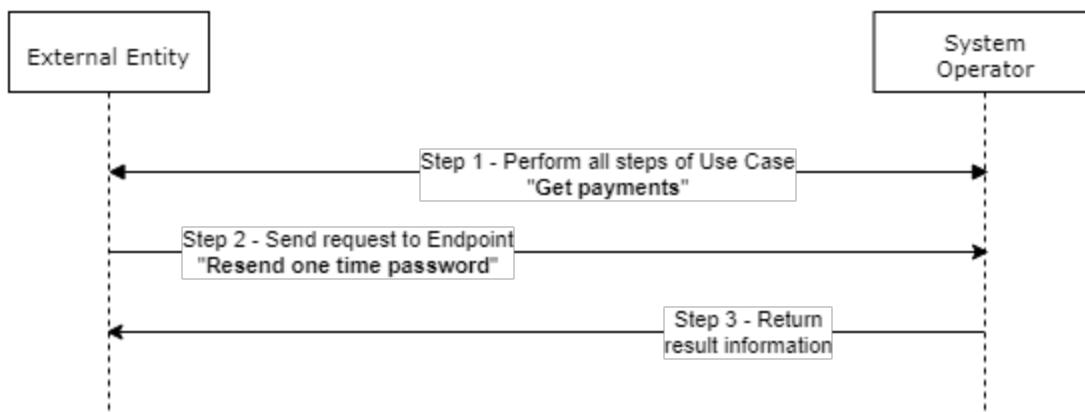
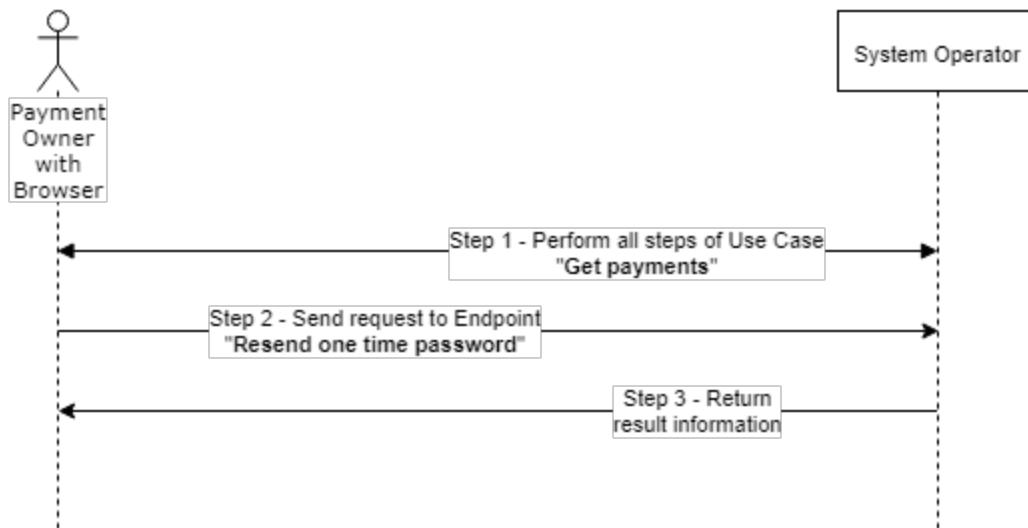
Post Conditions

New OTP is available for user via email or SMS.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Resend one time password scheme

Use case: Resend one time password**Basic FFlow****Optional Web UI Flow**

Merchant payment via external payment system

Create gate transaction for merchant payment description

Use Case Name

Create gate transaction for merchant payment

Brief Description

A User or External Entity on behalf of a User with role permission "GATE_OPERATION_EXECUTOR" will go through all steps of "Get points of sale" (to obtain PoS ID) and "Get merchant payment transactions for specified POS" (to obtain transaction ID) Use Cases, and then send a request to Endpoint "Create gate transaction for merchant payment".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "GATE_OPERATION_EXECUTOR", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get points of sale".
2. Perform all steps of Use Case "Get merchant payment transactions for specified POS".
3. External Entity sends a request to Endpoint "Create gate transaction for merchant payment".

Endpoint URL: POST /merchant-payments/{identifier}/gate/transactions

Parameters: {

```
"deviceId": "string",
"deviceOrderId": "string",
"method": {
    "accountId": "string",
    "way": "string"
}
```

1. System Operator returns information about a new gate transaction for a merchant payment to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get points of sale".

2. Perform all steps of Use Case “Get merchant payment transactions for specified POS”.
3. A user sends a request to Endpoint “Create gate transaction for merchant payment”.

Endpoint URL: POST /merchant-payments/{identifier}/gate/transactions

Parameters: {

```
"deviceId": "string",
"deviceOrderId": "string",
"method": {
  "accountId": "string",
  "way": "string"
}
}
```

1. System Operator returns information about a new gate transaction for a merchant payment to User (See Result example below).

Post Conditions

Merchant payment transactions for a specified POS is available.

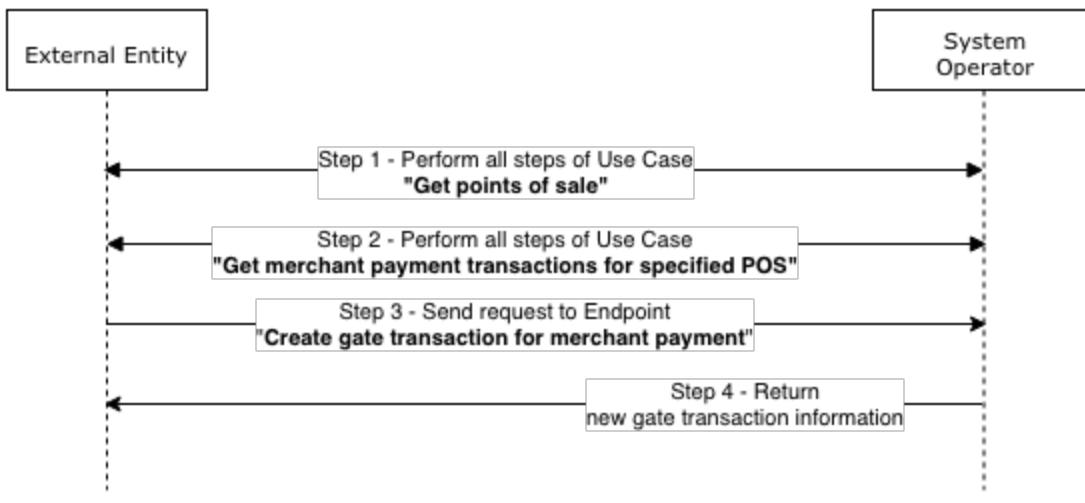
Result example

```
{
  "transaction": {
    "id": "string",
    "orderId": 0,
    "deviceId": "string",
    "deviceOrderId": "string",
    "type": "TOPUP",
    "status": "INITIATED",
    "errorCode": "UNKNOWN",
    "coin": {
      "serial": "string",
      "name": "string",
      "amount": 0,
      "availableAmount": 0,
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "active": false,
      "type": "regular_commission"
    }
  }
}
```

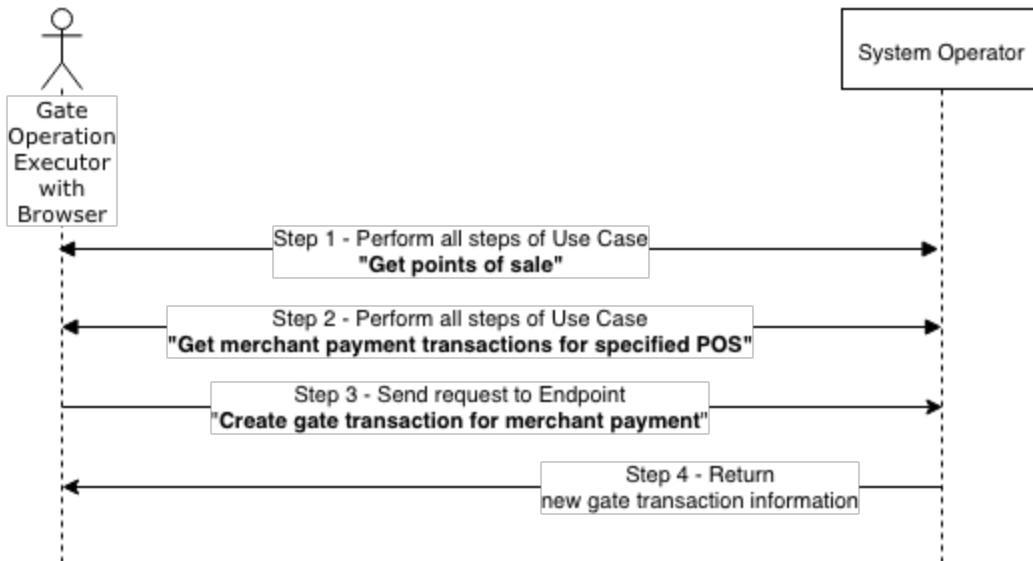
```
 },
 "paymentMethod": {
   "accountId": "string",
   "account": {
     "id": "string",
     "provider": {
       "name": "string"
     }
   },
   "way": "string"
 },
 "sourceAmount": 0,
 "amountToSend": 0,
 "finalAmount": 0,
 "processId": "string",
 "payerData": {}
},
 "status": "ok",
 "message": "string"
}
Create gate transaction for merchant payment scheme
```

Use case: Create gate transaction for merchant payment

Basic FFlow



Optional Web UI Flow



Templates

Add new purchase template to a cart template description

Use Case Name

Add new purchase template to a cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Add new purchase template to a cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Add new purchase template to a cart template”.

Endpoint URL: POST /templates/cart/{id}/purchase

Parameters:

```
{
  "purchase template name": "string",
  "productId": "string",
  "coinSerial": "string",
  "amount": 0,
  "description": "string",
  "optionName": "default",
  "fields": [
    {
      "name": "string",
      "value": {
        "name": "string"
      }
    }
  ],
  "templateDto": {
    "name": "string",
    "productId": "string",
    "coin": "string",
    "amount": 0,
    "description": "string",
    "optionName": "string",
    "payerFields": {
      "name": "string"
    }
  }
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.
2. A user sends a request to Endpoint “Add new purchase template to a cart template”.

Endpoint URL: POST /templates/cart/{id}/purchase

Parameters:

```
{
    "purchaseTemplate": {
        "name": "string",
        "productId": "string",
        "coinSerial": "string",
        "amount": 0,
        "description": "string",
        "optionName": "default",
        "fields": [
            {
                "name": "string",
                "value": {
                    ...
                }
            }
        ],
        "templateDto": {
            "name": "string",
            "productId": "string",
            "coin": "string",
            "amount": 0,
            "description": "string",
            "optionName": "string",
            "payerFields": {
                ...
            }
        }
    }
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

New purchase template is available.

Result example

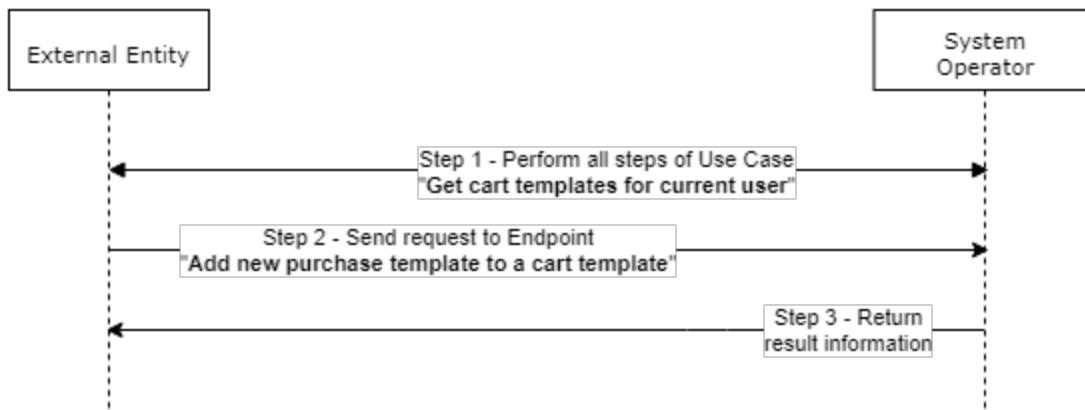
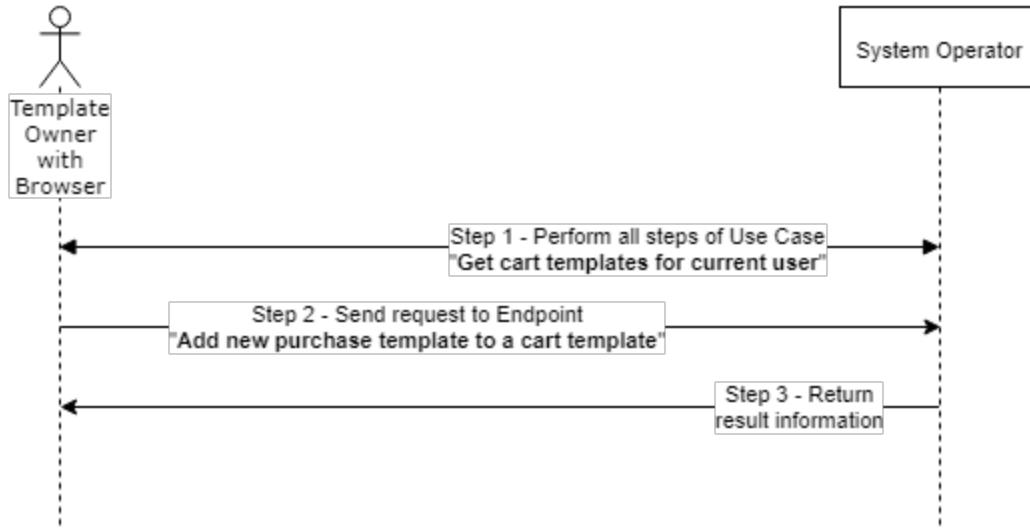
```
{
    "cartTemplates": [
        {
            "templates": [
                {
                    "productDto": {
                        "id": "string",
                        "name": "string",
                        "description": "string",
                        "category": "MOBILE",
                        "icon": "string",
                        "accountId": "string",
                        ...
                    }
                }
            ]
        }
    ]
}
```

```
"currencies": [
    {
        "currency": {
            "code": "string",
            "digitalCode": "string",
            "symbol": "string",
            "name": "string",
            "description": "string"
        },
        "minAmount": 0,
        "maxAmount": 0
    }
],
},
"optionName": "string",
"coin": {
    "serial": "string",
    "name": "string",
    "amount": 0,
    "availableAmount": 0,
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    },
    "active": false,
    "type": "regular_commission"
},
"id": "string",
"type": "TRANSFER",
"name": "string",
"amount": 0,
"description": "string"
},
],
"id": "string",
"type": "TRANSFER",
"name": "string",
"amount": 0,
"description": "string"
}
],
```

```
"status": "ok",
"message": "string"
}
```

Add new purchase template to a cart template scheme

Use case: Add new purchase template to a cart template

Basic FLow**Optional Web UI Flow**

Create a cart template description

Use Case Name

Create a cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Create a cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Create a cart template”.

Endpoint URL: POST /templates/cart

Parameters:

```
{  
    "name": "string",  
    "description": "string",  
    "regular": false,  
    "templates": [  
        {  
            "name": "string",  
            "productId": "string",  
            "coinSerial": "string",  
            "amount": 0,  
            "description": "string",  
            "optionName": "default",  
            "fields": [  
                {  
                    "name": "string",  
                    "value": {  
  
                }  
            }  
        },  
        {  
            "templateDto": {  
                "name": "string",  
                "productId": "string",  
                "coin": "string",  
                "amount": 0,  
                "description": "string",  
                "optionName": "string",  
                "payerFields": {  
  
                }  
            }  
        }  
    ],  
    "templateDtos": [  
        {  
            "name": "string",  
            "productId": "string",  
            "coin": "string",  
            "amount": 0,  
            "description": "string",  
            "optionName": "string",  
            "payerFields": {  
  
            }  
        }  
    ]  
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.

2. A user sends a request to Endpoint “Create a cart template”.

Endpoint URL: POST /templates/cart

Parameters: same as in Basic Flow

3. System Operator returns result information to User (See Result example below).

Post Conditions

The new template is available for usage.

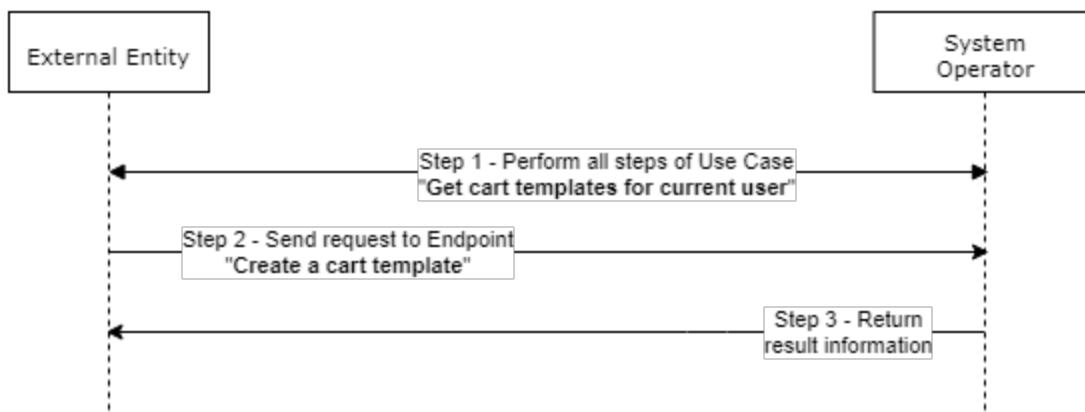
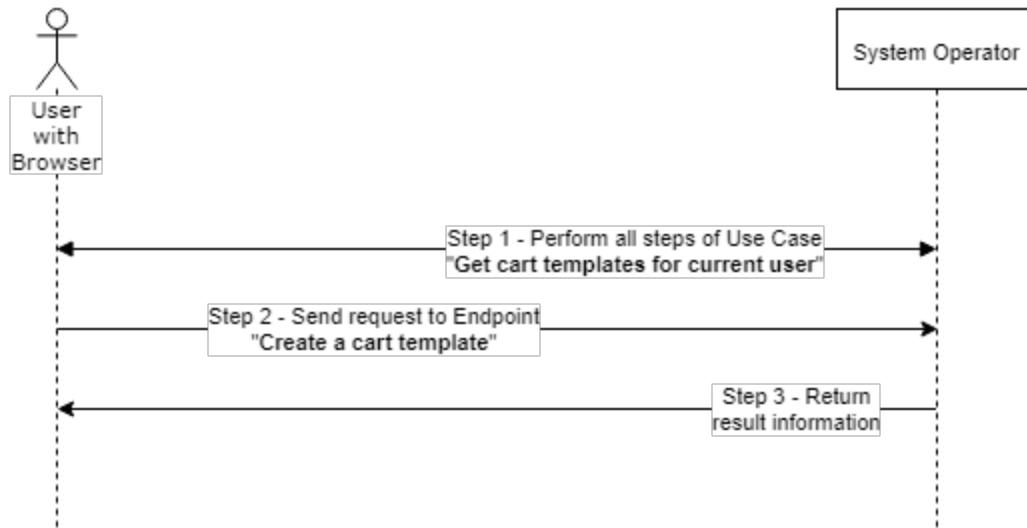
Result example

```
{
  "id": "string",
  "name": "string",
  "amount": 0,
  "description": "string",
  "regular": false,
  "templates": [
    {
      "productDto": {
        "id": "string",
        "name": "string",
        "description": "string",
        "category": "MOBILE",
        "icon": "string",
        "accountId": "string",
        "currencies": [
          {
            "currency": {
              "code": "string",
              "digitalCode": "string",
              "symbol": "string",
              "name": "string",
              "description": "string"
            },
            "minAmount": 0,
            "maxAmount": 0
          }
        ]
      },
      "optionName": "string",
      "coin": {
        "serial": "string",
        "name": "string",
        "amount": 0,
        "value": 0
      }
    }
  ]
}
```

```
        "availableAmount":0,
        "issuer":{
            "id":"string",
            "sn":"string",
            "currency":"string"
        },
        "active":false,
        "type":"regular_commission"
    },
    "id":"string",
    "type":"TRANSFER",
    "name":"string",
    "amount":0,
    "description":"string"
}
],
]
```

```
"status": "ok",
"message": "string"
}
```

Create a cart template scheme

Use case: Create a cart template**Basic FFlow****Optional Web UI Flow**

Create a transfer template description

Use Case Name

Create a transfer template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get transfer templates for current user” Use Case, and then send a request to Endpoint “Create a transfer template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. External Entity sends a request to Endpoint “Create a transfer template”.

Endpoint URL: POST /templates/transfer

Parameters:

```
{
  "name": "string",
  "recipientCoin": "string",
  "senderCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. A user sends a request to Endpoint “Create a transfer template”.

Endpoint URL: POST /templates/transfer

Parameters:

```
{
  "name": "string",
  "recipientCoin": "string",
  "senderCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator returns result information to User (See Result example below).

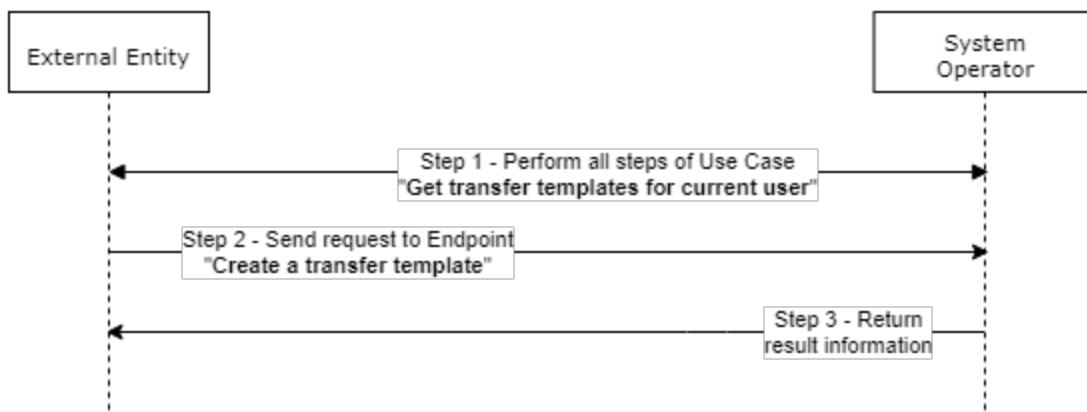
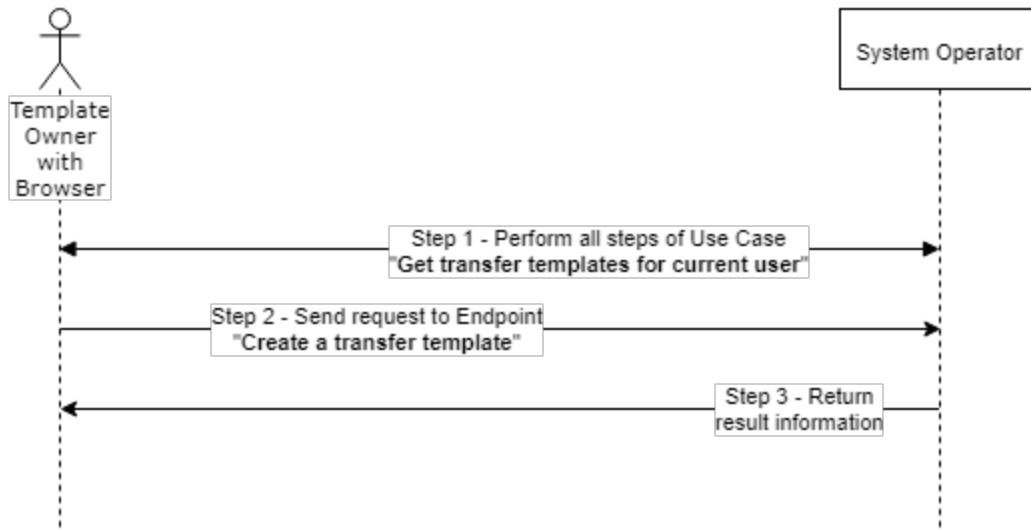
Post Conditions

The transfer template is available.

Result example

```
{  
    "id": "string",  
    "name": "string",  
    "senderCoin": {  
        "serial": "string",  
        "name": "string",  
        "amount": 0,  
        "availableAmount": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "recipientCoin": {  
        "serial": "string",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "recipientPhoneNumber": "string",  
    "amount": 0,  
    "description": "string",  
    "status": "ok",  
    "message": "string"  
}
```

Create a transfer template scheme

Use case: Create a transfer template**Basic FLow****Optional Web UI Flow**

Delete a transfer template description

Use Case Name

Delete a transfer template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get transfer templates for current user” Use Case, and then send a request to Endpoint “Delete a transfer template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. External Entity sends a request to Endpoint “Delete a transfer template”.

Endpoint URL: DELETE /templates/transfer/{id}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. A user sends a request to Endpoint “Delete a transfer template”.

Endpoint URL: DELETE /templates/transfer/{id}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

Transfer template is deleted..

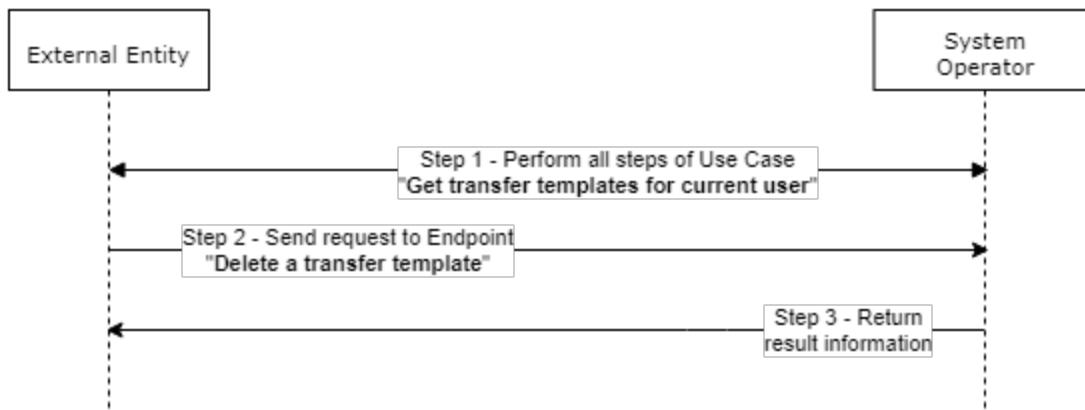
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

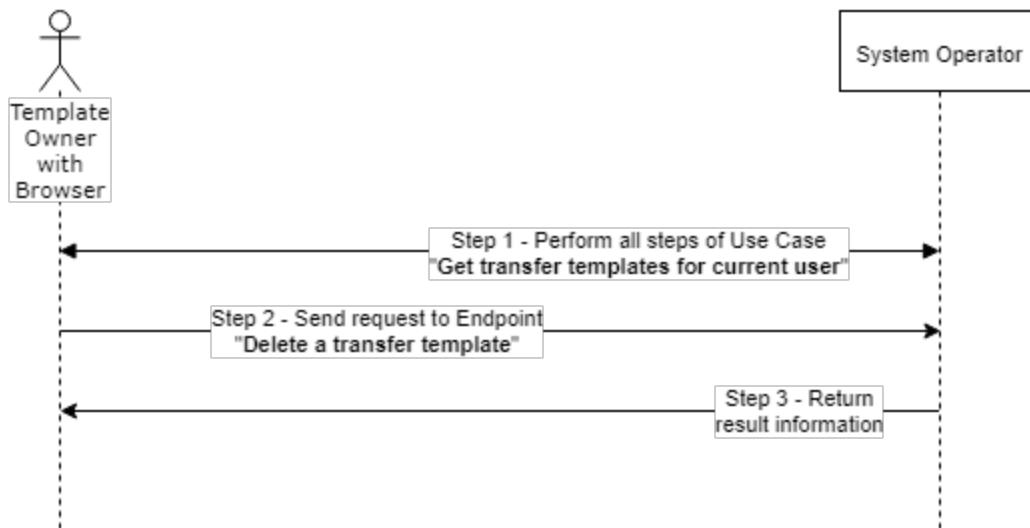
Delete a transfer template scheme

Use case: Delete a transfer template

Basic FFlow



Optional Web UI Flow



Delete cart template description

Use Case Name

Delete cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Delete cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Delete cart template”.

Endpoint URL: DELETE /templates/cart/{templateId}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.
2. A user sends a request to Endpoint “Delete cart template”.

Endpoint URL: DELETE /templates/cart/{templateId}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

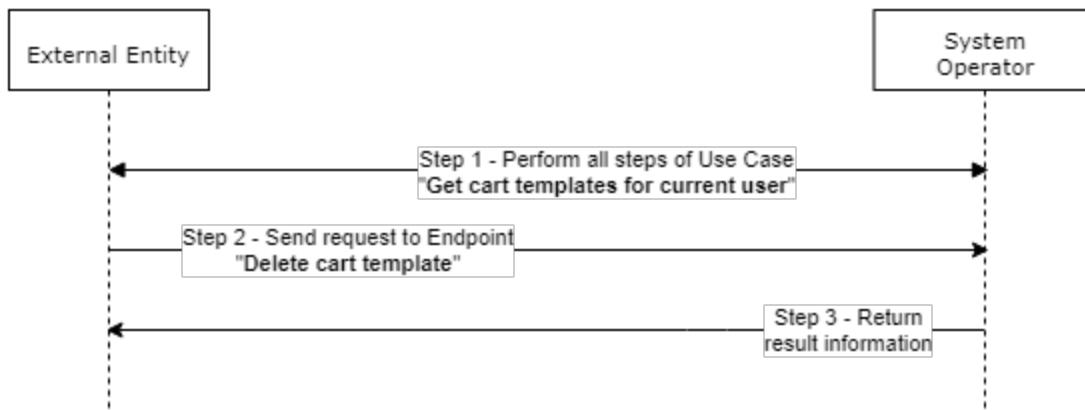
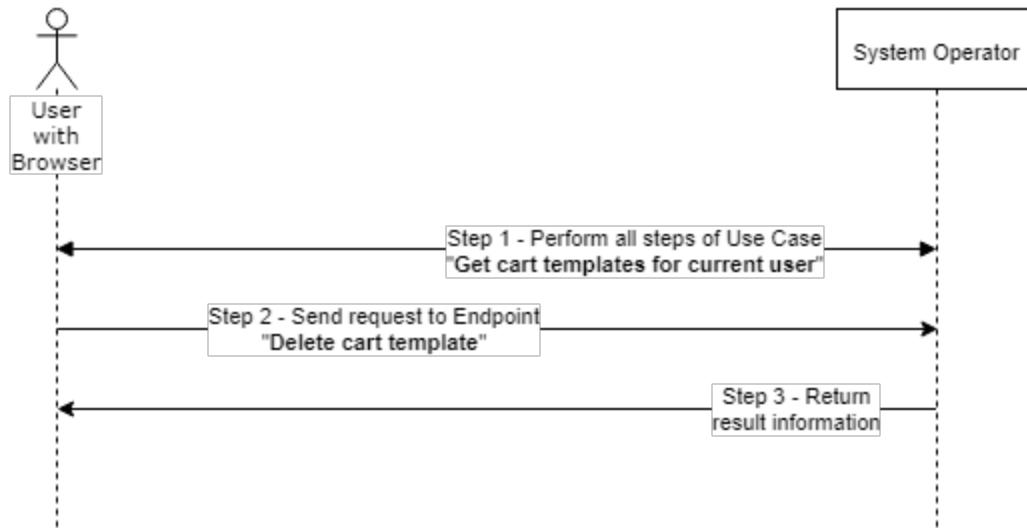
Post Conditions

Cart template is deleted..

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete cart template scheme

Use case: Delete cart template**Basic FFlow****Optional Web UI Flow**

Delete purchase template from a cart template description

Use Case Name

Delete purchase template from a cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Delete purchase template from a cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Delete purchase template from a cart template”.

Endpoint URL: DELETE /templates/cart/{cartId}/purchase/{templateId}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.
2. A user sends a request to Endpoint “Delete purchase template from a cart template”.

Endpoint URL: DELETE /templates/cart/{cartId}/purchase/{templateId}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

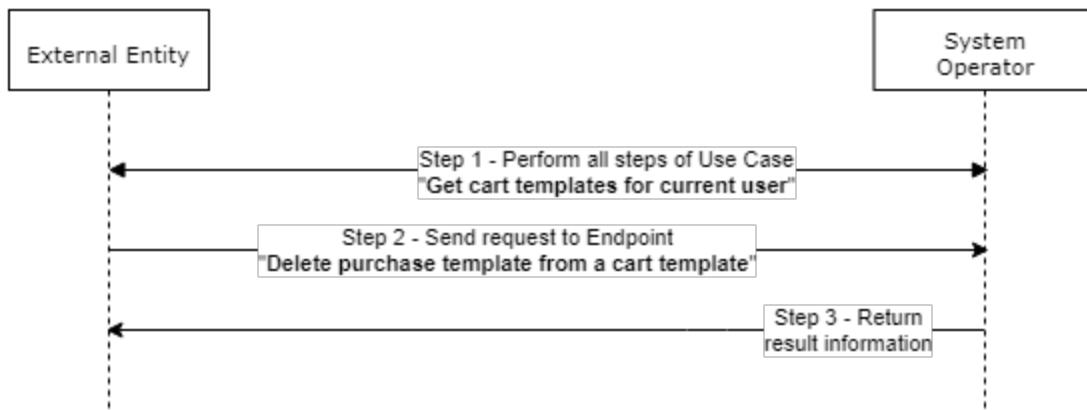
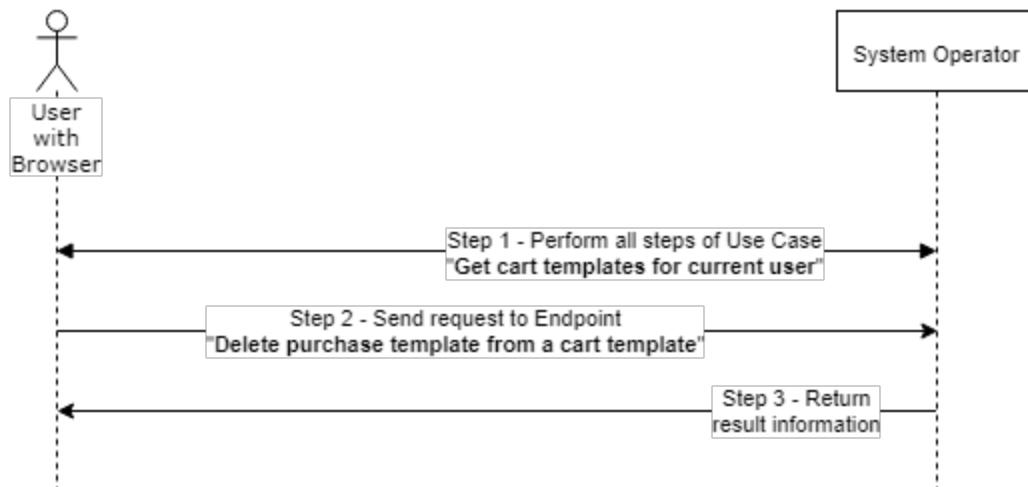
Post Conditions

purchase template is deleted.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete purchase template from a cart template scheme

Use case: Delete purchase template from a cart template**Basic FLow****Optional Web UI Flow**

Get cart templates for current user description

Use Case Name

Get cart templates for current user

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get cart templates for current user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get cart templates for current user”.

Endpoint URL: GET /templates/cart

Parameters: TOKEN

3. System Operator returns List of Templates to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get cart templates for current user”.

Endpoint URL: GET /templates/cart

Parameters: TOKEN

3. System Operator returns List of Templates to User (See Result example below).

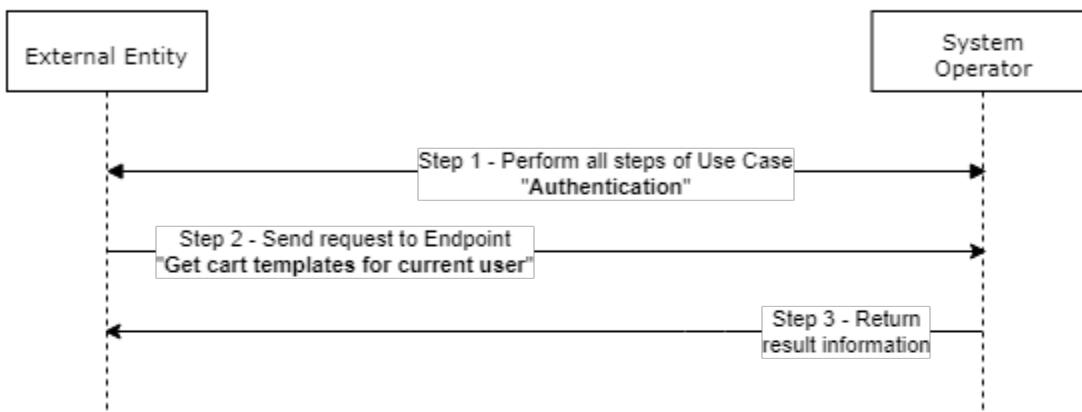
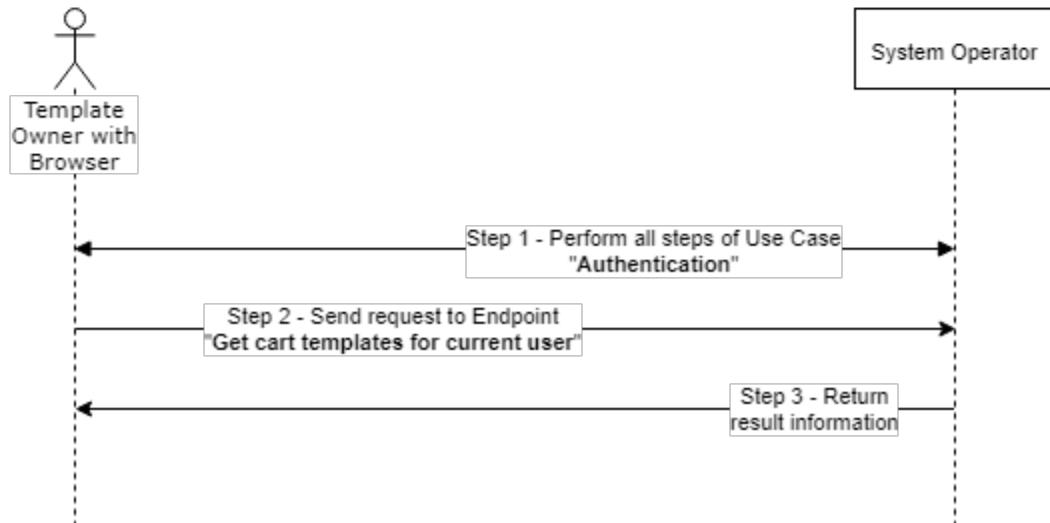
Post Conditions

List of templates is available.

Result example

```
{  
    "templates": [  
        {  
            "productDto": {  
                "id": "string",  
                "name": "string",  
                "description": "string",  
                "category": "MOBILE",  
                "icon": "string",  
                "accountId": "string",  
                "currencies": [  
                    {  
                        "currency": {  
                            "code": "string",  
                            "digitalCode": "string",  
                            "symbol": "string",  
                            "name": "string",  
                            "description": "string"  
                        },  
                        "minAmount": 0,  
                        "maxAmount": 0  
                    }  
                ]  
            },  
            "optionName": "string",  
            "coin": {  
                "serial": "string",  
                "name": "string",  
                "amount": 0,  
                "availableAmount": 0,  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "id": "string",  
            "type": "TRANSFER",  
            "name": "string",  
            "amount": 0,  
            "description": "string"  
        }  
    ]  
}
```

Get cart templates for current user scheme

Use case: Get cart templates for current user**Basic Flow****Optional Web UI Flow**

Get transfer templates for current user description

Use Case Name

Get transfer templates for current user

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get transfer templates for current user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get transfer templates for current user”.

Endpoint URL: GET /templates/transfer

Parameters: TOKEN

3. System Operator returns transfer templates to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get transfer templates for current user”.

Endpoint URL: GET /templates/transfer

Parameters: TOKEN

3. System Operator returns transfer templates to User (See Result example below).

Post Conditions

List of transfer templates is available.

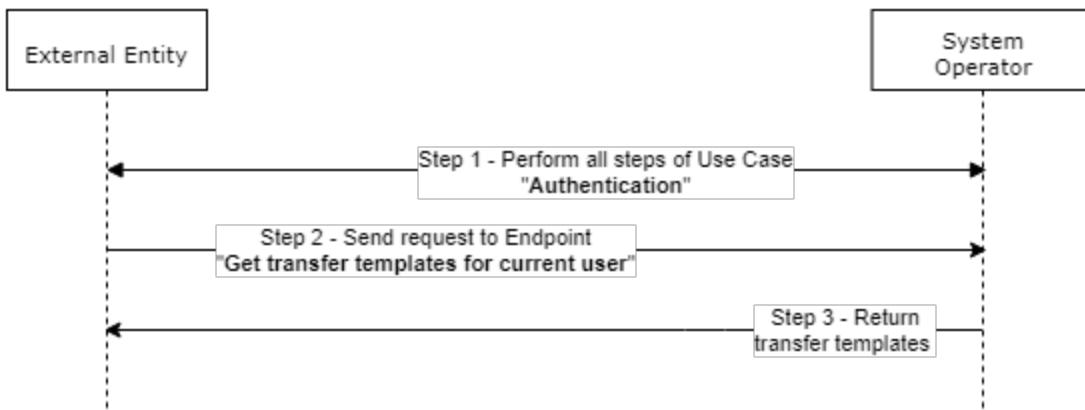
Result example

```
{  
    "transferTemplates": [  
        {  
            "senderCoin": {  
                "serial": "string",  
                "name": "string",  
                "amount": 0,  
                "availableAmount": 0,  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "recipientCoin": {  
                "serial": "string",  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "recipientPhoneNumber": "string",  
            "description": "string",  
            "id": "string",  
            "type": "TRANSFER",  
            "name": "string",  
            "amount": 0  
        }  
    "status": "ok",  
    "message": "string"  
}
```

Get transfer templates for current user scheme

Use case: Get transfer templates for current user

Basic FFlow



Optional Web UI Flow



Get transfer templates for current user with Filter And Pagination description

Use Case Name

Get transfer templates for current user with Filter And Pagination

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get transfer templates for current user” Use Case, find all needed templates, somehow record the details for further filtering, and then send a request to Endpoint “Get transfer templates for current user with Filter And Pagination”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. External Entity sends a request to Endpoint “Get transfer templates for current user with Filter And Pagination”.

Endpoint URL: POST /templates/transfer/view

Parameters:

```
{
  "filter": {
    "templateId": "string",
    "templateName": "string",
    "recipientPhoneNumber": "string"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns transfer templates to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. A user sends a request to Endpoint “Get transfer templates for current user with Filter And Pagination”.

Endpoint URL: POST /templates/transfer/view

Parameters:

```
{  
    "filter": {  
        "templateId": "string",  
        "templateName": "string",  
        "recipientPhoneNumber": "string"  
    },  
    "sort": {  
        "date": "asc"  
    },  
    "pageNumber": 0,  
    "pageSize": 0  
}
```

3. System Operator returns transfer templates to User (See Result example below).

Post Conditions

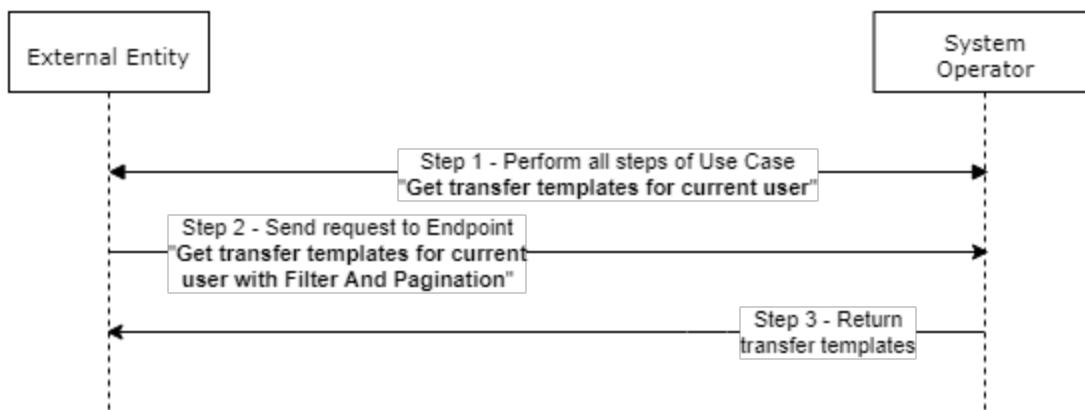
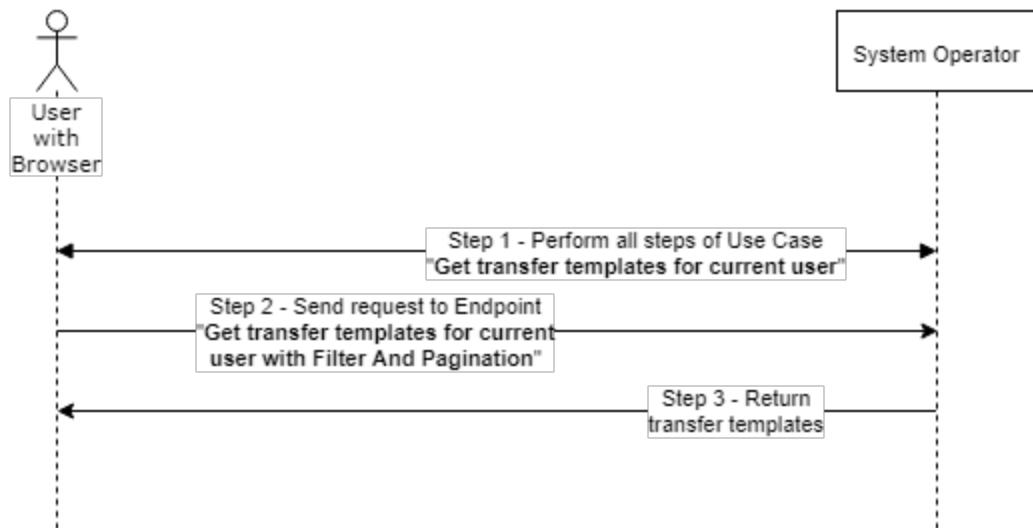
Transfer templates are available.

Result example

```
{  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0,  
    "transfer-templates": [  
        {  
            "senderCoin": {  
                "serial": "string",  
                "name": "string",  
                "amount": 0,  
                "availableAmount": 0,  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "recipientCoin": {  
                "serial": "string",  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "recipientPhoneNumber": "string",  
            "description": "string",  
            "id": "string",  
            "type": "TRANSFER",  
            "name": "string",  
            "amount": 0  
        }  
    ]  
}
```

Get transfer templates for current user with Filter And Pagination scheme

Use case: Get transfer templates for current user with Filter And Pagination

Basic FFlow**Optional Web UI Flow**

Payment by cart template description

Use Case Name

Payment by cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Payment by cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Payment by cart template”.

Endpoint URL: POST /templates/cart/{id}/payment

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.
2. A user sends a request to Endpoint “Payment by cart template”.

Endpoint URL: POST /templates/cart/{id}/payment

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

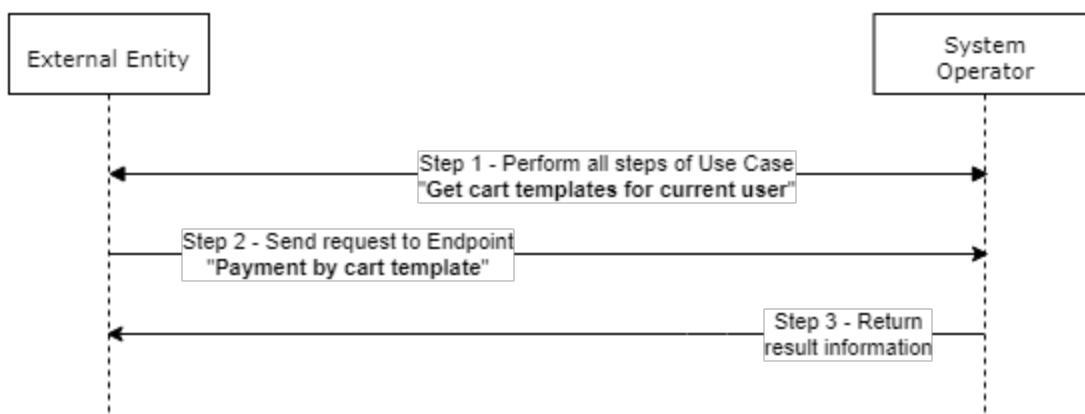
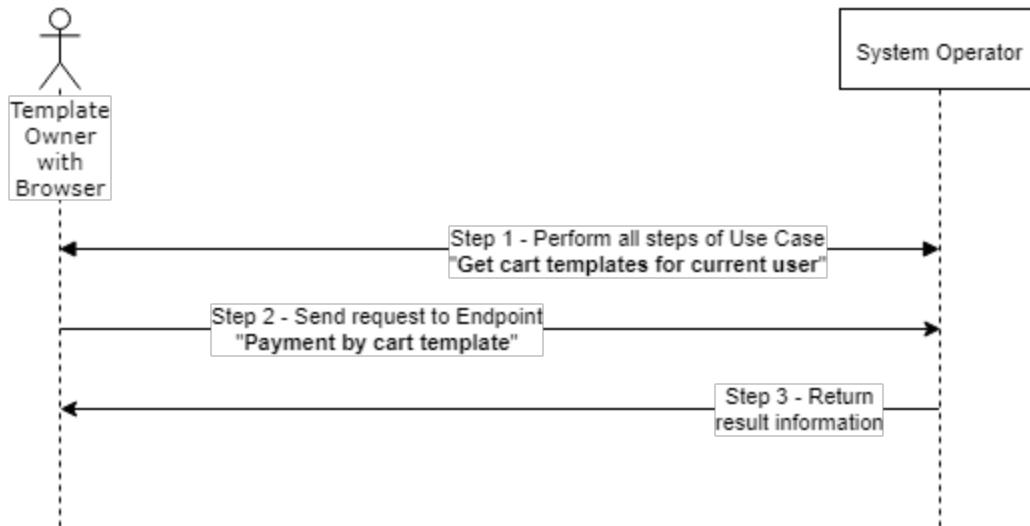
Post Conditions

Payment executed successfully.

Result example

```
{  
    "id": "string",  
    "name": "string",  
    "transactions": [  
        {  
            "id": "string",  
            "orderId": 0,  
            "deviceId": "string",  
            "deviceOrderId": "string",  
            "type": "TOPUP",  
            "status": "INITIATED",  
            "errorCode": "UNKNOWN",  
            "coin": {  
                "serial": "string",  
                "name": "string",  
                "amount": 0,  
                "availableAmount": 0,  
                "issuer": {  
                    "id": "string",  
                    "sn": "string",  
                    "currency": "string"  
                },  
                "active": false,  
                "type": "regular_commission"  
            },  
            "paymentMethod": {  
                "accountId": "string",  
                "account": {  
                    "id": "string",  
                    "provider": {  
                        "name": "string"  
                    }  
                },  
                "way": "string"  
            },  
            "sourceAmount": 0,  
            "amountToSend": 0,  
            "finalAmount": 0,  
            "processId": "string",  
            "payerData": {  
                "  
            }  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Payment by cart template scheme

Use case: Payment by cart template**Basic FLow****Optional Web UI Flow**

Update a cart template description

Use Case Name

Update a cart template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get cart templates for current user” Use Case, and then send a request to Endpoint “Update a cart template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get cart templates for current user”.
2. External Entity sends a request to Endpoint “Update a cart template”.

Endpoint URL: PATCH /templates/cart/{templateId}

Parameters:

```
{
  "name": "string",
  "description": "string",
  "templates": [
    {
      "id": "string",
      "name": "string",
      "amount": 0,
      "description": "string"
    }
  ],
  "templateDataDtos": [
    {
      "id": "string",
      "name": "string",
      "amount": 0,
      "description": "string"
    }
  ],
  "cartTemplateDataDto": {
    "name": "string",
    "description": "string"
  }
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get cart templates for current user”.

2. A user sends a request to Endpoint “Update a cart template”.

Endpoint URL: PATCH /templates/cart/{templateId}

Parameters: same as for Basic Flow above.

3. System Operator returns result information to User (See Result example below).

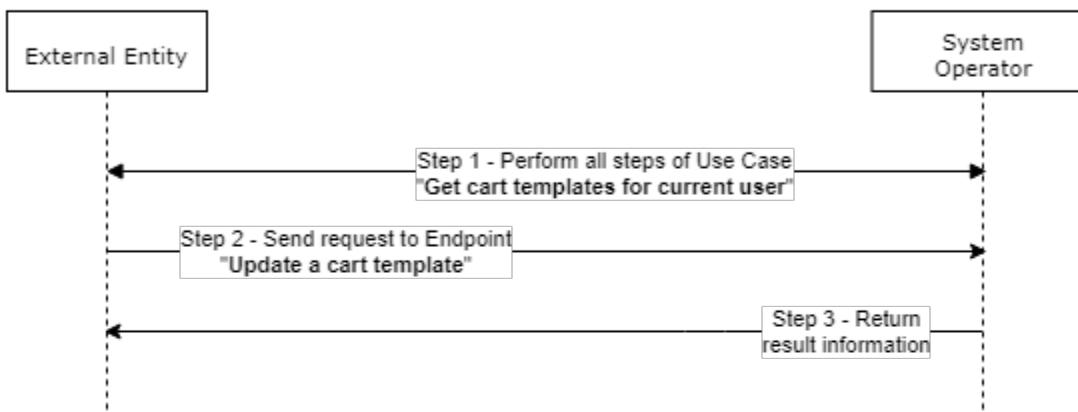
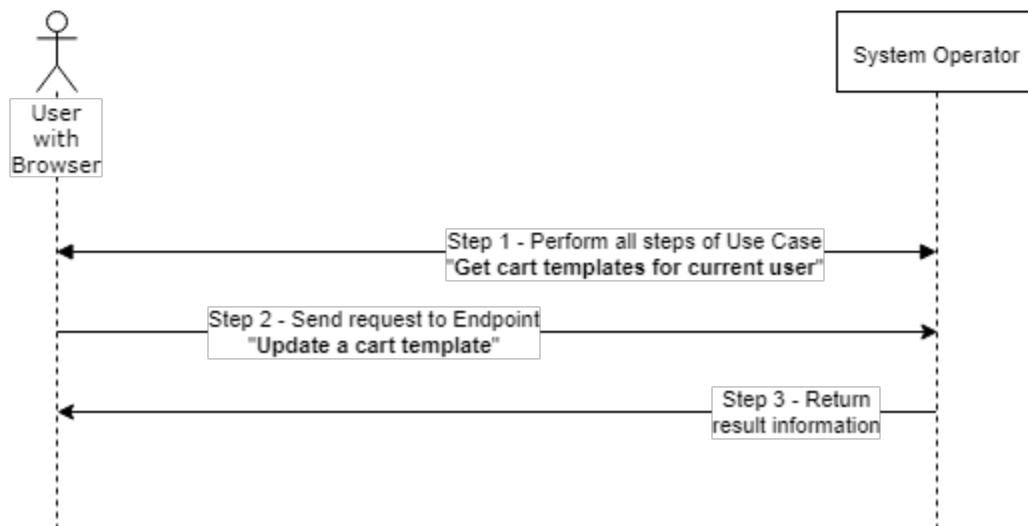
Post Conditions

New template is available.

Result example

```
{
  "id": "string",
  "name": "string",
  "senderCoin": {
    "serial": "string",
    "name": "string",
    "amount": 0,
    "availableAmount": 0,
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    },
    "active": false,
    "type": "regular_commission"
  },
  "recipientCoin": {
    "serial": "string",
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    },
    "active": false,
    "type": "regular_commission"
  },
  "recipientPhoneNumber": "string",
  "amount": 0,
  "description": "string",
  "status": "ok",
  "message": "string"
}
```

Update a cart template scheme

Use case: Update a cart template**Basic FLow****Optional Web UI Flow**

Update a transfer template description

Use Case Name

Update a transfer template

Brief Description

A User or External Entity on behalf of a User with role permission TEMPLATES_OWNER will go through all steps of “Get transfer templates for current user” Use Case, and then send a request to Endpoint “Update a transfer template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TEMPLATES_OWNER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. External Entity sends a request to Endpoint “Update a transfer template”.

Endpoint URL: PATCH /templates/transfer/{id}

Parameters:

```
{
  "name": "string",
  "recipientCoin": "string",
  "senderCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get transfer templates for current user”.
2. A user sends a request to Endpoint “Update a transfer template”.

Endpoint URL: PATCH /templates/transfer/{id}

Parameters:

```
{
  "name": "string",
  "recipientCoin": "string",
  "senderCoin": "string",
  "amount": 0,
  "description": "string"
}
```

3. System Operator returns result information to User (See Result example below).

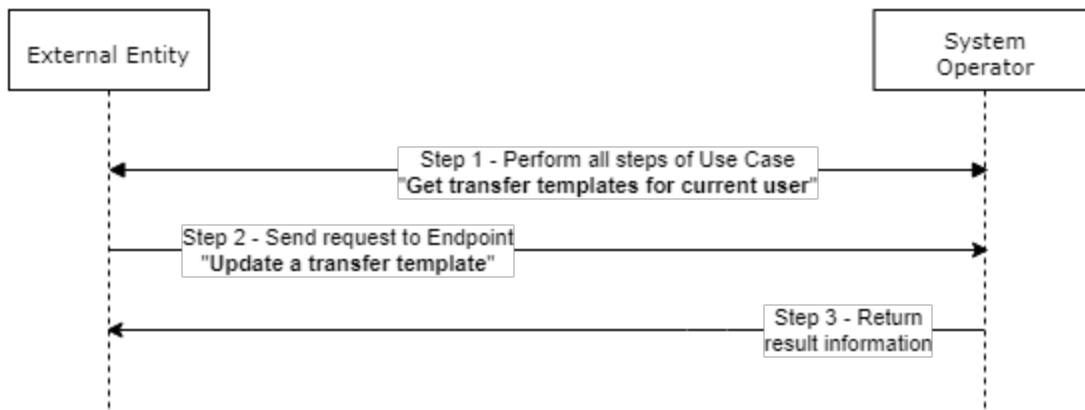
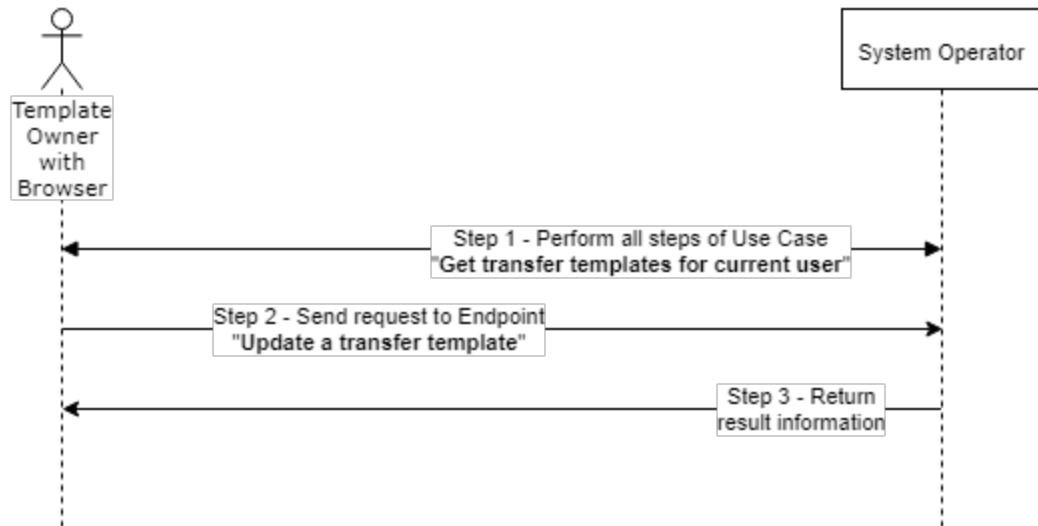
Post Conditions

Updated template is available.

Result example

```
{  
    "id": "string",  
    "name": "string",  
    "senderCoin": {  
        "serial": "string",  
        "name": "string",  
        "amount": 0,  
        "availableAmount": 0,  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "recipientCoin": {  
        "serial": "string",  
        "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
        },  
        "active": false,  
        "type": "regular_commission"  
    },  
    "recipientPhoneNumber": "string",  
    "amount": 0,  
    "description": "string",  
    "status": "ok",  
    "message": "string"  
}
```

Update a transfer template scheme

Use case: Update a transfer template**Basic FLow****Optional Web UI Flow**

Invoice Template

Create an invoice template description

Use Case Name

Top up of authorized capital with cash

Brief Description

A User or External Entity on behalf of a User with role permission "CASH_DESK_INVESTMENT_EXECUTOR" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Top up of authorized capital with cash".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CASH_DESK_INVESTMENT_EXECUTOR", eg. accountant or CFO.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Top up of authorized capital with cash".

Endpoint URL: POST /investments

Parameters: TOKEN - identifies authenticated user

1. System Operator returns cash investment information to User (See Result example below).

Post Conditions

Serial and authorized person are available.

Result example

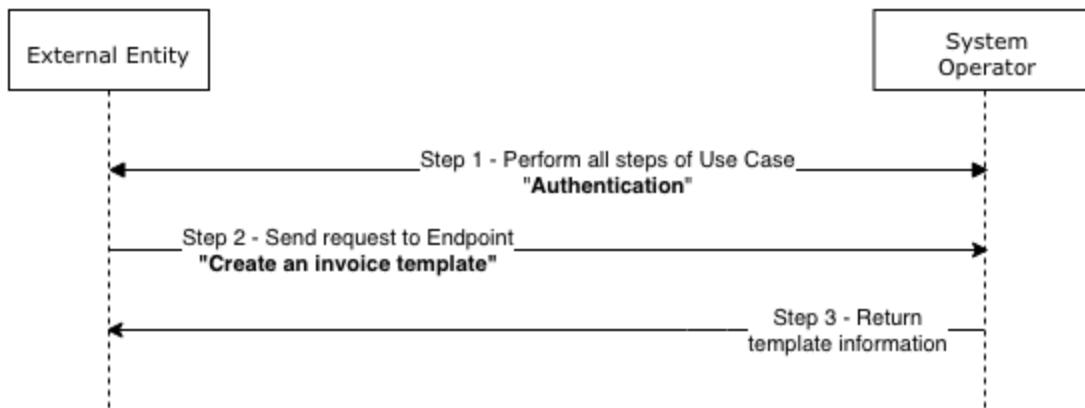
```
{  
  "process": {  
    "id": "string",  
    "createdAt": "2018-09-17T11:07:13.977Z",  
    "updatedAt": "2018-09-17T11:07:13.977Z",  
    "type": "string",  
    "status": "limited",  
    "requestIdentifier": 0,  
    "requestStatus": "limited",  
    "transactions": [  
      {  
        "id": "string",  
        "parentId": "string",  
        "type": "transfer",  
        "from": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        },  
        "to": {  
          "serial": "string",  
          "organizationId": "string",  
          "organizationName": "string",  
          "technical": false,  
          "type": "regular_commission",  
          "issuer": {  
            "id": "string",  
            "sn": "string",  
            "currency": "string"  
          }  
        }  
      }  
    ]  
  }  
}
```

```
 },
 "amount": 0,
 "performedAt": "2018-09-17T11:07:13.977Z",
 "issuer": {
   "id": "string",
   "sn": "string",
   "currency": "string"
 },
 },
 ],
 "children": [
   {
     "errorMessage": "string"
   },
   "status": "ok",
   "message": "string"
 }
```

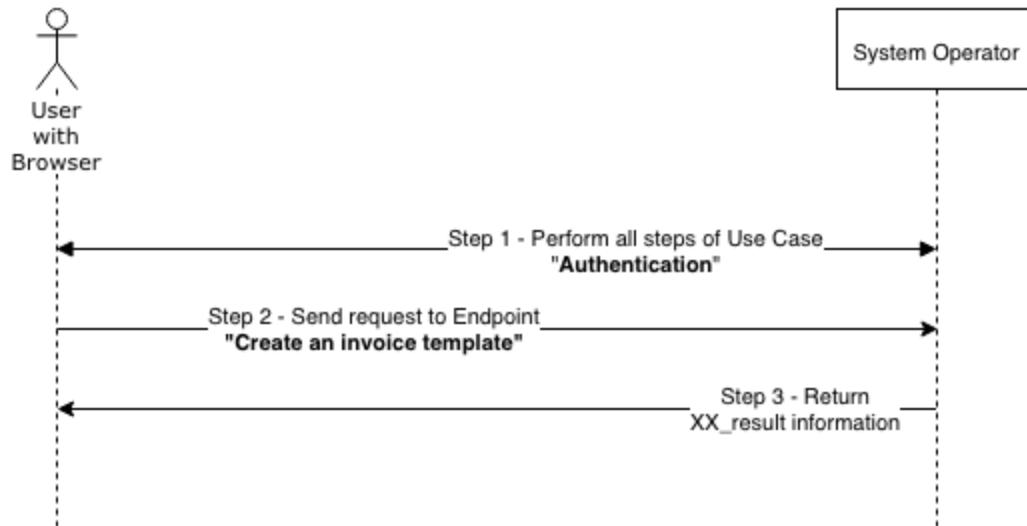
Create an invoice template scheme

Use case: Create an invoice template

Basic FFlow



Optional Web UI Flow



Delete an invoice template description

Use Case Name

Delete an invoice template

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER will go through all steps of “View templates” Use Case, and then send a request to Endpoint “Delete an invoice template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View templates”.
2. External Entity sends a request to Endpoint “Delete an invoice template”.

Endpoint URL: DELETE /invoice-templates/{id}

Parameters: Token

1. System Operator returns system response to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View templates”.
2. A user sends a request to Endpoint “Delete an invoice template”.

Endpoint URL: DELETE /invoice-templates/{id}

Parameters: Token

1. System Operator returns system response to User (See Result example below).

Post Conditions

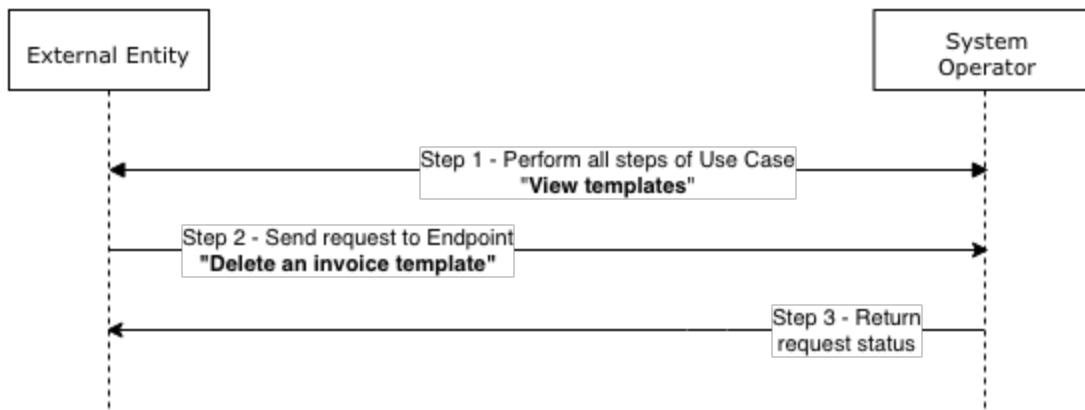
Delete action successful.

Result example

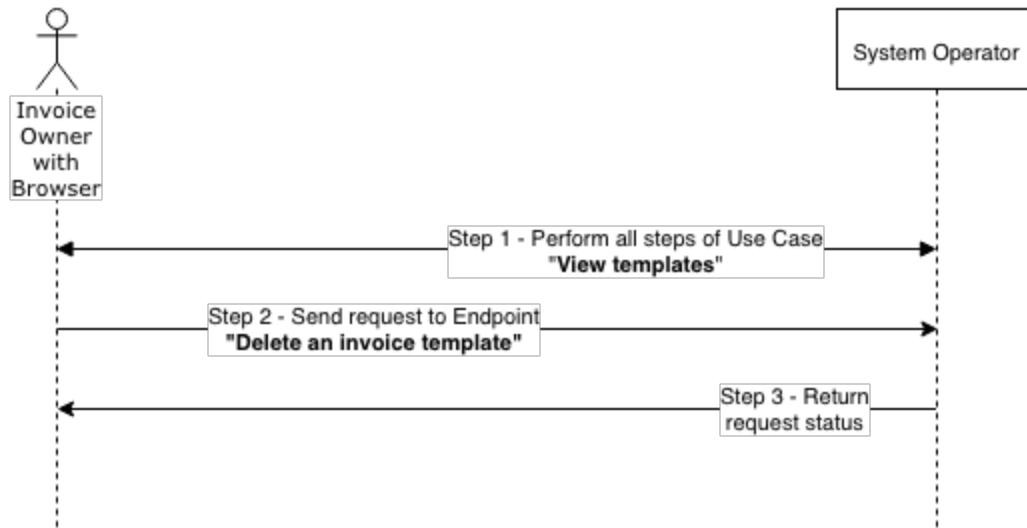
```
{  
  "status": "ok",  
  "message": "string"  
}  
Delete an invoice template scheme
```

Use case: Delete an invoice template

Basic FLow



Optional Web UI Flow



Update invoice template description

Use Case Name

Update invoice template

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER will go through all steps of “View templates” Use Case, and then send a request to Endpoint “Update invoice template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View templates”.
2. External Entity sends a request to Endpoint “Update invoice template”.

Endpoint URL: PATCH /invoice-templates/{id}

Parameters: {

```
"invoiceIdentifier": "string",
"invoiceDraft": {
  "name": "string",
  "amount": 0,
  "payerContact": "string",
  "recipientCoin": "string",
  "data": {
    "productCode": "string",
    "productPrice": 0,
    "description": "string",
    "count": 0,
    "terms": "string"
  },
  "expiresAt": "2018-08-22T09:12:24.542Z"
}
```

```
}
```

1. System Operator returns updated template information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View templates”.
2. A user sends a request to Endpoint “Update invoice template”.

Endpoint URL: PATCH /invoice-templates/{id}

Parameters: {

 "invoiceIdentifier": "string",

 "invoiceDraft": {

 "name": "string",

 "amount": 0,

 "payerContact": "string",

 "recipientCoin": "string",

 "data": {

 "productCode": "string",

 "productPrice": 0,

 "description": "string",

 "count": 0,

 "terms": "string"

 },

 "expiresAt": "2018-08-22T09:12:24.542Z"

 }

}

1. System Operator returns updated template information to User (See Result example below).

Post Conditions

Template is updated.

Result example

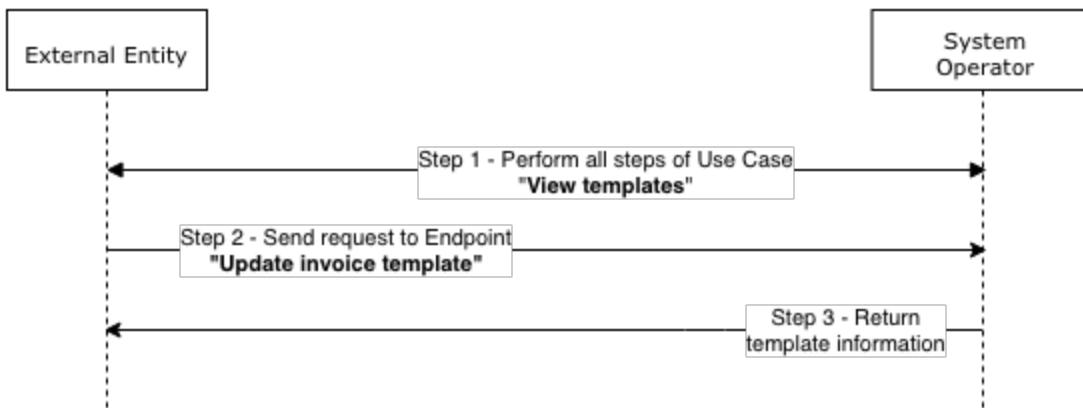
```
{
    "template": {
        "id": "string",
        "templateName": "string",
        "name": "string",
        "amount": 0,
```

```
"payerContact": "string",
"recipientCoin": "string",
"data": {
  "productCode": "string",
  "productPrice": 0,
  "description": "string",
  "count": 0,
  "terms": "string"
},
"createdAt": "2018-08-22T09:12:25.624Z",
"expiresAt": "2018-08-22T09:12:25.624Z"
},
"status": "ok",
"message": "string"
}
```

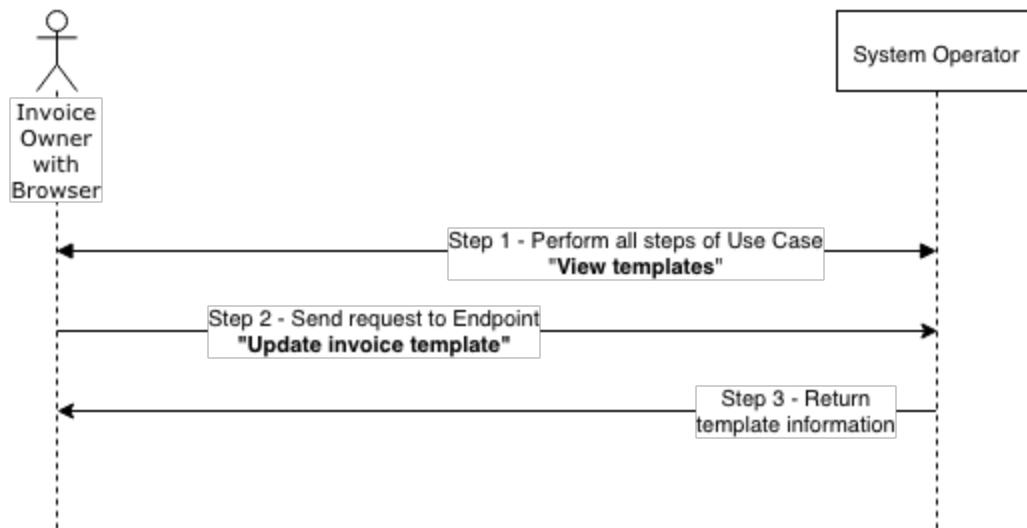
Update invoice template scheme

Use case: Update invoice template

Basic FFlow



Optional Web UI Flow



View templates description

Use Case Name

Update invoice template

Brief Description

A User or External Entity on behalf of a User with role permission INVOICE_OWNER will go through all steps of “View templates” Use Case, and then send a request to Endpoint “Update invoice template”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: INVOICE_OWNER, e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View templates”.
2. External Entity sends a request to Endpoint “Update invoice template”.

Endpoint URL: PATCH /invoice-templates/{id}

Parameters: {

```
"invoiceIdentifier": "string",
"invoiceDraft": {
  "name": "string",
  "amount": 0,
  "payerContact": "string",
  "recipientCoin": "string",
  "data": {
    "productCode": "string",
    "productPrice": 0,
    "description": "string",
    "count": 0,
    "terms": "string"
  },
  "expiresAt": "2018-08-22T09:12:24.542Z"
}
}
```

1. System Operator returns updated template information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View templates”.
2. A user sends a request to Endpoint “Update invoice template”.

Endpoint URL: PATCH /invoice-templates/{id}

Parameters: {

“invoiceIdentifier”: “string”,

“invoiceDraft”: {

“name”: “string”,

“amount”: 0,

“payerContact”: “string”,

“recipientCoin”: “string”,

“data”: {

“productCode”: “string”,

“productPrice”: 0,

“description”: “string”,

“count”: 0,

“terms”: “string”

},

“expiresAt”: “2018-08-22T09:12:24.542Z”

}

}

1. System Operator returns updated template information to User (See Result example below).

Post Conditions

Template is updated.

Result example

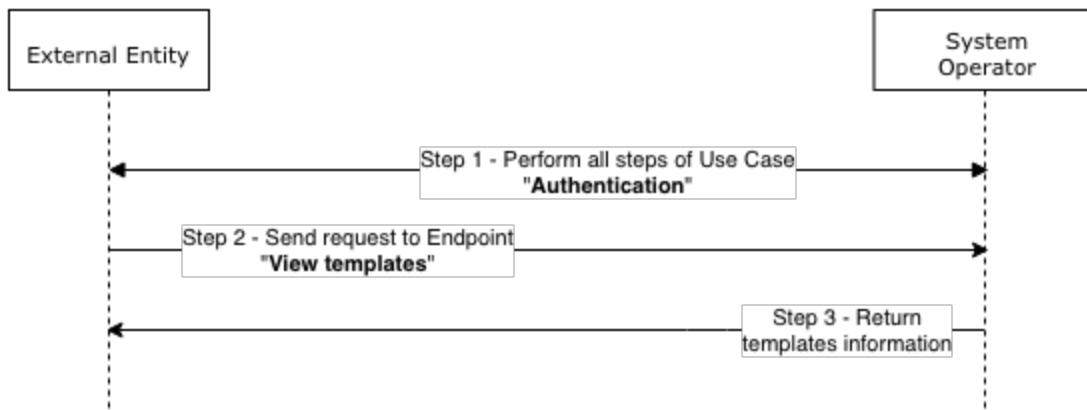
```
{
  “template”: {
    “id”: “string”,
    “templateName”: “string”,
    “name”: “string”,
    “amount”: 0,
    “payerContact”: “string”,
```

```
"recipientCoin": "string",
"data": {
  "productCode": "string",
  "productPrice": 0,
  "description": "string",
  "count": 0,
  "terms": "string"
},
"createdAt": "2018-08-22T09:12:25.624Z",
"expiresAt": "2018-08-22T09:12:25.624Z"
},
"status": "ok",
"message": "string"
}
```

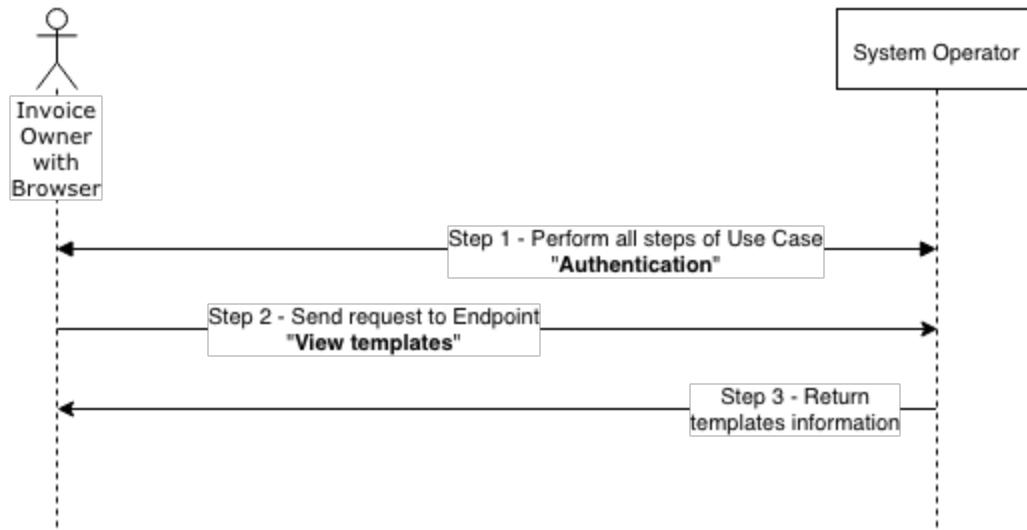
[View templates scheme](#)

Use case: View templates

Basic FFlow



Optional Web UI Flow



Merchant product categories

Add product into category description

Use Case Name

Add product into category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product categories belonging to this merchant" Use Case (to obtain category ID) and "View product list" Use Case (to obtain product ID), and then send a request to Endpoint "Add product into category".

Note:

Alternatively, you can create a new product using the "Create one product" endpoint.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. Perform all steps of Use Case "View product list".
3. External Entity sends a request to Endpoint "Add product into category".

Endpoint URL: PUT /merchant-product-categories/{categoryId}/merchant-products/{productId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. Perform all steps of Use Case "View product list".
3. A user sends a request to Endpoint "Add product into category".

Endpoint URL: PUT /merchant-product-categories/{categoryId}/merchant-products/{productId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product and category are available.

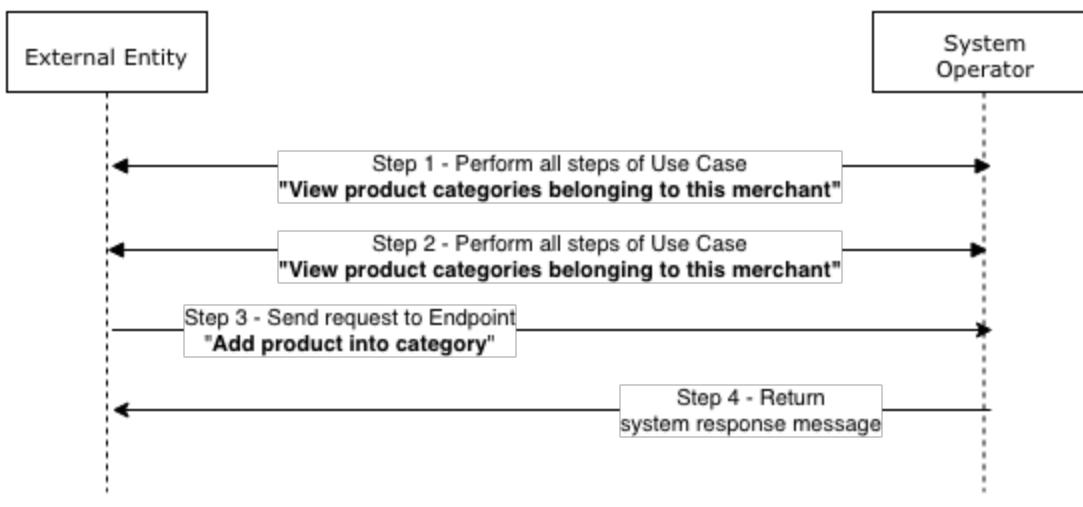
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

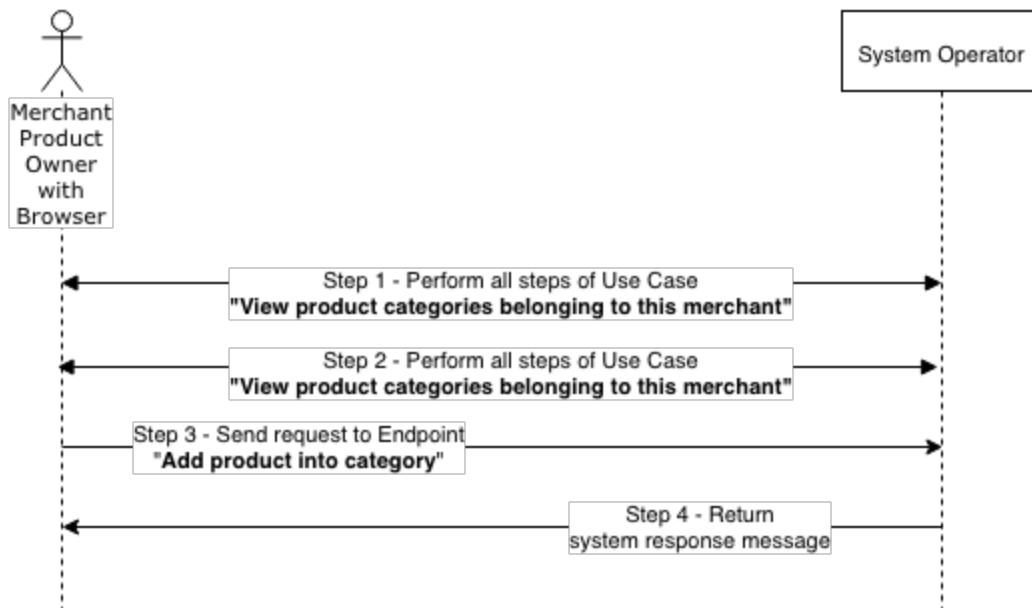
Add product into category scheme

Use case: Add product into category

Basic FFlow



Optional Web UI Flow



Create a product category description

Use Case Name

Create a product category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Create a product category".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create a product category".

Endpoint URL: POST /merchant-product-categories

Parameters: {

```
"externalCode": "string",
"names": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
]
```

```
}
```

1. System Operator returns new category information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a product category”.

Endpoint URL: POST /merchant-product-categories

Parameters: {

```
"externalCode": "string",
"names": [
{
  "locale": "en",
  "value": "translated_string_value"
},
],
"descriptions": [
{
  "locale": "en",
  "value": "translated_string_value"
}
]
```

1. System Operator returns new category information to User (See Result example below).

Post Conditions

Category input parameters are available.

Result example

```
{
  "category": {
    "id": 0,
    "externalCode": "string",
    "name": "string",
    "description": "string"
  },
  "status": "ok",
}
```

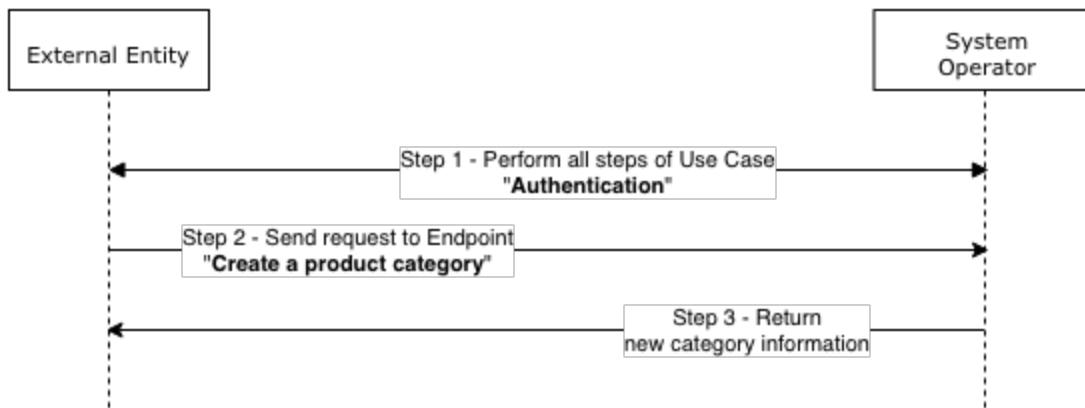
```
"message": "string"
```

```
}
```

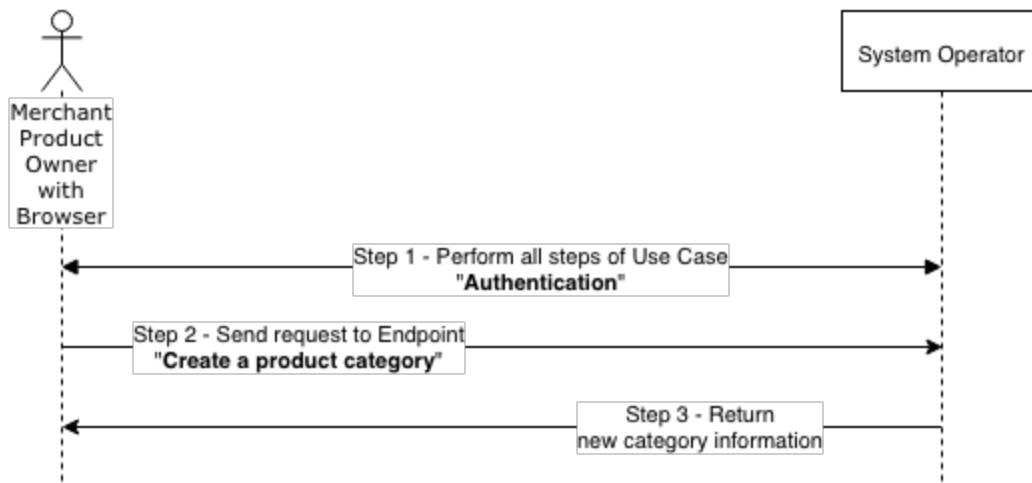
Create a product category scheme

Use case: Create a product category

Basic FFlow



Optional Web UI Flow



Delete a product category description

Use Case Name

Delete a product category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product categories belonging to this merchant" Use Case (to obtain category ID), and then send a request to Endpoint "Delete a product category".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. External Entity sends a request to Endpoint "Delete a product category".

Endpoint URL: DELETE /merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. A user sends a request to Endpoint "Delete a product category".

Endpoint URL: DELETE /merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Category is available.

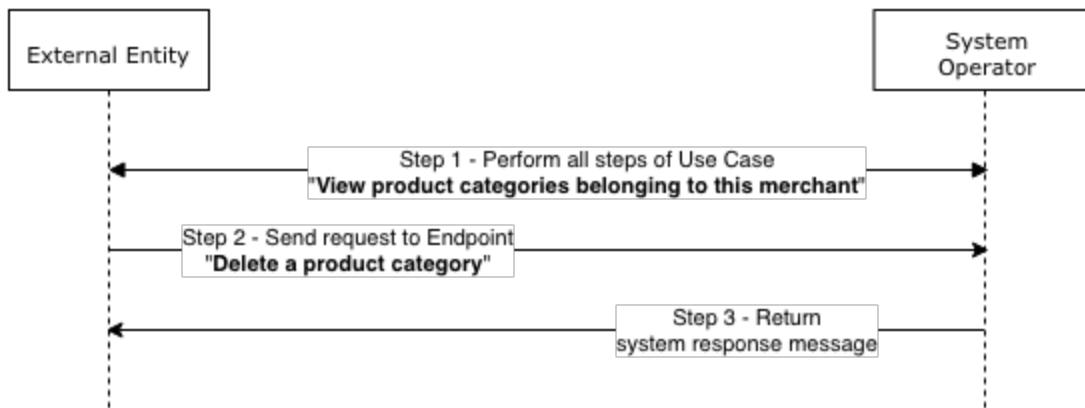
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

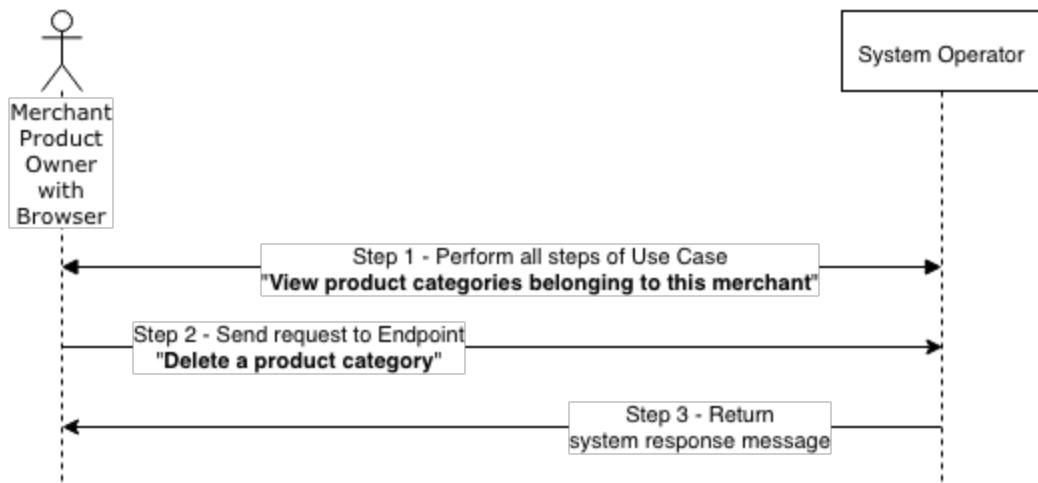
Delete a product category scheme

Use case: Delete a product category

Basic FFlow



Optional Web UI Flow



Remove product from category description

Use Case Name

Remove product from category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View products belonging to category" Use Case (to obtain product and category IDs), and then send a request to Endpoint "Remove product from category".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View products belonging to category".
2. External Entity sends a request to Endpoint "Remove product from category".

Endpoint URL: DELETE /merchant-product-categories/{categoryId}/merchant-products/{productId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View products belonging to category".
2. A user sends a request to Endpoint "Remove product from category".

Endpoint URL: DELETE /merchant-product-categories/{categoryId}/merchant-products/{productId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product and category are available.

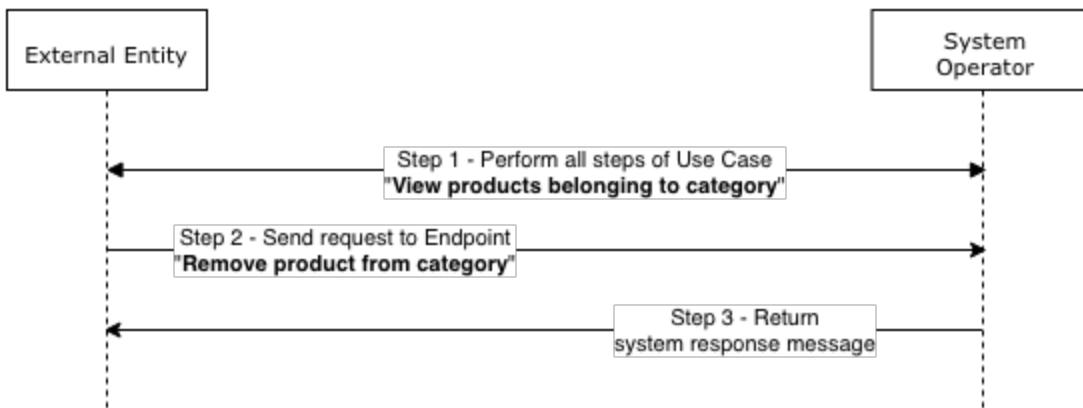
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

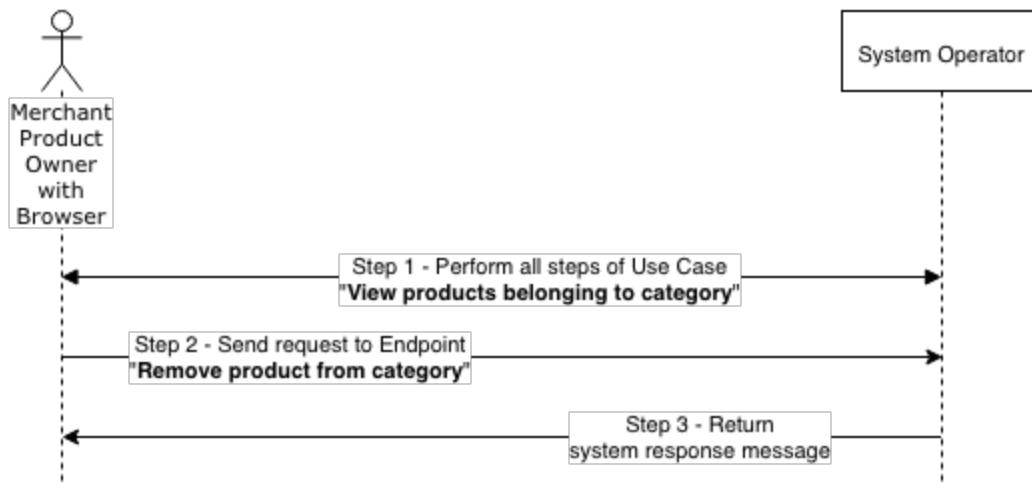
Remove product from category scheme

Use case: Remove product from category

Basic FFlow



Optional Web UI Flow



Update a product category scheme

Use Case Name

Update a product category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product categories belonging to this merchant" Use Case (to obtain category ID), and then send a request to Endpoint "Update a product category".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. External Entity sends a request to Endpoint "Update a product category".

Endpoint URL: PATCH /merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns updated category information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. A user sends a request to Endpoint "Update a product category".

Endpoint URL: PATCH /merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns updated category information to User (See Result example below).

Post Conditions

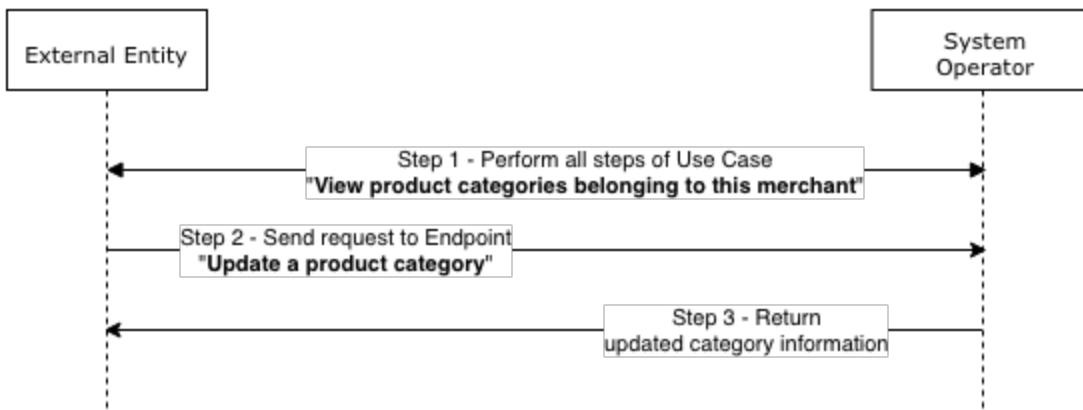
Category is available.

Result example

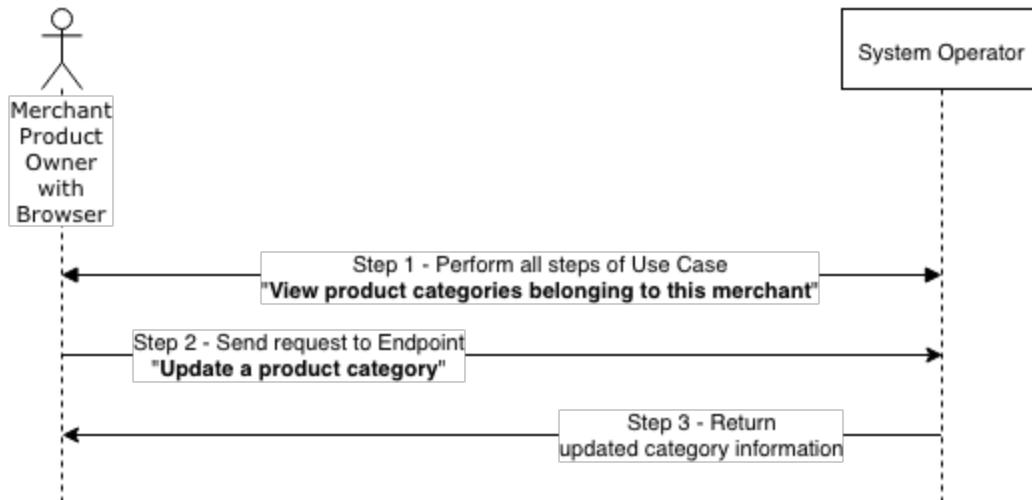
```
{  
  "category": {  
    "id": 0,  
    "externalCode": "string",  
    "name": "string",  
    "description": "string"  
  },  
  "status": "ok",  
  "message": "string"  
}  
Update a product category description
```

Use case: Update a product category

Basic FFlow



Optional Web UI Flow



View product categories belonging to this merchant description

Use Case Name

View product categories belonging to this merchant

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "View product categories belonging to this merchant".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View product categories belonging to this merchant".

Endpoint URL: GET /merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of merchant's product categories to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "View product categories belonging to this merchant".

Endpoint URL: GET /merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of merchant's product categories to User (See Result example below).

Post Conditions

Product category is available.

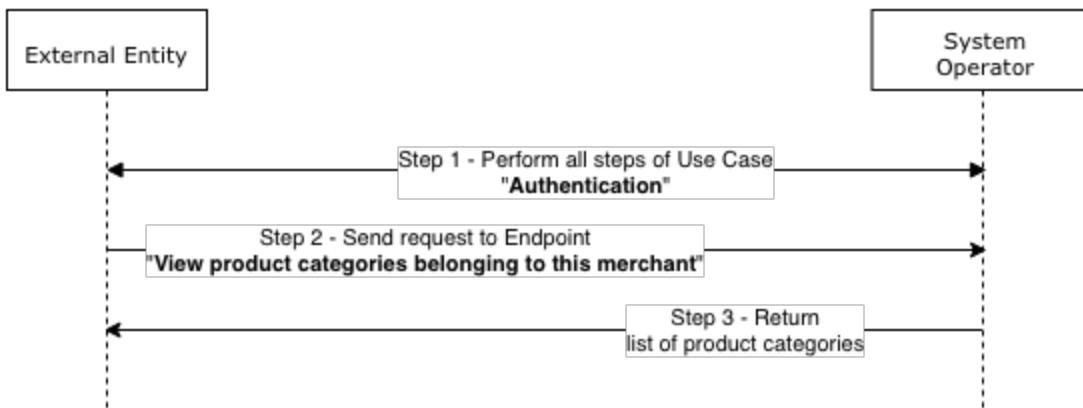
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "name": "string",  
      "description": "string"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

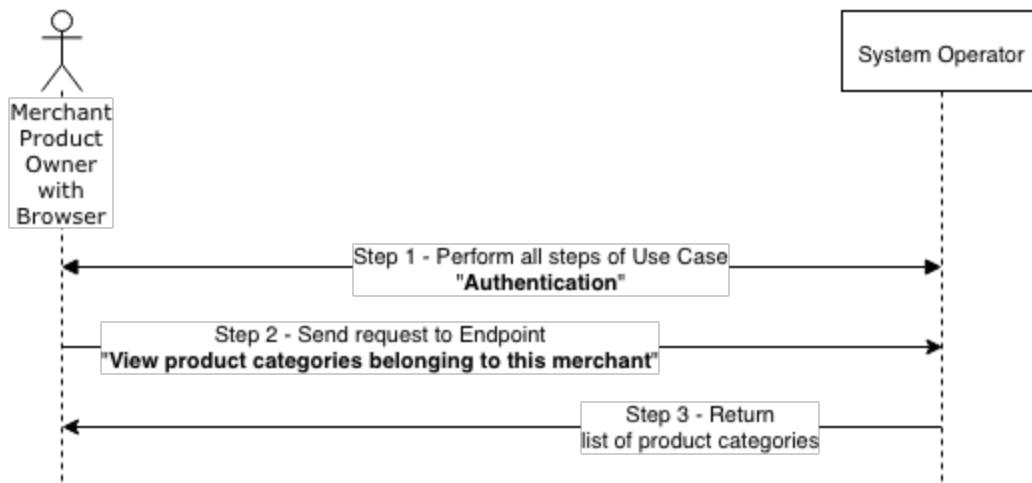
View product categories belonging to this merchant scheme

Use case: View product categories belonging to this merchant

Basic FFlow



Optional Web UI Flow



View products belonging to category description

Use Case Name

View products belonging to category

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product categories belonging to this merchant" Use Case (to obtain category ID), and then send a request to Endpoint "View products belonging to category".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER", e.g. e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. External Entity sends a request to Endpoint "View products belonging to category".

Endpoint URL: GET /merchant-product-categories/{categoryId}/merchant-products

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of products belonging to a category to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product categories belonging to this merchant".
2. A user sends a request to Endpoint "View products belonging to category".

Endpoint URL: GET /merchant-product-categories/{categoryId}/merchant-products

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of products belonging to a category to User (See Result example below).

Post Conditions

Category is available.

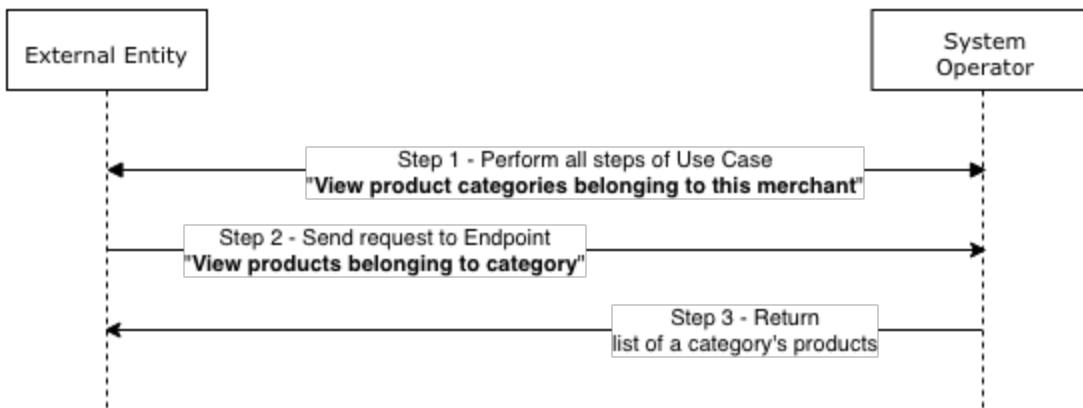
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "name": "string",  
      "description": "string",  
      "measureUnit": {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
      }  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

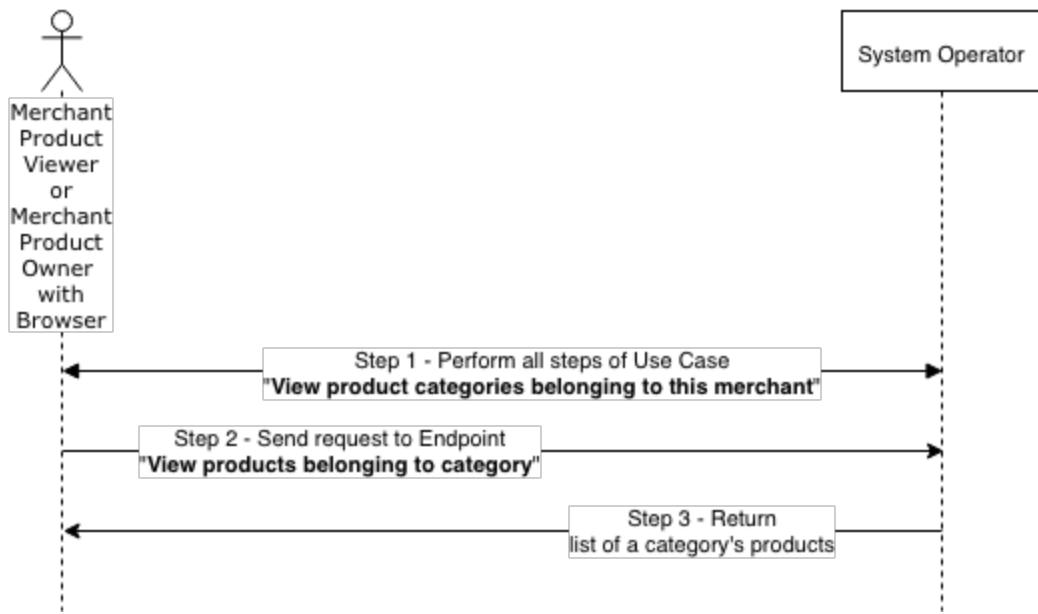
View products belonging to category scheme

Use case: View products belonging to category

Basic FFlow



Optional Web UI Flow



Merchant product

Create one product description

Use Case Name

Create one product

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Create one product".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create one product".

Endpoint URL: POST /merchant-products

Parameters: {

```
"externalCode": "string",
"measureUnit": {
    "identifier": 0,
    "externalCode": "string"
},
"names": [
    {
        "locale": "en",
        "value": "translated_string_value"
    }
],
"descriptions": [
    {
        "locale": "en",
        "value": "translated_string_value"
    }
]
```

```

    "value": "translated_string_value"
}
]
}

```

1. System Operator returns new product information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create one product”.

Endpoint URL: POST /merchant-products

```

Parameters: {
  "externalCode": "string",
  "measureUnit": {
    "identifier": 0,
    "externalCode": "string"
  },
  "names": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ],
  "descriptions": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ]
}

```

1. System Operator returns new product information to User (See Result example below).

Post Conditions

Product information is available.

Result example

```
{

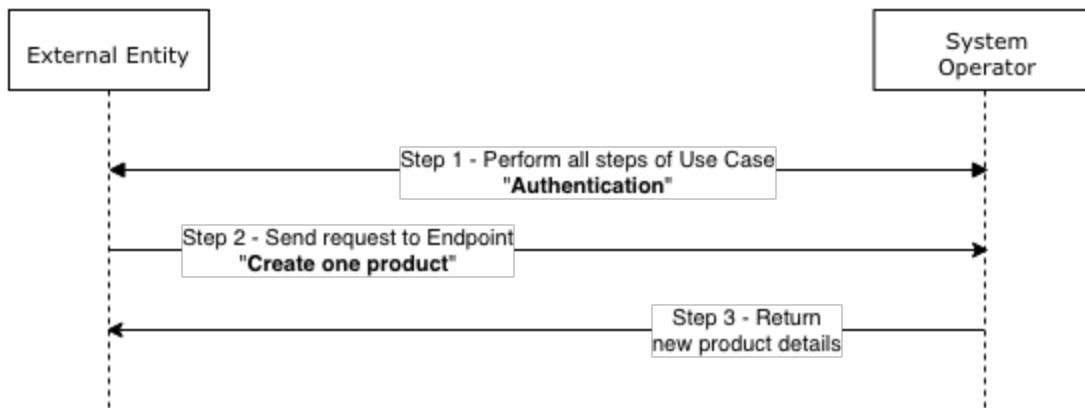
```

```
"product": {  
    "id": 0,  
    "externalCode": "string",  
    "name": "string",  
    "description": "string",  
    "measureUnit": {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
    }  
},  
"status": "ok",  
"message": "string"  
}
```

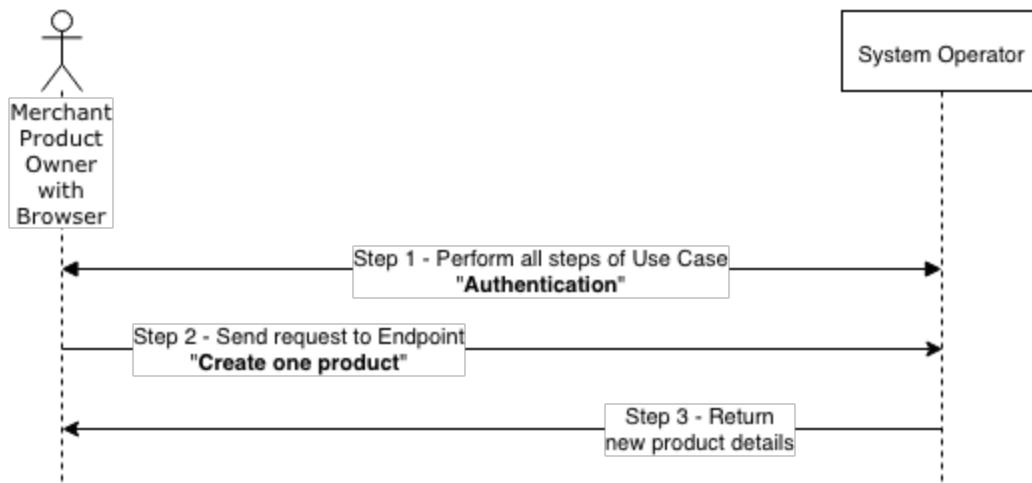
Create one product scheme

Use case: Create one product

Basic FFlow



Optional Web UI Flow



Create products from batch description

Use Case Name

Create products from batch

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Create products from batch".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create products from batch".

Endpoint URL: POST /merchant-products/batch-create

Parameters: {

```
"products": [
  {
    "externalCode": "string",
    "measureUnit": {
      "identifier": 0,
      "externalCode": "string"
    },
    "names": [
      {
        "locale": "en",
        "value": "translated_string_value"
      }
    ],
    "descriptions": [
      {
        "locale": "en",
        "value": "translated_string_value"
      }
    ]
}
```

```

    }
]
}
]
}
}
```

1. System Operator returns list of new products to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create products from batch”.

Endpoint URL: POST /merchant-products/batch-create

Parameters: {

```

"products": [
{
  "externalCode": "string",
  "measureUnit": {
    "identifier": 0,
    "externalCode": "string"
  },
  "names": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ],
  "descriptions": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ]
}
```

1. System Operator returns list of new products to User (See Result example below).

Post Conditions

Product input parameters are available.

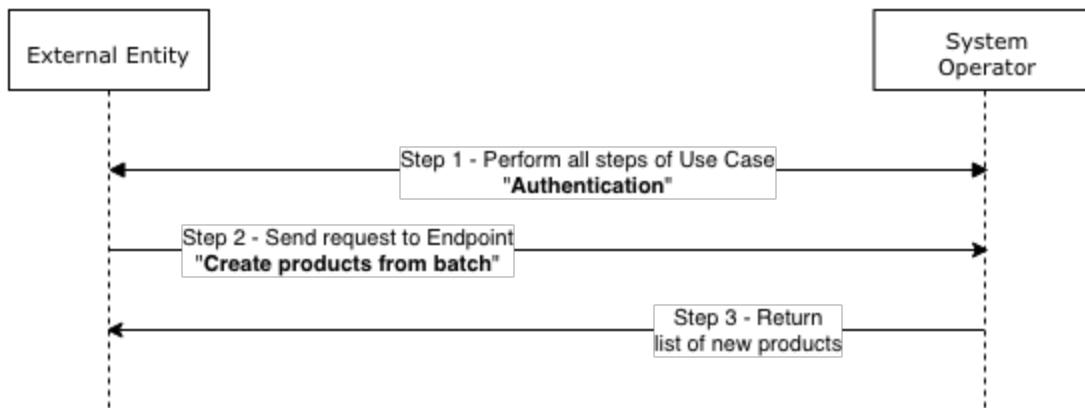
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "name": "string",  
      "description": "string",  
      "measureUnit": {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
      }  
    }  
  ],  
  "status": "ok",  
  "message": "string",  
  "pageNumber": 0,  
  "pageSize": 0,  
  "totalRecords": 0,  
  "totalPages": 0  
}
```

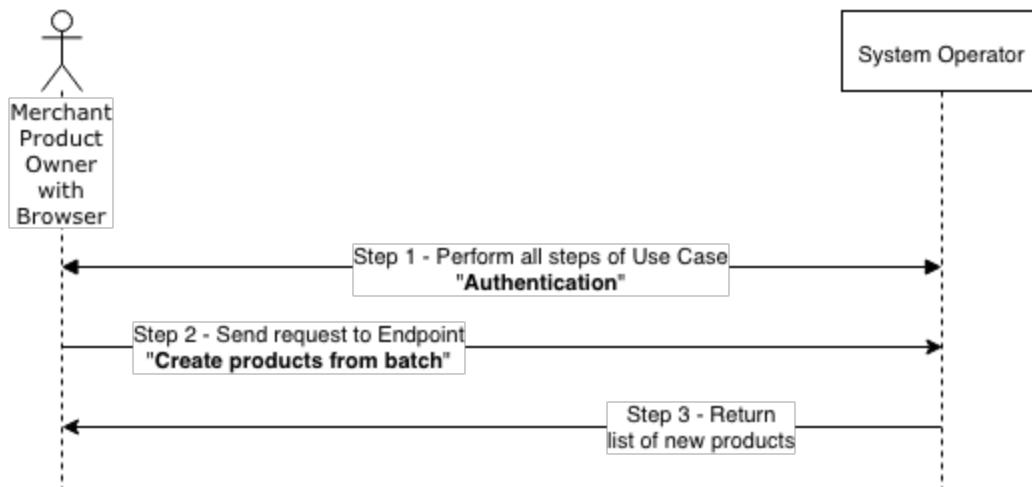
Create products from batch scheme

Use case: Create products from batch

Basic FFlow



Optional Web UI Flow



Delete by external code description

Use Case Name

Delete by external code

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product by it's ID" Use Case (to obtain external code parameter), and then send a request to Endpoint "Delete by external code".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product by it's ID".
2. External Entity sends a request to Endpoint "Delete by external code".

Endpoint URL: POST /merchant-products/delete-by-external-code

Parameters: {

```
"externalCode": "string"
}
```

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product by it's ID".
2. A user sends a request to Endpoint "Delete by external code".

Endpoint URL: POST /merchant-products/delete-by-external-code

Parameters: {

```
"externalCode": "string"
}
```

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product is available.

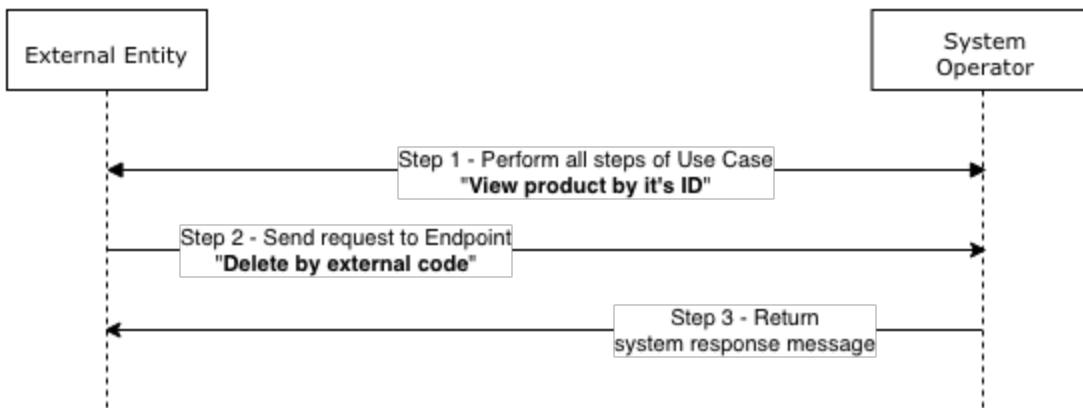
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

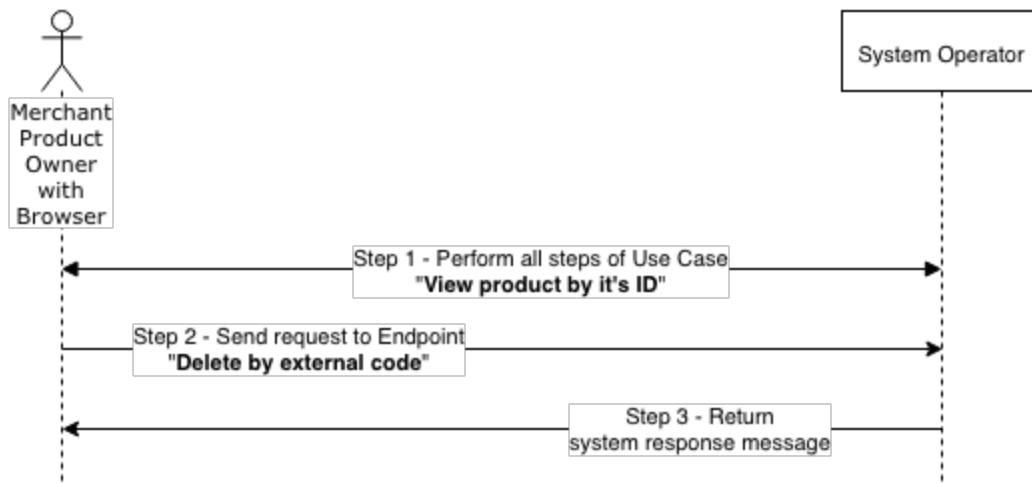
Delete by external code scheme

Use case: Delete by external code

Basic FFlow



Optional Web UI Flow



Delete by ID description

Use Case Name

Delete by ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product by it's ID" Use Case (to obtain product identifier), and then send a request to Endpoint "Delete by ID".

Note:

Note:

"View product by it's ID" Use Case calls "View product list" which can alternatively be used to obtain a list of product identifiers.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product by it's ID".
2. External Entity sends a request to Endpoint "Delete by ID".

Endpoint URL: DELETE /merchant-products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product by it's ID".
2. A user sends a request to Endpoint "Delete by ID".

Endpoint URL: DELETE /merchant-products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product is available.

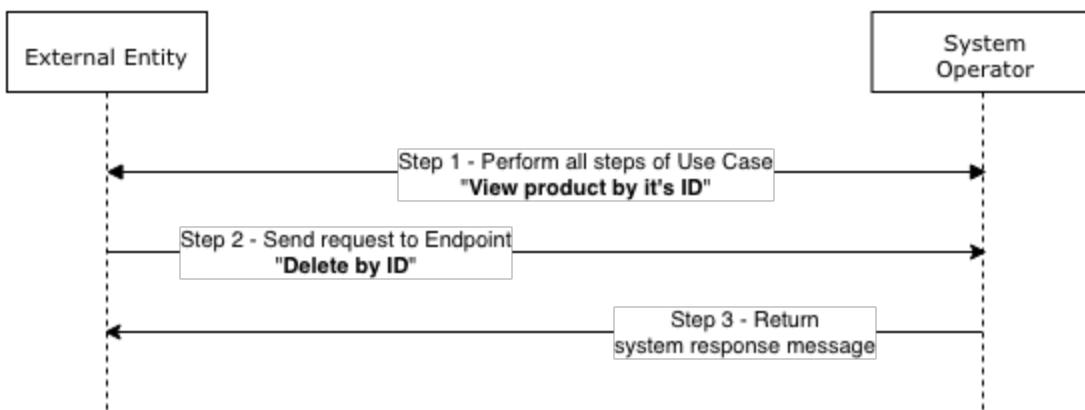
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

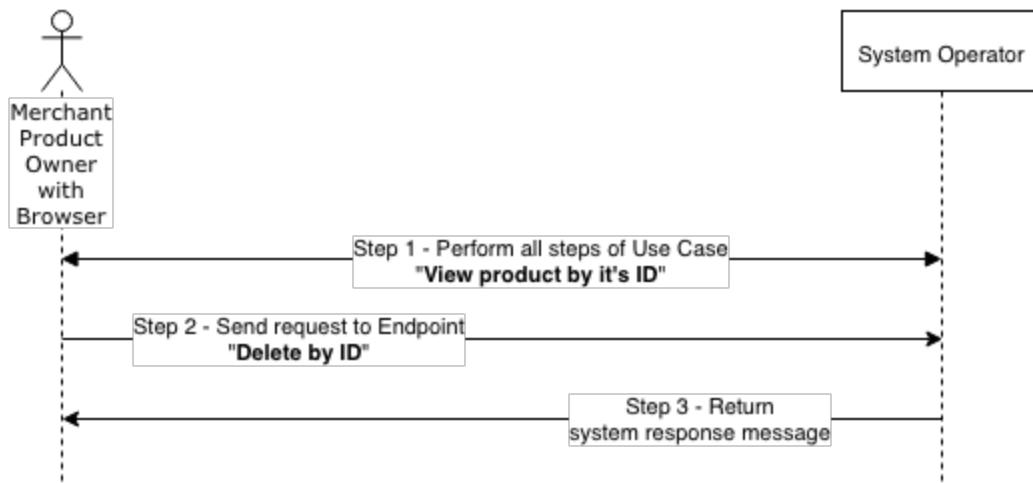
Delete by ID scheme

Use case: Delete by ID

Basic FFlow



Optional Web UI Flow



Delete list of products description

Use Case Name

Delete list of products

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain list of product identifiers), and then send a request to Endpoint "Delete list of products".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "Delete list of products".

Endpoint URL: POST /merchant-products/batch-delete

```
Parameters: {
  "productIds": [
    "string"
  ]
}
```

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product list".
2. A user sends a request to Endpoint "Delete list of products".

Endpoint URL: POST /merchant-products/batch-delete

```
Parameters: {
  "productIds": [
    "string"
  ]
}
```

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product is available.

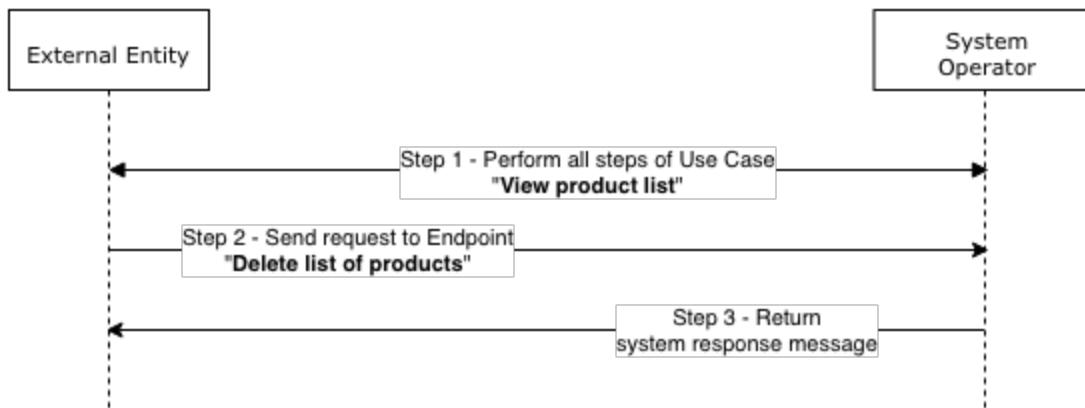
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

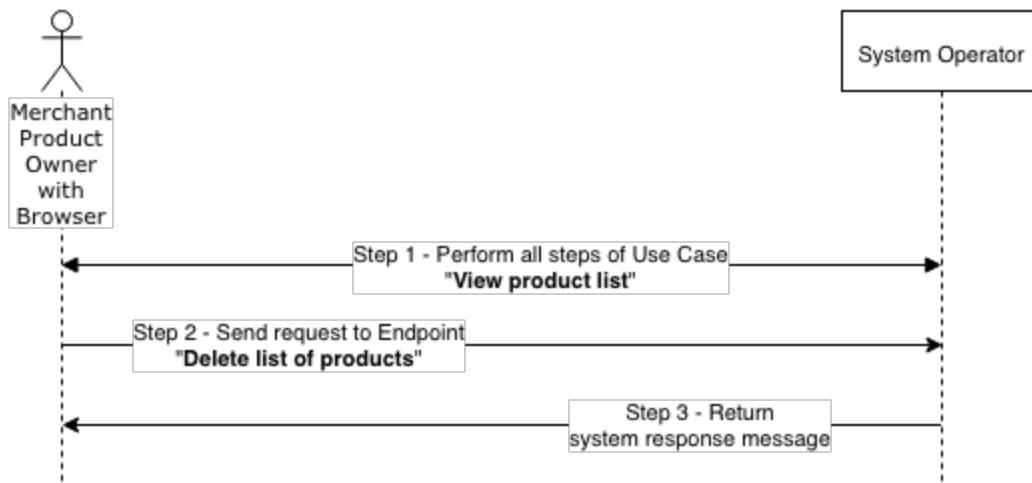
Delete list of products scheme

Use case: Delete list of products

Basic FFlow



Optional Web UI Flow



Update product by it's external code description

Use Case Name

Update product by it's external code

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product by it's ID" Use Case (to obtain external code), and then send a request to Endpoint "Update product by it's external code".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product by it's ID".
2. External Entity sends a request to Endpoint "Update product by it's external code".

Endpoint URL: POST /merchant-products/update-by-external-code

Parameters: {

```
"names": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
```

```
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
```

```
"measureUnit": {
  "identifier": 0,
  "externalCode": "string"
},
```

```

"externalCode": "string"
}

```

1. System Operator returns updated product information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View product by it's ID”.
2. A user sends a request to Endpoint “Update product by it's external code”.

Endpoint URL: POST /merchant-products/update-by-external-code

Parameters: {

```

"names": [
{
  "locale": "en",
  "value": "translated_string_value"
},
],
"descriptions": [
{
  "locale": "en",
  "value": "translated_string_value"
},
],
"measureUnit": {
  "identifier": 0,
  "externalCode": "string"
},
"externalCode": "string"
}

```

1. System Operator returns updated product information to User (See Result example below).

Post Conditions

Product is available.

Result example

```
{
"product": {
  "id": 0,

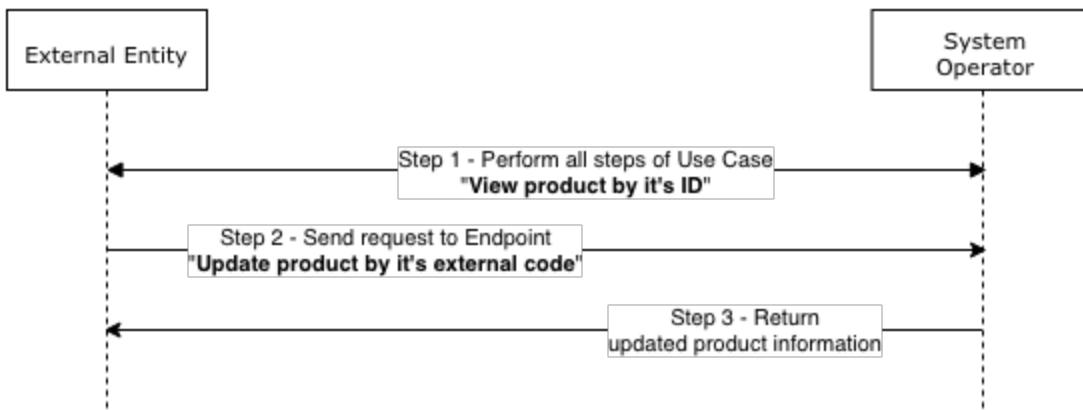
```

```
"externalCode": "string",
"name": "string",
"description": "string",
"measureUnit": {
  "id": 0,
  "externalCode": "string",
  "code": "string",
  "description": "string"
},
},
"status": "ok",
"message": "string"
}
```

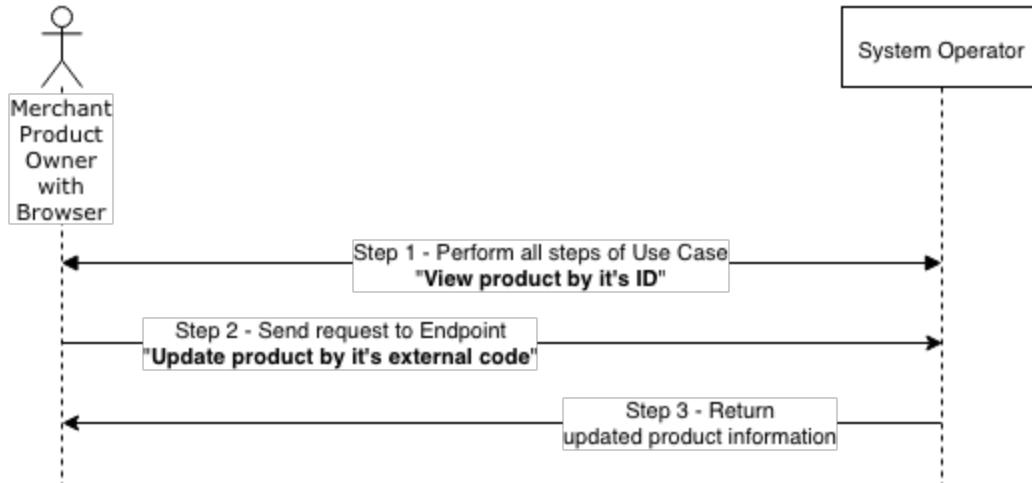
Update product by it's external code scheme

Use case: Update product by it's external code

Basic FFlow



Optional Web UI Flow



Update product by it's identifier description

Use Case Name

Update product by it's identifier

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product by it's ID" Use Case (to obtain product identifier), and then send a request to Endpoint "Update product by it's identifier".

Note:

"View product by it's ID" Use Case calls "View product list" which can alternatively be used to obtain a list of product identifiers.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product by it's ID".
2. External Entity sends a request to Endpoint "Update product by it's identifier".

Endpoint URL: PATCH /merchant-products/{id}

Parameters: {

```
"names": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"measureUnit": {
  "identifier": 0,
  "label": "Label"
}
```

```

    "externalCode": "string"
}
}
}
```

1. System Operator returns updated product information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View product by it's ID”.
2. A user sends a request to Endpoint “Update product by it's identifier”.

Endpoint URL: PATCH /merchant-products/{id}

Parameters: {

```

"names": [
{
  "locale": "en",
  "value": "translated_string_value"
},
],
"descriptions": [
{
  "locale": "en",
  "value": "translated_string_value"
},
],
"measureUnit": {
  "identifier": 0,
  "externalCode": "string"
}
}
```

1. System Operator returns updated product information to User (See Result example below).

Post Conditions

Product is available.

Result example

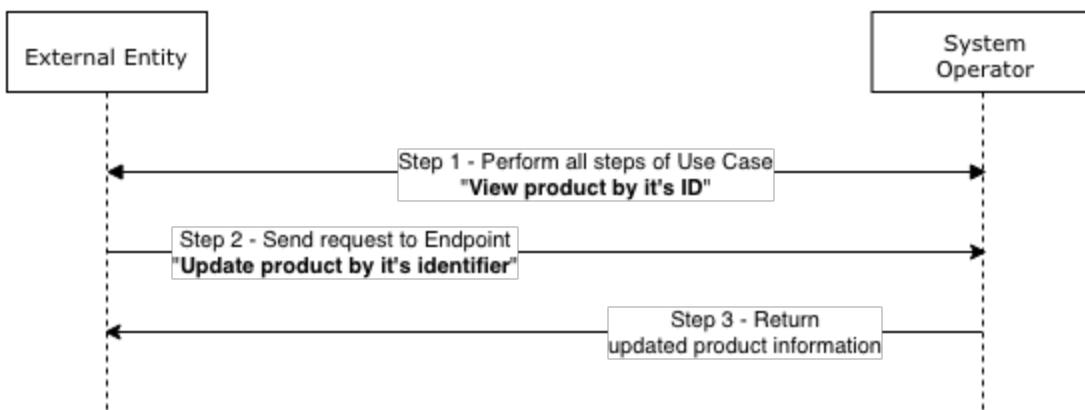
```
{
  "product": {
    "id": 0,
```

```
"externalCode": "string",
"name": "string",
"description": "string",
"measureUnit": {
  "id": 0,
  "externalCode": "string",
  "code": "string",
  "description": "string"
},
},
"status": "ok",
"message": "string"
}
```

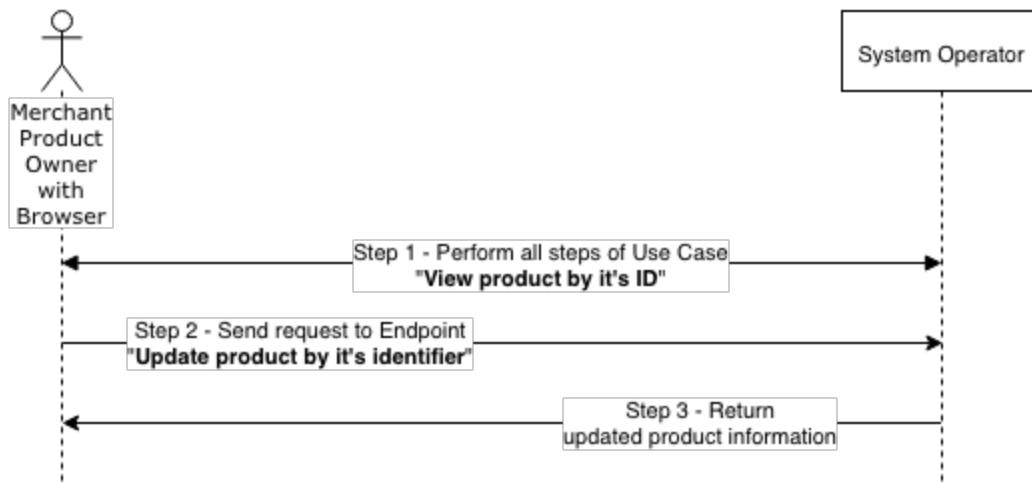
Update product by it's identifier scheme

Use case: Update product by it's identifier

Basic FFlow



Optional Web UI Flow



View categories specified product belongs to description

Use Case Name

View categories specified product belongs to

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain product identifier), and then send a request to Endpoint "View categories specified product belongs to".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER", e.g. e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "View categories specified product belongs to".

Endpoint URL: GET /merchant-products/{productId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of categories to which product belongs to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product list".
2. A user sends a request to Endpoint "View categories specified product belongs to".

Endpoint URL: GET /merchant-products/{productId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of categories to which product belongs to User (See Result example below).

Post Conditions

Product is available.

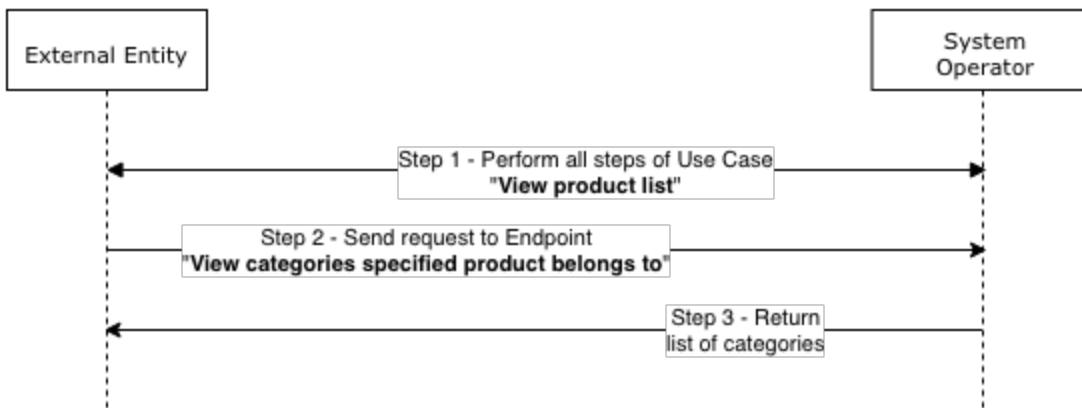
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "name": "string",  
      "description": "string"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

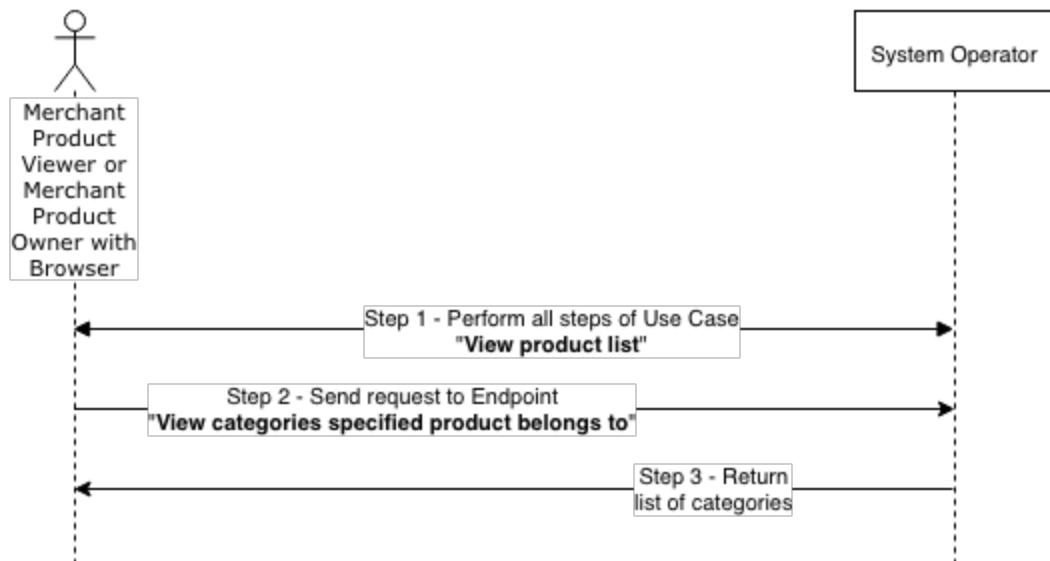
View categories specified product belongs to scheme

Use case: View categories specified product belongs to

Basic FFlow



Optional Web UI Flow



View product by it's ID description

Use Case Name

View product by it's ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain product identifier), and then send a request to Endpoint "View product by it's ID".

Note:

Optional: Accept language - Two-digit language code for localized variables, list of available locale codes can be found at <http://www.oracle.com/technetwork/java/javase/javase8locales-2095355.html>.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "View product by it's ID".

Endpoint URL: GET /merchant-products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns product information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product list".
2. A user sends a request to Endpoint "View product by it's ID".

Endpoint URL: GET /merchant-products/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns product information to User (See Result example below).

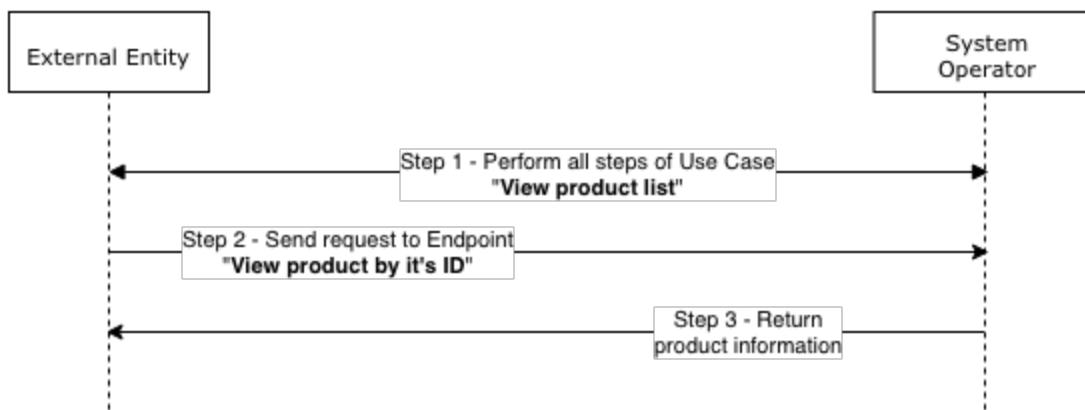
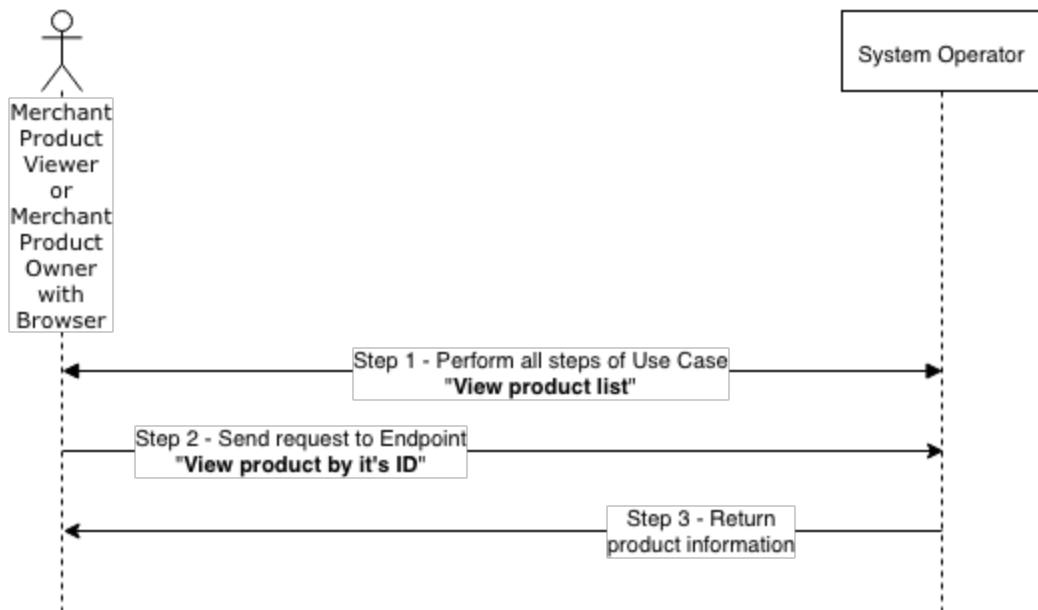
Post Conditions

Product is available.

Result example

```
{  
  "product": {  
    "id": 0,  
    "externalCode": "string",  
    "name": "string",  
    "description": "string",  
    "measureUnit": {  
      "id": 0,  
      "externalCode": "string",  
      "code": "string",  
      "description": "string"  
    }  
  },  
  "status": "ok",  
  "message": "string"  
}
```

View product by it's ID scheme

Use case: View product by it's ID**Basic FFlow****Optional Web UI Flow**

View product list description

Use Case Name

View product list

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View product list".

Notes:

- The filter is optional. When there are no filtering parameter available, the filter could be omitted to get all products in the system.
- Optional: Accept language - Two-digit language code for localized variables, list of available locale codes can be found at <http://www.oracle.com/technetwork/java/javase/java8locales-2095355.html>

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER", e.g. e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View product list".

Endpoint URL: POST /merchant-products/view

Parameters: {

"pageNumber": 0,

"pageSize": 0,

"filter": {

"ids": [

0

],

"externalCodes": [

"string"

],

```

    "name": "string"
},
"sort": {
    "externalCode": "asc",
    "date": "asc"
}
}

```

1. System Operator returns list of products to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View product list”.

Endpoint URL: POST /merchant-products/view

Parameters: {

```

    "pageNumber": 0,
    "pageSize": 0,
    "filter": {
        "ids": [
            0
        ],
        "externalCodes": [
            "string"
        ],
        "name": "string"
    },
    "sort": {
        "externalCode": "asc",
        "date": "asc"
    }
}

```

1. System Operator returns list of products to User (See Result example below).

Post Conditions

One or more products are available.

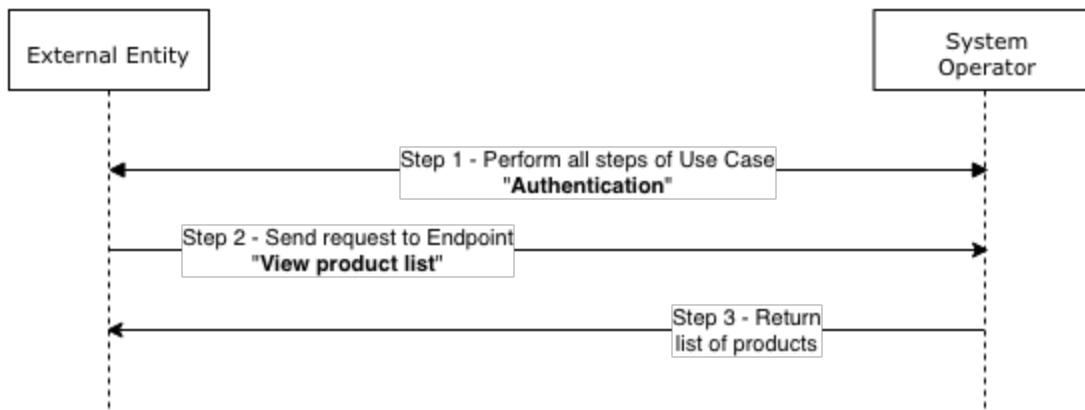
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "name": "string",  
      "description": "string",  
      "measureUnit": {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
      }  
    },  
    ],  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0  
  }  
}
```

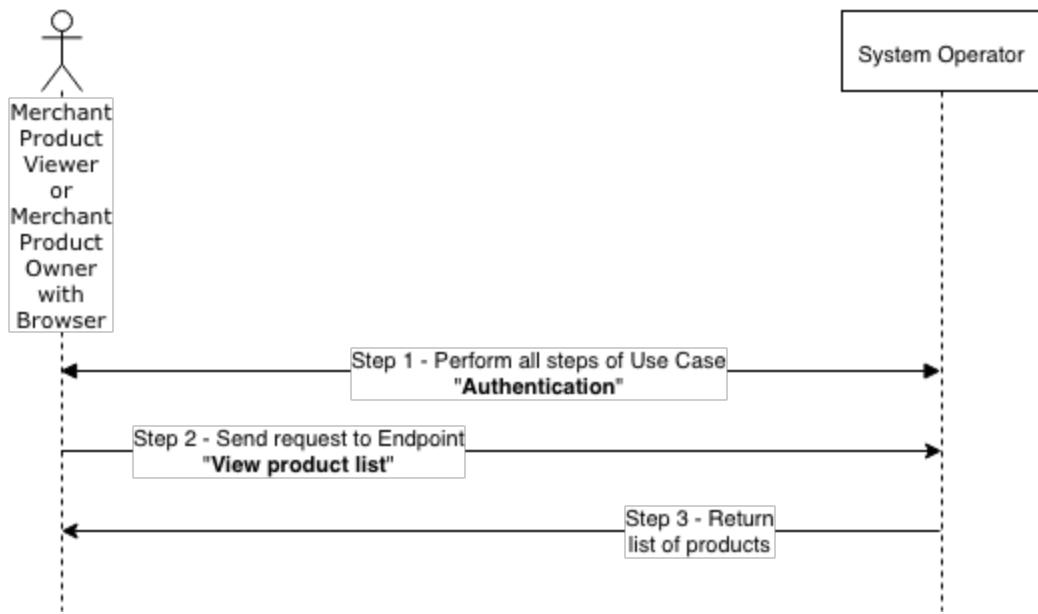
View product list scheme

Use case: View product list

Basic FFlow



Optional Web UI Flow



Merchant product measure units

Create a measure unit description

Use Case Name

Create a measure unit

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Create a measure unit".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create a measure unit".

Endpoint URL: POST /merchant-measure-units

Parameters: {

```
"externalCode": "abc-1224",
"codes": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
]
```

1. System Operator returns new measure unit information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a measure unit”.

Endpoint URL: POST /merchant-measure-units

Parameters: {

 "externalCode": "abc-1224",

 "codes": [

 {

 "locale": "en",

 "value": "translated_string_value"

 }

],

 "descriptions": [

 {

 "locale": "en",

 "value": "translated_string_value"

 }

]

}

1. System Operator returns new measure unit information to User (See Result example below).

Post Conditions

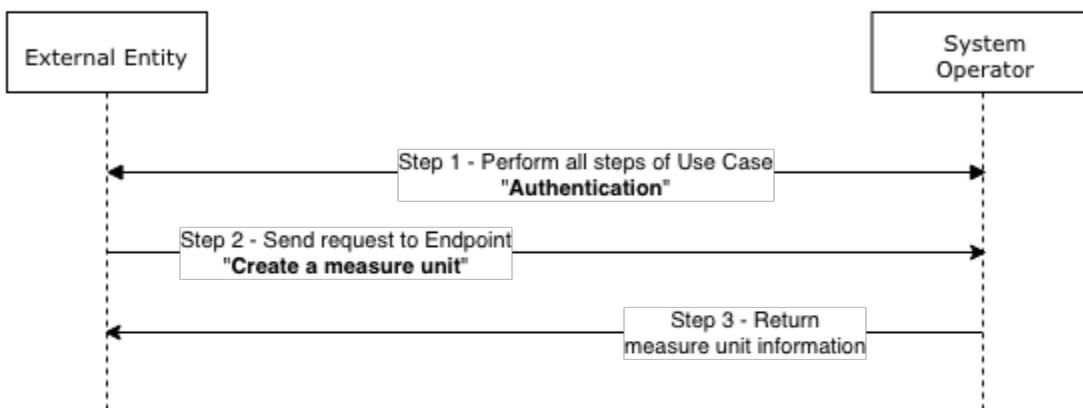
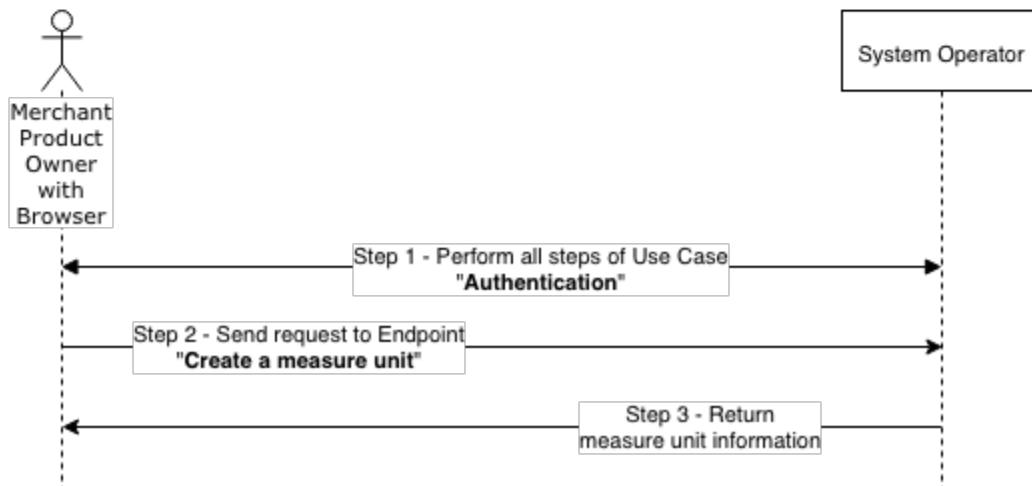
Input parameters for external code are available

Result example

```
{
  "measureUnit": {
    "id": 0,
    "externalCode": "string",
    "code": "string",
    "description": "string"
  },
  "status": "ok",
  "message": "string"
}
```

{}

Create a measure unit scheme

Use case: Create a measure unit**Basic FLow****Optional Web UI Flow**

Create a measure unit from batch description

Use Case Name

Create a measure unit from batch

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Create a measure unit from batch".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Create a measure unit from batch".

Endpoint URL: POST /merchant-measure-units/batch-create

Parameters: {

```
"units": [
  {
    "externalCode": "abc-1224",
    "codes": [
      {
        "locale": "en",
        "value": "translated_string_value"
      }
    ],
    "descriptions": [
      {
        "locale": "en",
        "value": "translated_string_value"
      }
    ]
}
```

```

    ]
}

]
}

```

1. System Operator returns list of new measure units to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a measure unit from batch”.

Endpoint URL: POST /merchant-measure-units/batch-create

Parameters: {

```

"units": [
{
  "externalCode": "abc-1224",
  "codes": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ],
  "descriptions": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ]
}
]
```

1. System Operator returns list of new measure units to User (See Result example below).

Post Conditions

Input parameters for external code are available.

Result example

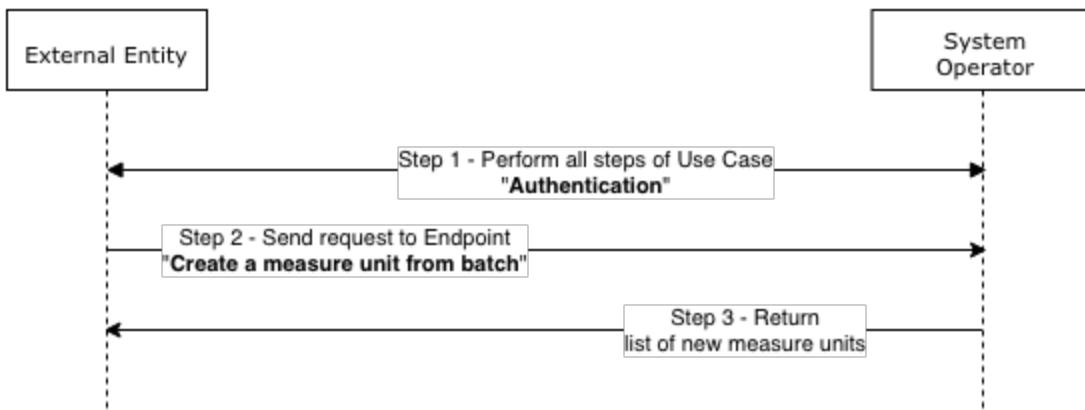
```
{
```

```
"records": [  
    {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
    }  
,  
    "status": "ok",  
    "message": "string"  
}
```

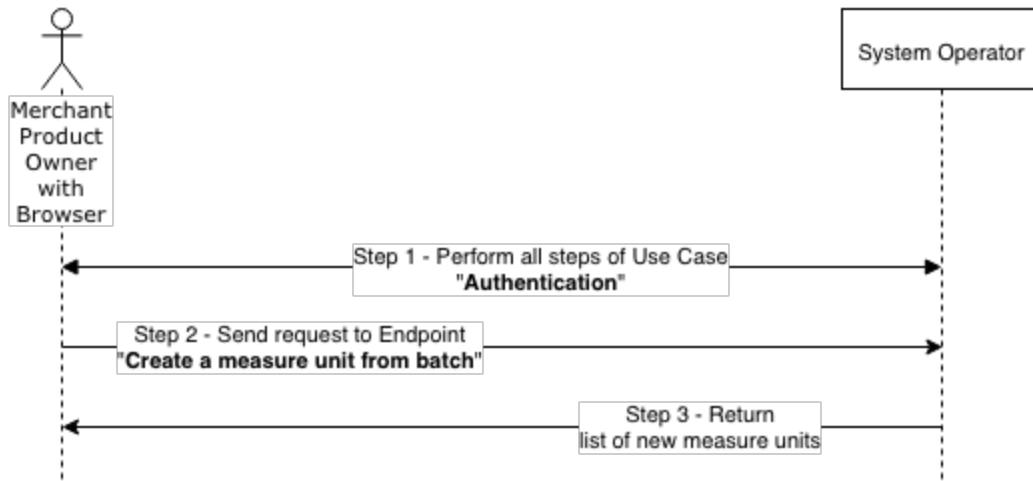
Create a measure unit from batch scheme

Use case: Create a measure unit from batch

Basic FFlow



Optional Web UI Flow



Delete a measure unit by it's external code description

Use Case Name

Delete a measure unit by it's external code

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get list of all measure units belonging to this merchant" Use Case (to obtain external code), and then send a request to Endpoint "Delete a measure unit by it's external code".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get list of all measure units belonging to this merchant".
2. External Entity sends a request to Endpoint "Delete a measure unit by it's external code".

Endpoint URL: POST /merchant-measure-units/delete-by-external-code

Parameters: {

```
"externalCode": "string"
}
```

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get list of all measure units belonging to this merchant".
2. A user sends a request to Endpoint "Delete a measure unit by it's external code".

Endpoint URL: POST /merchant-measure-units/delete-by-external-code

Parameters: {

```
"externalCode": "string"
}
```

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Measure unit is available.

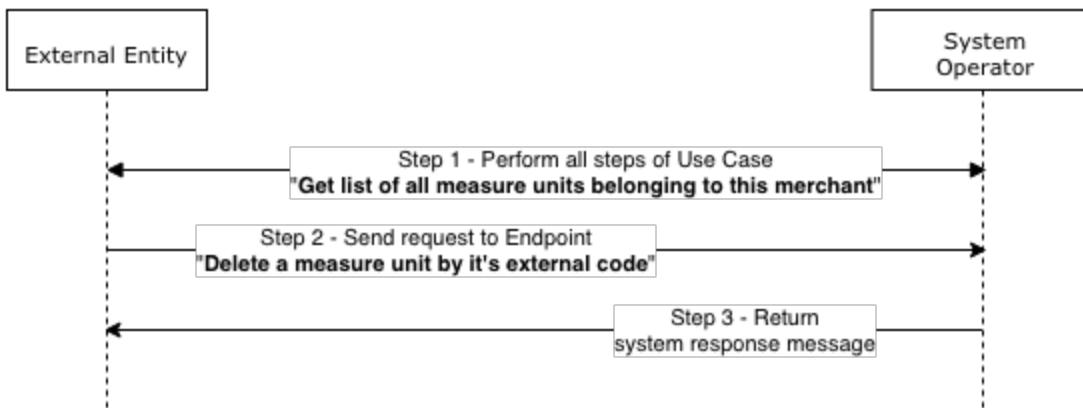
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

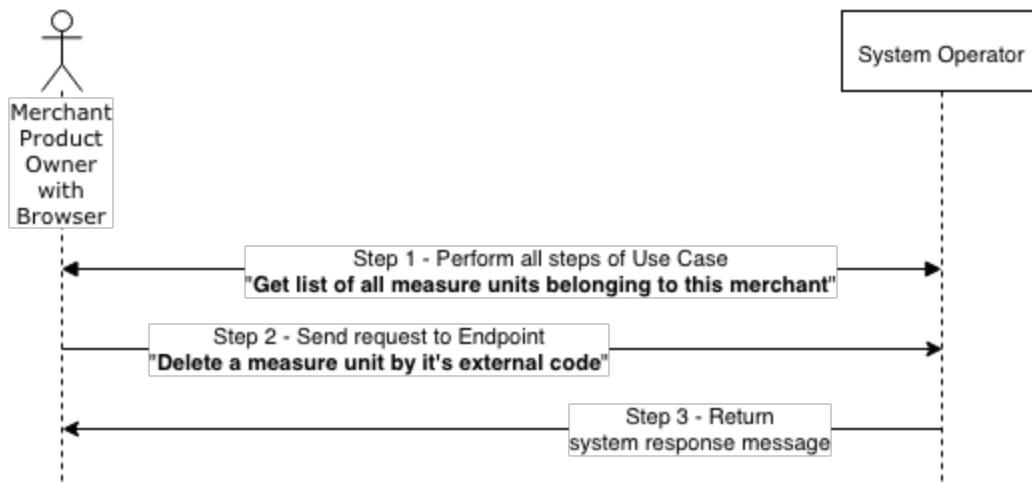
Delete a measure unit by it's external code scheme

Use case: Delete a measure unit by it's external code

Basic FLow



Optional Web UI Flow



Delete a measure unit by its ID description

Use Case Name

Delete a measure unit by it's ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get list of all measure units belonging to this merchant" Use Case (to obtain measure unit ID), and then send a request to Endpoint "Delete a measure unit by it's ID".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get list of all measure units belonging to this merchant".
2. External Entity sends a request to Endpoint "Delete a measure unit by it's ID".

Endpoint URL: DELETE /merchant-measure-units/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get list of all measure units belonging to this merchant".
2. A user sends a request to Endpoint "Delete a measure unit by it's ID".

Endpoint URL: DELETE /merchant-measure-units/{id}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Measure unit is available.

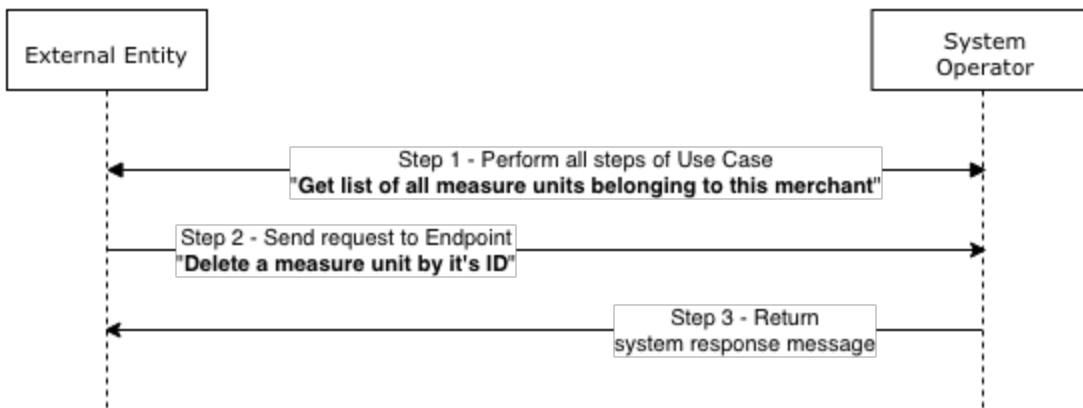
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

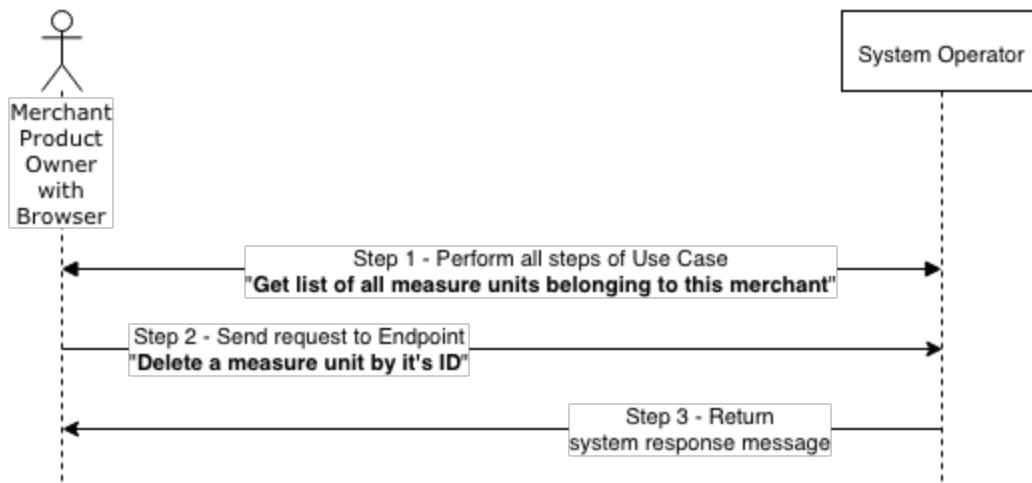
Delete a measure unit by it's ID scheme

Use case: Delete a measure unit by it's ID

Basic FFlow



Optional Web UI Flow



Get list of all measure units belonging to this merchant description

Use Case Name

Get list of all measure units belonging to this merchant

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Authentication" Use Case (to obtain token), and then send a request to Endpoint "Get list of all measure units belonging to this merchant".

Note:

Optional: Accept language - Two-digit language code for localized variables, list of available locale codes can be found at <http://www.oracle.com/technetwork/java/javase/javalocales-2095355.html>.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get list of all measure units belonging to this merchant".

Endpoint URL: GET /merchant-measure-units

Parameters: TOKEN - identifies authenticated user

1. System Operator returns returns list of measure units to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get list of all measure units belonging to this merchant".

Endpoint URL: GET /merchant-measure-units

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of measure units to User (See Result example below).

Post Conditions

Measure unit is available.

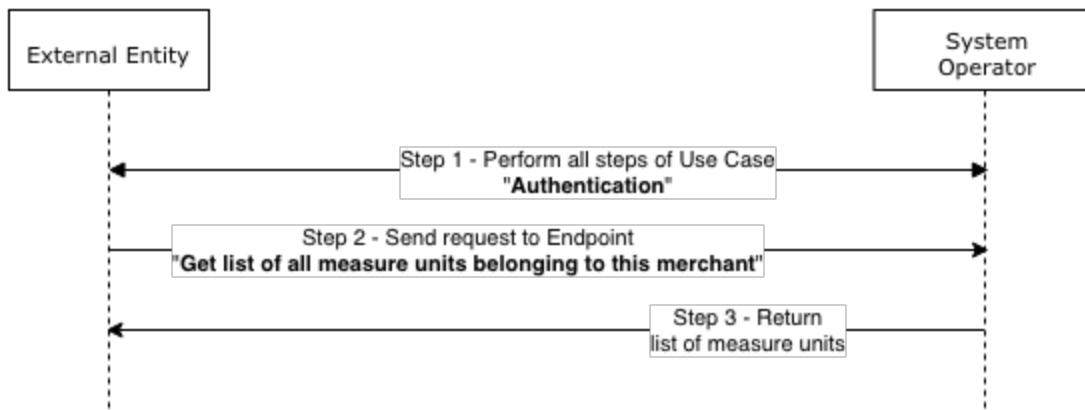
Result example

```
{  
  "records": [  
    {  
      "id": 0,  
      "externalCode": "string",  
      "code": "string",  
      "description": "string"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

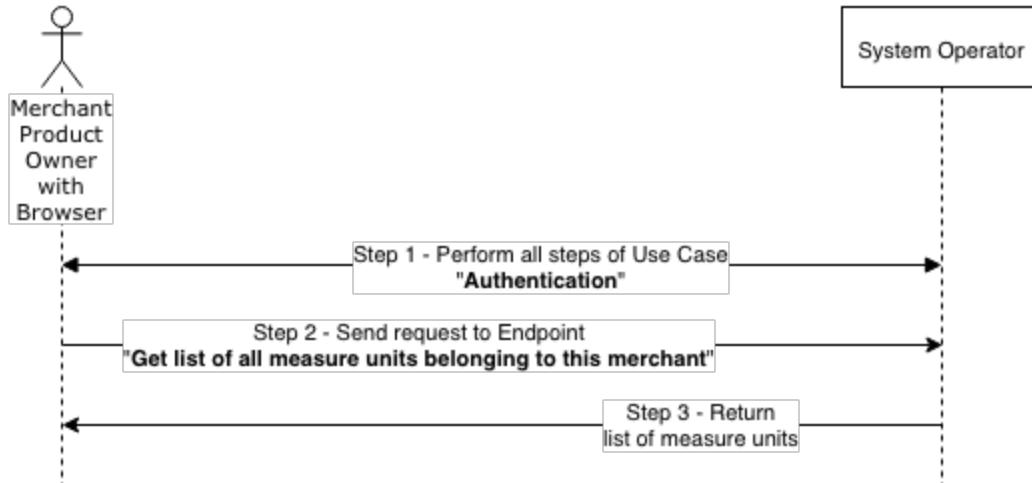
Get list of all measure units belonging to this merchant scheme

Use case: Get list of all measure units belonging to this merchant

Basic FFlow



Optional Web UI Flow



Update measure unit data by it's external code description

Use Case Name

Update measure unit data by it's external code

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get list of all measure units belonging to this merchant" Use Case (to obtain external code), and then send a request to Endpoint "Update measure unit data by it's external code".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get list of all measure units belonging to this merchant".
2. External Entity sends a request to Endpoint "Update measure unit data by it's external code".

Endpoint URL: POST /merchant-measure-units/update-by-external-code

Parameters: {

```
"codes": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"externalCode": "string"
}
```

1. System Operator returns details of updated measure unit to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get list of all measure units belonging to this merchant”.
2. A user sends a request to Endpoint “Update measure unit data by it's external code”.

Endpoint URL: POST /merchant-measure-units/update-by-external-code

```
Parameters: {
  "codes": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ],
  "descriptions": [
    {
      "locale": "en",
      "value": "translated_string_value"
    }
  ],
  "externalCode": "string"
}
```

1. System Operator returns details of updated measure unit to User (See Result example below).

Post Conditions

Measure unit is available.

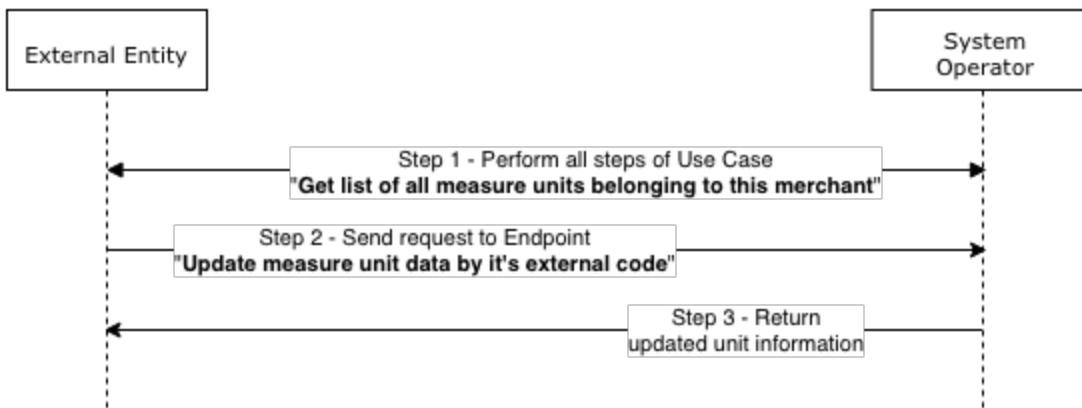
Result example

```
{
  "measureUnit": {
    "id": 0,
    "externalCode": "string",
    "code": "string",
    "description": "string"
  },
  "status": "ok",
  "message": "string"
}
```

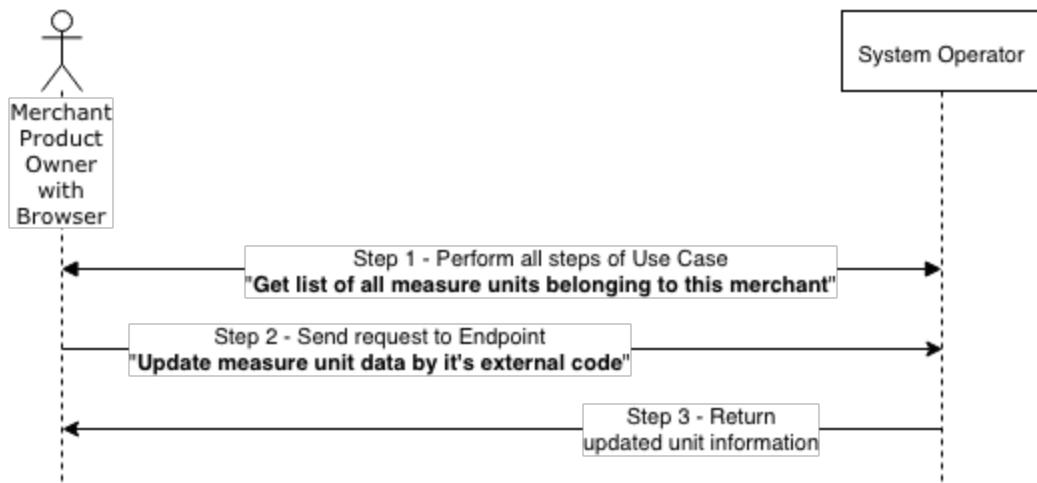
Update measure unit data by it's external code scheme

Use case: Update measure unit data by it's external code

Basic FFlow



Optional Web UI Flow



Update measure unit data by it's ID description

Use Case Name

Update measure unit data by it's ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get list of all measure units belonging to this merchant" Use Case (to obtain unit ID), and then send a request to Endpoint "Update measure unit data by it's ID".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "
2. Get list of all measure units belonging to this merchant".
3. External Entity sends a request to Endpoint "Update measure unit data by it's ID".

Endpoint URL: PATCH /merchant-measure-units/{id}

Parameters: {

"codes": [

{

"locale": "en",

"value": "translated_string_value"

}

],

"descriptions": [

{

"locale": "en",

"value": "translated_string_value"

}

]

}

1. System Operator returns details of updated measure unit to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “
2. Get list of all measure units belonging to this merchant”.
3. A user sends a request to Endpoint “Update measure unit data by its ID”.

Endpoint URL: PATCH /merchant-measure-units/{id}

Parameters: {

```
"codes": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
],
"descriptions": [
  {
    "locale": "en",
    "value": "translated_string_value"
  }
]
```

1. System Operator returns details of updated measure unit to User (See Result example below).

Post Conditions

Measure unit is available.

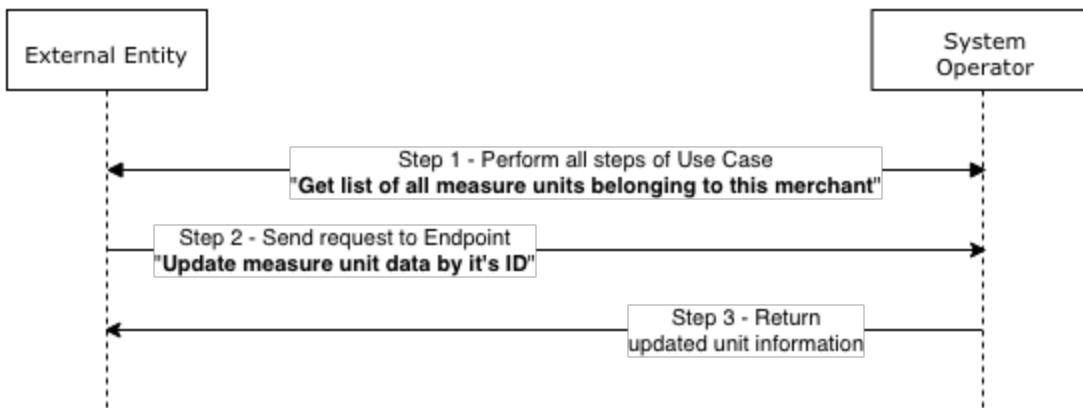
Result example

```
{
  "measureUnit": {
    "id": 0,
    "externalCode": "string",
    "code": "string",
    "description": "string"
  },
  "status": "ok",
  "message": "string"
}
```

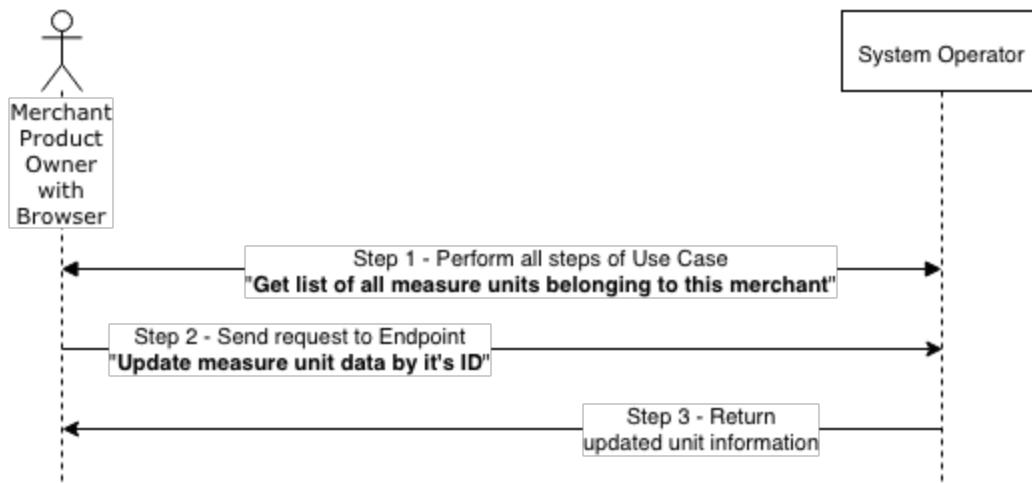
Update measure unit data by its ID scheme

Use case: Update measure unit data by it's ID

Basic Flow



Optional Web UI Flow



Merchant product price

Create prices from batch description

Use Case Name

Create prices from batch

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get prices by product ID" OR "Get prices by product filter" Use Case (to obtain product ID), and then send a request to Endpoint "Create prices from batch".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get prices by product ID" OR "Get prices by product filter".
2. External Entity sends a request to Endpoint "Create prices from batch".

Endpoint URL: POST /merchant-products/batch-create-prices

Parameters: {

```
"prices": [
  {
    "product": {
      "identifier": 0,
      "externalCode": "string"
    },
    "posId": "string",
    "emitterSnPrefix": "string",
    "measureUnit": {
      "identifier": 0,
      "externalCode": "string"
    },
    "dateStart": "2018-09-17T11:07:12.686Z",
    "dateEnd": "2018-09-17T11:07:12.686Z",
    "value": 0
  }
]
```

```

    }
]
}

```

1. System Operator returns list of prices to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get prices by product ID” OR “Get prices by product filter”.
2. A user sends a request to Endpoint “Create prices from batch”.

Endpoint URL: POST /merchant-products/batch-create-prices

Parameters: {

```

"prices": [
{
  "product": {
    "identifier": 0,
    "externalCode": "string"
  },
  "posId": "string",
  "emitentSnPrefix": "string",
  "measureUnit": {
    "identifier": 0,
    "externalCode": "string"
  },
  "dateStart": "2018-09-17T11:07:12.686Z",
  "dateEnd": "2018-09-17T11:07:12.686Z",
  "value": 0
}
]
```

1. System Operator returns list of prices information to User (See Result example below).

Post Conditions

Product is available.

Result example

```
{
  "status": "ok",

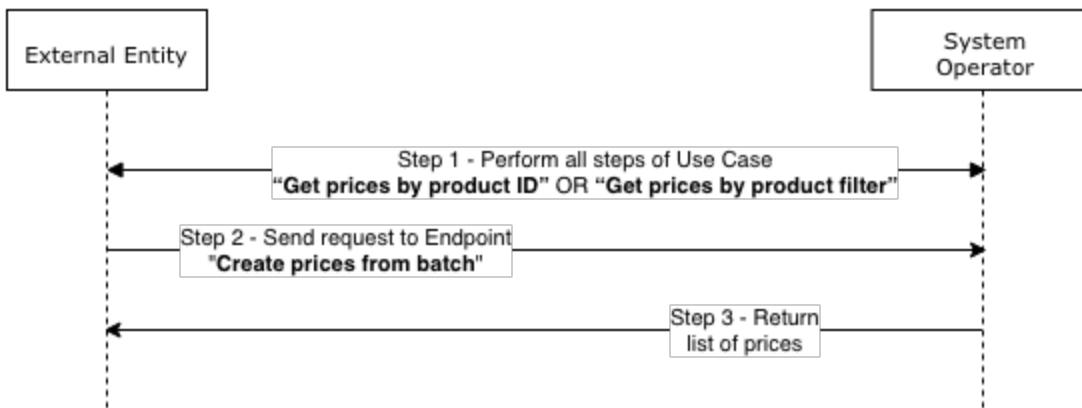
```

```
"message": "string",
"prices": [
{
    "priceId": 0,
    "pos": "string",
    "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
    },
    "measureUnit": {
        "id": 0,
        "externalCode": "string",
        "code": "string",
        "description": "string"
    },
    "dateStart": "2018-09-17T11:07:14.772Z",
    "dateEnd": "2018-09-17T11:07:14.772Z",
    "value": 0
},
]
}
```

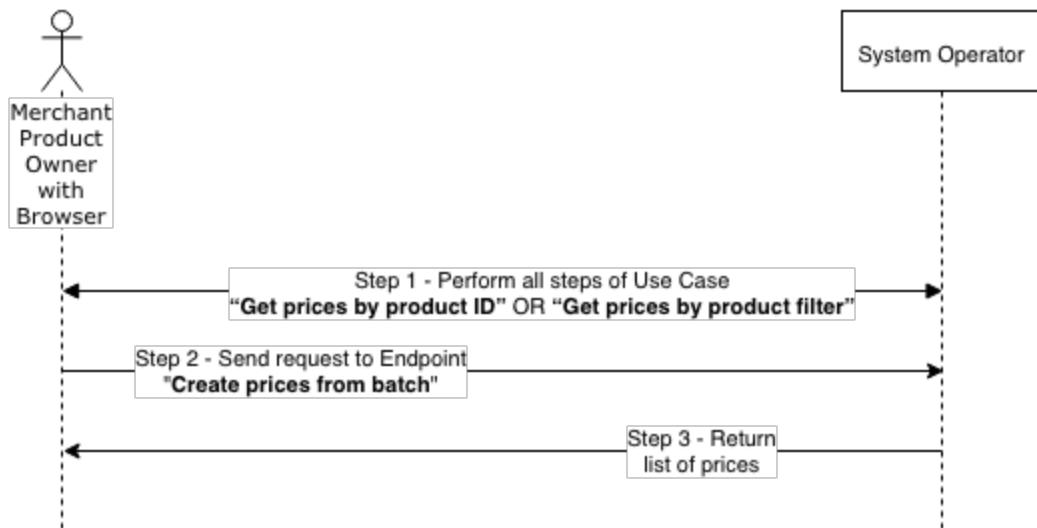
Create prices from batch scheme

Use case: Create prices from batch

Basic FLow



Optional Web UI Flow



Delete a price by it's ID for the given product ID description

Use Case Name

Delete a price by it's ID for the given product ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get prices by product ID" Use Case (to obtain both product ID and product price ID), and then send a request to Endpoint "Delete a price by it's ID for the given product ID".

Note:

Alternatively, you may use "Get prices by product filter" to obtain the product and price IDs.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get prices by product ID".
2. External Entity sends a request to Endpoint "Delete a price by it's ID for the given product ID".

Endpoint URL: DELETE /merchant-products/{productId}/prices/{priceId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get prices by product ID".
2. A user sends a request to Endpoint "Delete a price by it's ID for the given product ID".

Endpoint URL: DELETE /merchant-products/{productId}/prices/{priceId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

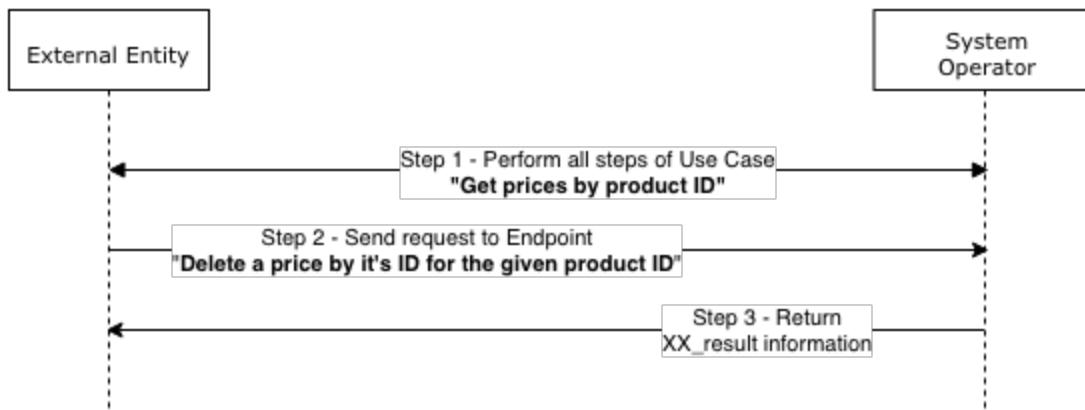
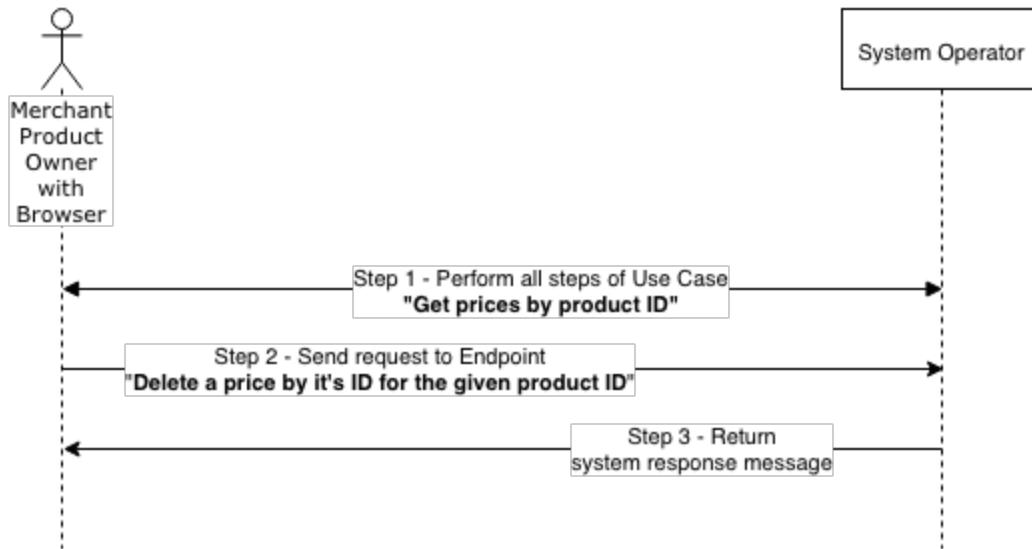
Product is available.

Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

Delete a price by it's ID for the given product ID scheme

Use case: Delete a price by it's ID for the given product ID

Basic FFlow**Optional Web UI Flow**

Delete a price by product filter description

Use Case Name

Delete a price by product filter

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get prices by product filter" Use Case (to obtain product ID), and then send a request to Endpoint "Delete a price by product filter".

Note:

You may filter products by ID, external code and PoS ID. The filter is optional. If it is omitted, all products will be returned.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get prices by product filter".
2. External Entity sends a request to Endpoint "Delete a price by product filter".

Endpoint URL: POST /merchant-products/delete-prices

```
Parameters: {
  "product": {
    "identifier": 0,
    "externalCode": "string"
  }
}
```

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get prices by product filter".
2. A user sends a request to Endpoint "Delete a price by product filter".

Endpoint URL: POST /merchant-products/delete-prices

```
Parameters: {  
  "product": {  
    "identifier": 0,  
    "externalCode": "string"  
  }  
}
```

1. System Operator returns system response message to User (See Result example below).

Post Conditions

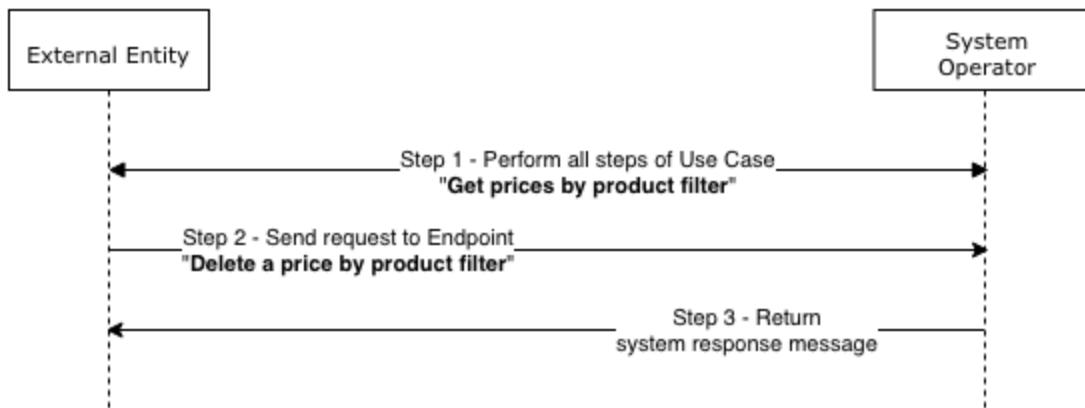
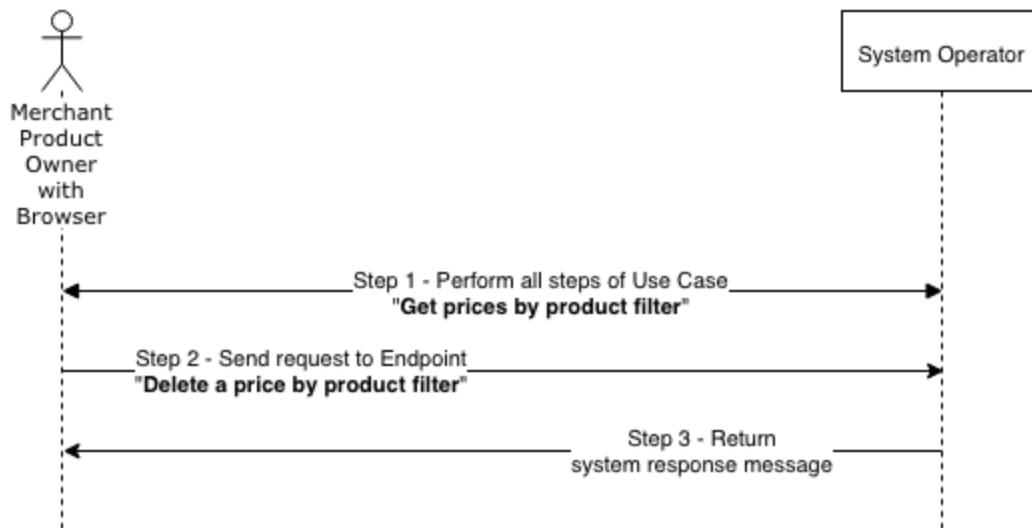
Product is available.

Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

Delete a price by product filter scheme

Use case: Delete a price by product filter

Basic FFlow**Optional Web UI Flow**

Delete all prices for product and point of sale description

Use Case Name

Delete all prices for product and point of sale

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "Get prices by product ID" Use Case (to obtain product ID and body parameters), and then send a request to Endpoint "Delete all prices for product and point of sale".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get prices by product ID".
2. External Entity sends a request to Endpoint "Delete all prices for product and point of sale".

Endpoint URL: POST /merchant-products/delete-prices-for-point-of-sale

Parameters: {

```
"product": {
  "identifier": 0,
  "externalCode": "string"
},
"posId": "string"
}
```

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get prices by product ID".
2. A user sends a request to Endpoint "Delete all prices for product and point of sale".

Endpoint URL: POST /merchant-products/delete-prices-for-point-of-sale

Parameters: {

```
"product": {
  "identifier": 0,
```

```
"externalCode": "string"  
},  
"posId": "string"  
}
```

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Product is available.

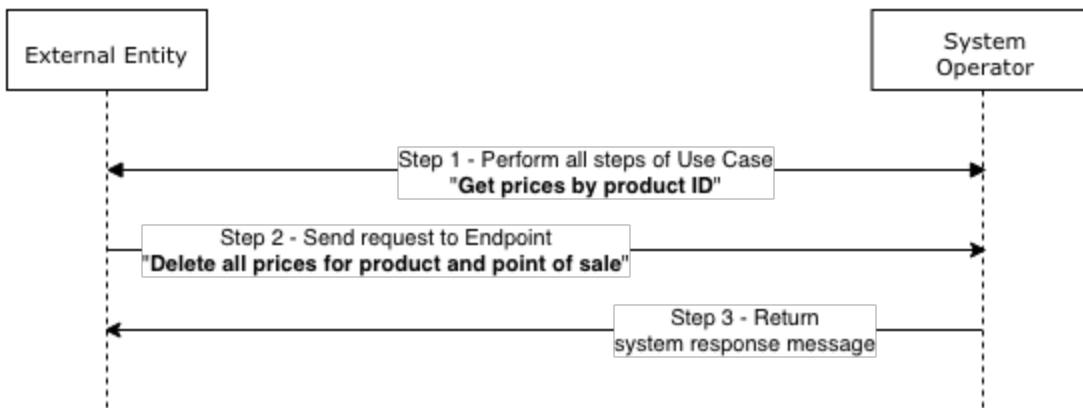
Result example

```
{  
"status": "ok",  
"message": "string"  
}
```

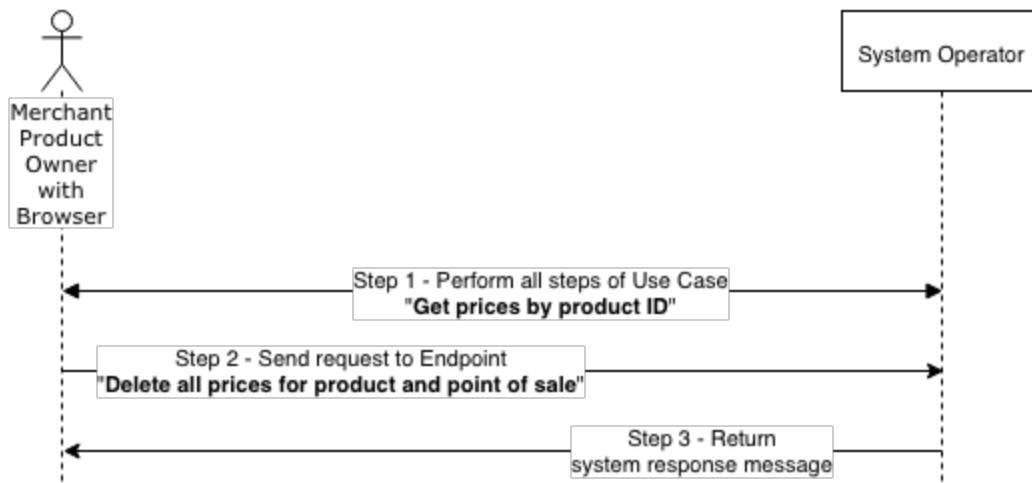
Delete all prices for product and point of sale scheme

Use case: Delete all prices for product and point of sale

Basic FFlow



Optional Web UI Flow



Get prices by product filter description

Use Case Name

Get prices by product filter

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain filters), and then send a request to Endpoint "Get prices by product filter".

Notes:

- The filter options include product ID, external code and PoS IDs.
- Optional: Accept language - Two-digit language code for localized variables, list of available locale codes can be found at <http://www.oracle.com/technetwork/java/javase/java8locales-2095355.html>.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "Get prices by product filter".

Endpoint URL: POST /merchant-products/view-prices

```
Parameters: {
  "product": {
    "identifier": 0,
    "externalCode": "string"
  },
  "posIdList": [
    "string"
  ]
}
```

1. System Operator returns list of prices (and details of measure units, PoS ID, and issuer) for a product to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View product list”.
2. A user sends a request to Endpoint “Get prices by product filter”.

Endpoint URL: POST /merchant-products/view-prices

Parameters: {

```
"product": {
  "identifier": 0,
  "externalCode": "string"
},
"posIdList": [
  "string"
}
```

1. System Operator returns list of prices (and details of measure units, PoS ID, and issuer) for a product to User (See Result example below).

Post Conditions

Product is available.

Result example

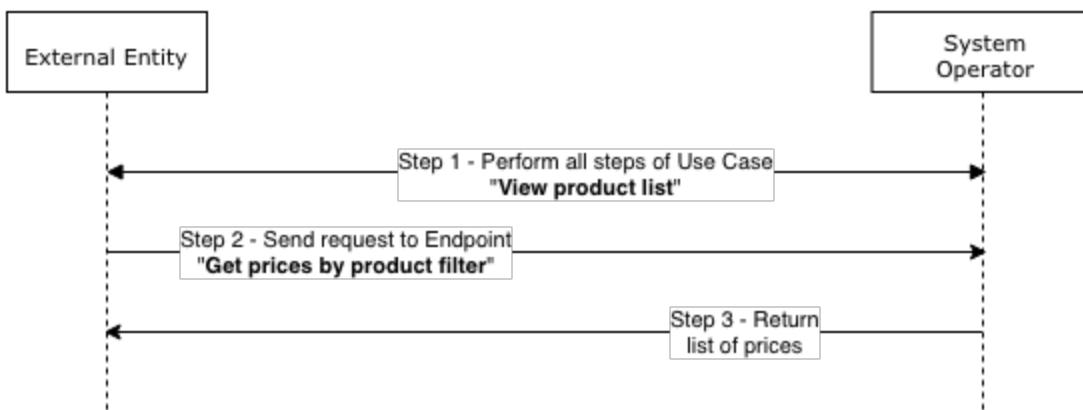
```
{
  "status": "ok",
  "message": "string",
  "prices": [
    {
      "priceId": 0,
      "pos": "string",
      "issuer": {
        "id": "string",
        "sn": "string",
        "currency": "string"
      },
      "measureUnit": {
        "id": 0,
        "externalCode": "string",
        "code": "string",
        "description": "string"
      },
      "dateStart": "2018-09-17T11:07:14.785Z",
      "dateEnd": "2018-09-17T11:07:14.785Z",
      "unit": "string"
    }
  ]
}
```

```
        "value": 0  
    }  
]  
}
```

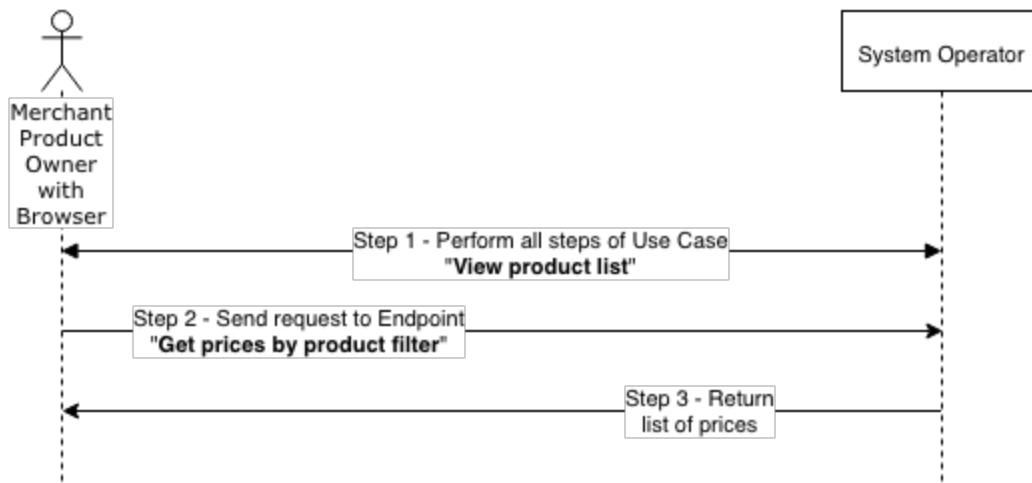
Get prices by product filter scheme

Use case: Get prices by product filter

Basic FFlow



Optional Web UI Flow



Get prices by product ID description

Use Case Name

Get prices by product ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain product ID), and then send a request to Endpoint "Get prices by product ID".

Note:

Optional: Accept language - Two-digit language code for localized variables, list of available locale codes can be found at <http://www.oracle.com/technetwork/java/javase/javalocales-2095355.html>.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_VIEWER" or "MERCHANT_PRODUCT_OWNER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "Get prices by product ID".

Endpoint URL: GET /merchant-products/{productId}/prices

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product prices to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product list".
2. A user sends a request to Endpoint "Get prices by product ID".

Endpoint URL: GET /merchant-products/{productId}/prices

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product prices to User (See Result example below).

Post Conditions

Product is available.

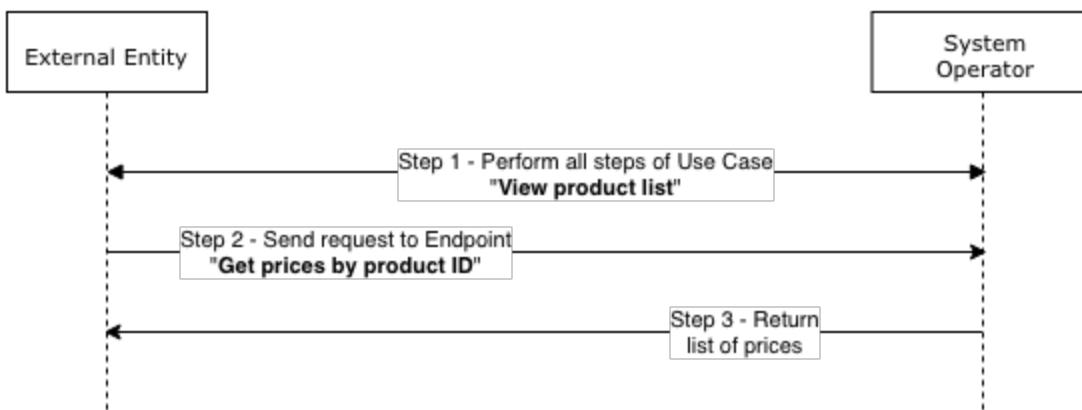
Result example

```
{  
  "status": "ok",  
  "message": "string",  
  "prices": [  
    {  
      "priceId": 0,  
      "pos": "string",  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "measureUnit": {  
        "id": 0,  
        "externalCode": "string",  
        "code": "string",  
        "description": "string"  
      },  
      "dateStart": "2018-09-17T11:07:14.792Z",  
      "dateEnd": "2018-09-17T11:07:14.792Z",  
      "value": 0  
    }  
  ]  
}
```

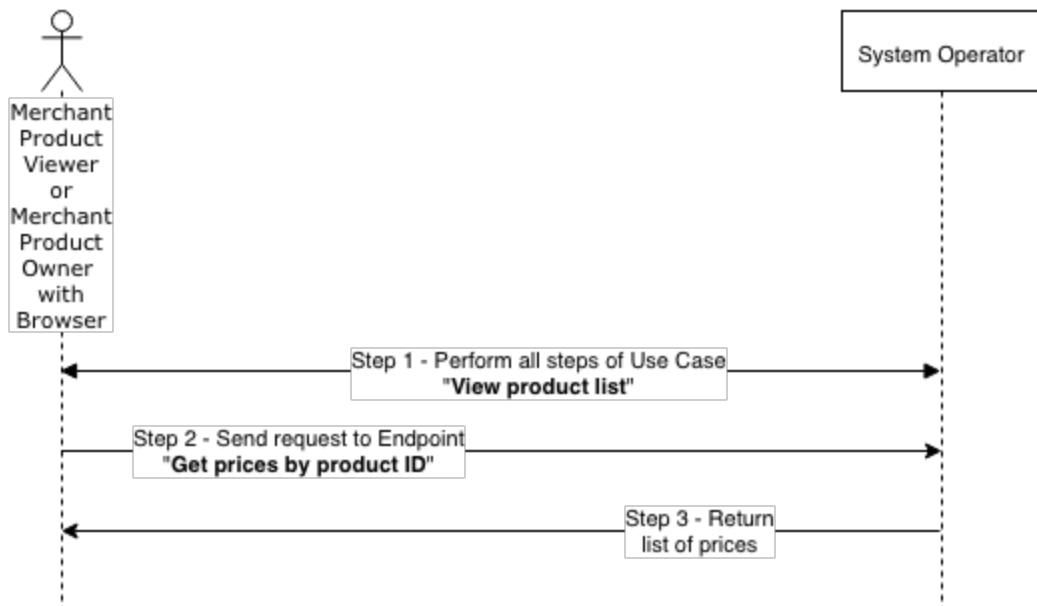
Get prices by product ID scheme

Use case: Get prices by product ID

Basic FFlow



Optional Web UI Flow



Add price for a product with the given ID description

Use Case Name

Add price for a product with the given ID

Brief Description

A User or External Entity on behalf of a User with role permission "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product list" Use Case (to obtain product ID), and then send a request to Endpoint "Add price for a product with the given ID".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "MERCHANT_PRODUCT_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product list".
2. External Entity sends a request to Endpoint "Add price for a product with the given ID".

Endpoint URL: POST /merchant-products/{productId}/prices

Parameters: {

```
"posId": "string",
"issuerId": "string",
"measureUnit": {
    "identifier": 0,
    "externalCode": "string"
},
"dateStart": "2018-09-17T11:07:12.737Z",
"dateEnd": "2018-09-17T11:07:12.737Z",
"value": 0
}
```

1. System Operator returns details of new price result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View product list”.
2. A user sends a request to Endpoint “Add price for a product with the given ID”.

Endpoint URL: POST /merchant-products/{productId}/prices

```
Parameters: {
  "posId": "string",
  "issuerId": "string",
  "measureUnit": {
    "identifier": 0,
    "externalCode": "string"
  },
  "dateStart": "2018-09-17T11:07:12.737Z",
  "dateEnd": "2018-09-17T11:07:12.737Z",
  "value": 0
}
```

1. System Operator returns details of new price information to User (See Result example below).

Post Conditions

Product is available.

Result example

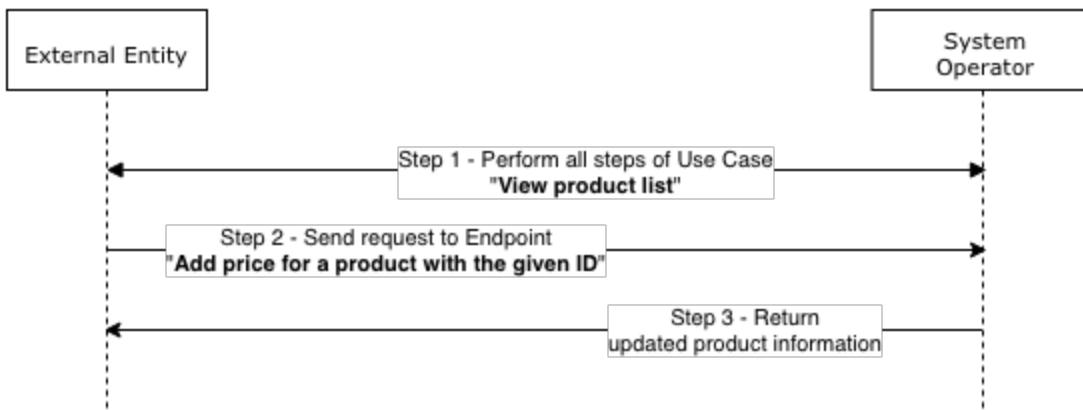
```
{
  "price": {
    "priceId": 0,
    "pos": "string",
    "issuer": {
      "id": "string",
      "sn": "string",
      "currency": "string"
    },
    "measureUnit": {
      "id": 0,
      "externalCode": "string",
      "code": "string",
      "description": "string"
    },
    "dateStart": "2018-09-17T11:07:14.798Z",
    "dateEnd": "2018-09-17T11:07:14.798Z"
  }
}
```

```
"dateEnd": "2018-09-17T11:07:14.798Z",
"value": 0
},
"status": "ok",
"message": "string"
}
```

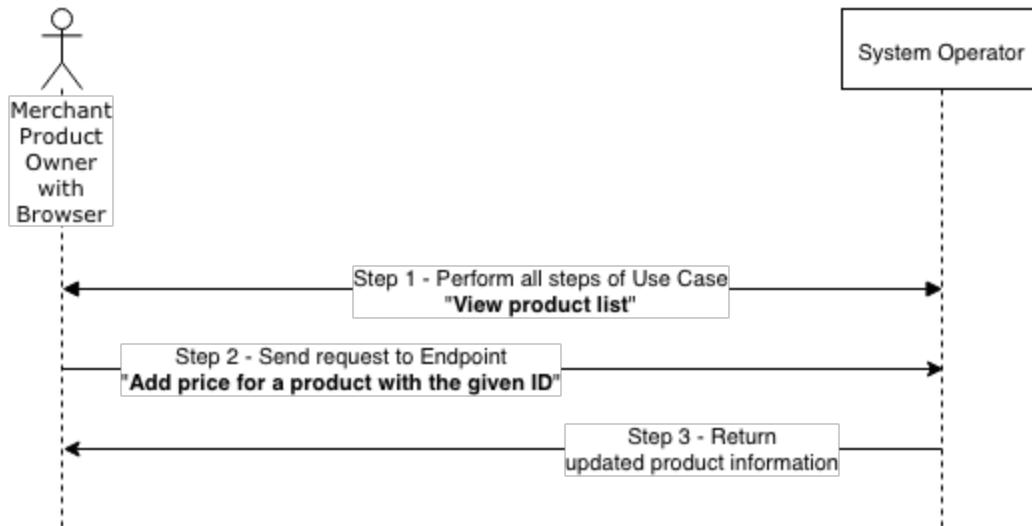
Add price for a product with the given ID scheme

Use case: Add price for a product with the given ID

Basic FFlow



Optional Web UI Flow



Loyalty groups management

Create a new loyalty group description

Use Case Name

Create a new loyalty group

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Get profile fields eligible to use during loyalty group creation" Use Case, and then send a request to Endpoint "Create a new loyalty group". The first call provides a list of parameters that could be used inside the {} condition.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get profile fields eligible to use during loyalty group creation".
2. External Entity sends a request to Endpoint "Create a new loyalty group".

Endpoint URL: POST /loyalty-groups

```
Parameters: {
  "name": "string",
  "conditions": {}
}
```

1. System Operator returns information about the new loyalty group created to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get profile fields eligible to use during loyalty group creation".
2. A user sends a request to Endpoint "Create a new loyalty group".

Endpoint URL: POST /loyalty-groups

```
Parameters: {
  "name": "string",
  "conditions": {}
}
```

1. System Operator returns information about the new loyalty group created to User (See Result example below).

Post Conditions

Loyalty group conditions are available.

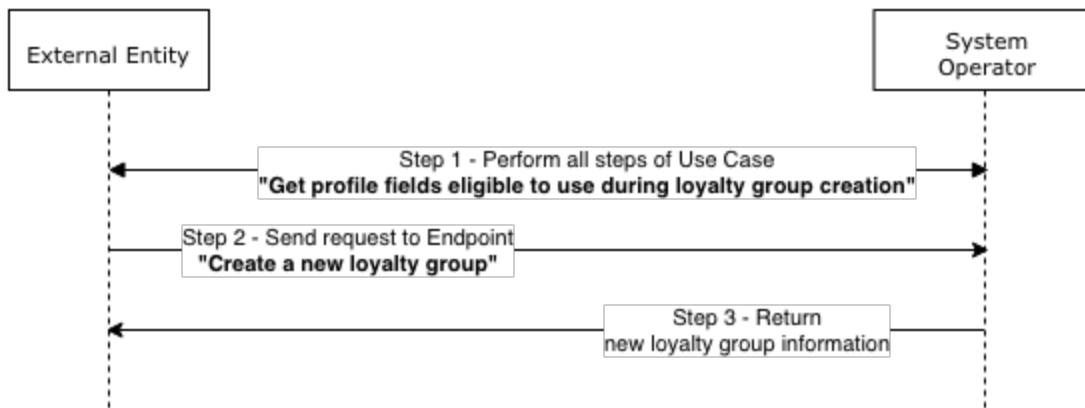
Result example

```
{  
  "group": {  
    "id": "string",  
    "createdAt": "2018-09-13T10:50:37.232Z",  
    "name": "string",  
    "conditions": {}  
  },  
  "status": "ok",  
  "message": "string"  
}
```

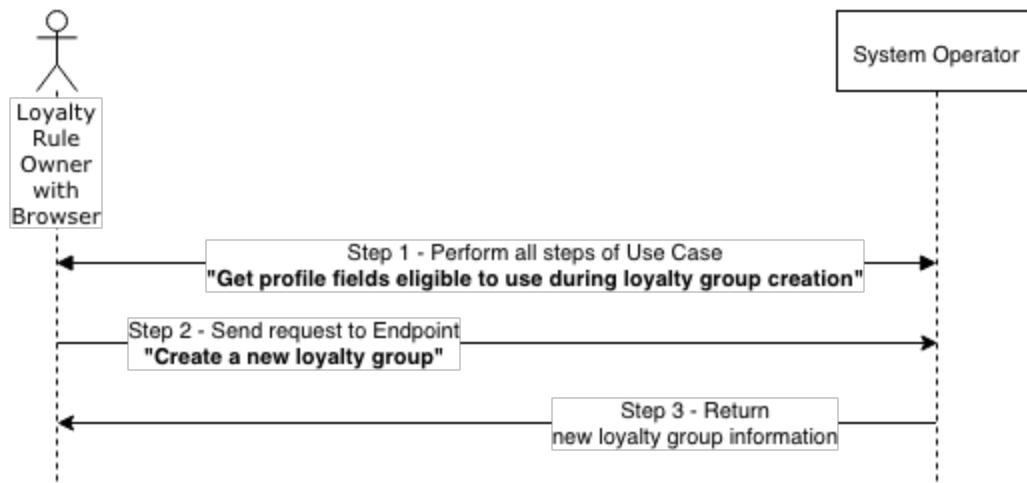
Create a new loyalty group scheme

Use case: Create a new loyalty group

Basic FFlow



Optional Web UI Flow



Delete loyalty group description

Use Case Name

Delete loyalty group

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Get loyalty groups" Use Case (to obtain Group ID), and then send a request to Endpoint "Delete loyalty group".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Get loyalty groups".
2. External Entity sends a request to Endpoint "Delete loyalty group".

Endpoint URL: DELETE /loyalty-groups/{groupId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Get loyalty groups".
2. A user sends a request to Endpoint "Delete loyalty group".

Endpoint URL: DELETE /loyalty-groups/{groupId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response to User (See Result example below).

Post Conditions

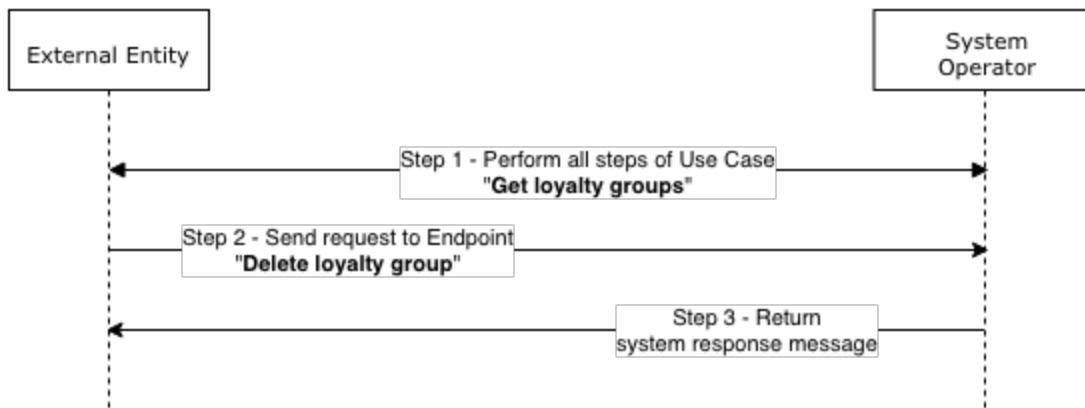
Group to be deleted is available.

Result example

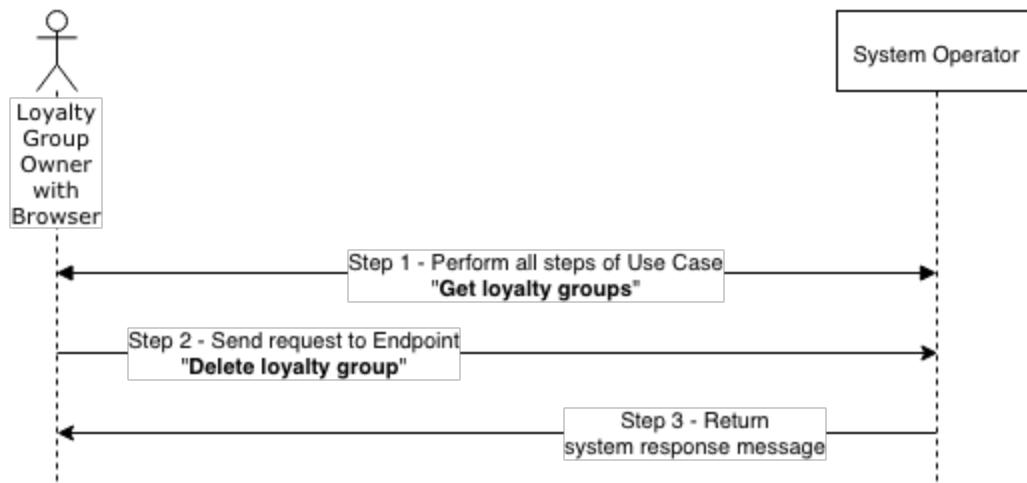
```
{  
  "status": "ok",  
  "message": "string"  
}  
Delete loyalty group scheme
```

Use case: Delete loyalty group

Basic FFlow



Optional Web UI Flow



Get loyalty groups description

Use Case Name

Get loyalty groups

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get loyalty groups".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to User (See Result example below).

Post Conditions

One or more loyalty groups are available.

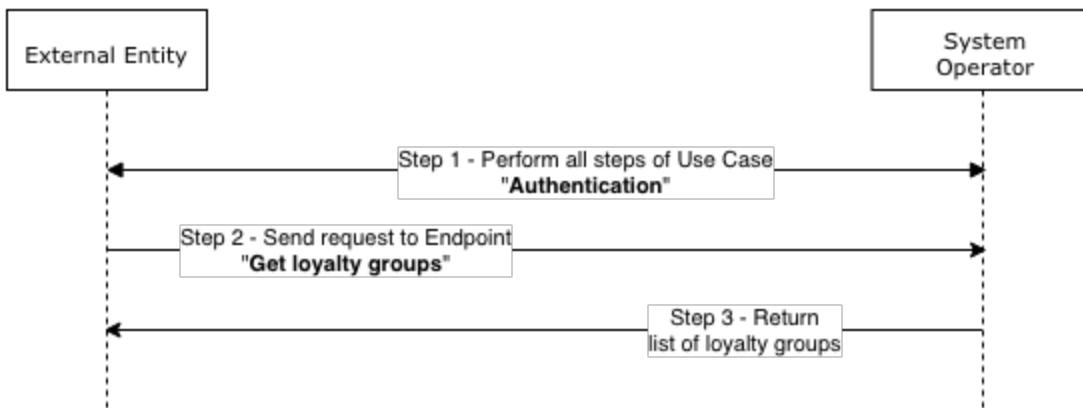
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-08-31T14:37:04.117Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

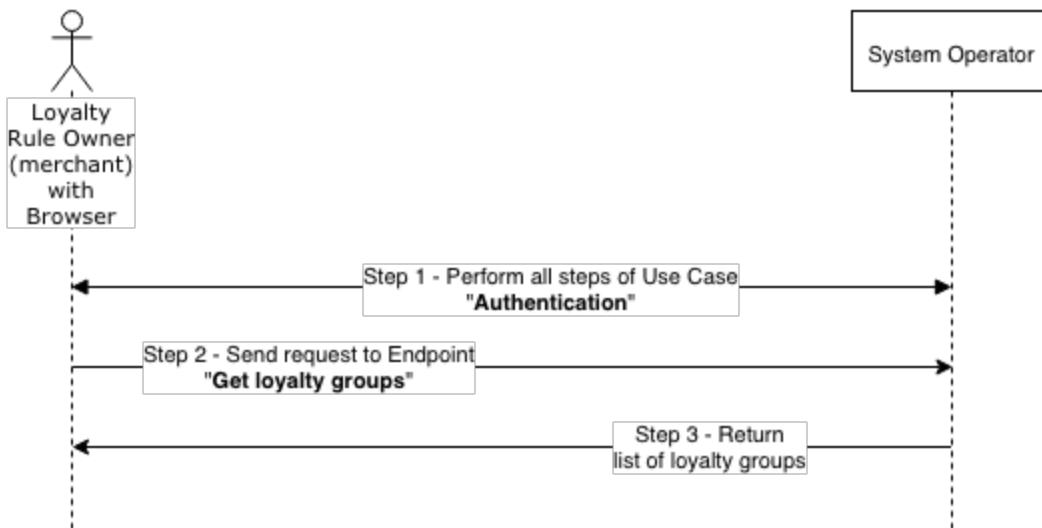
Get loyalty groups scheme

Use case: Get loyalty groups

Basic FFlow



Optional Web UI Flow



Get profile fields eligible to use during loyalty group creation description

Use Case Name

Get loyalty groups

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get loyalty groups".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to User (See Result example below).

Post Conditions

One or more loyalty groups are available.

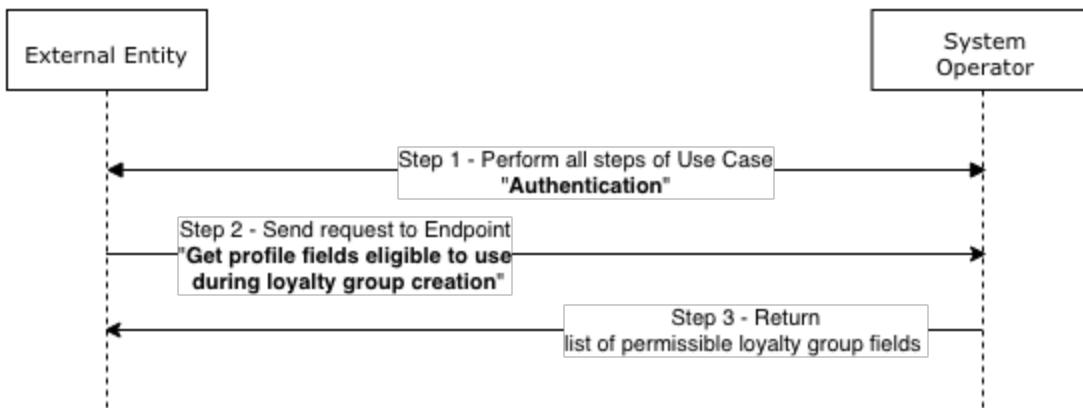
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-08-31T14:37:04.117Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

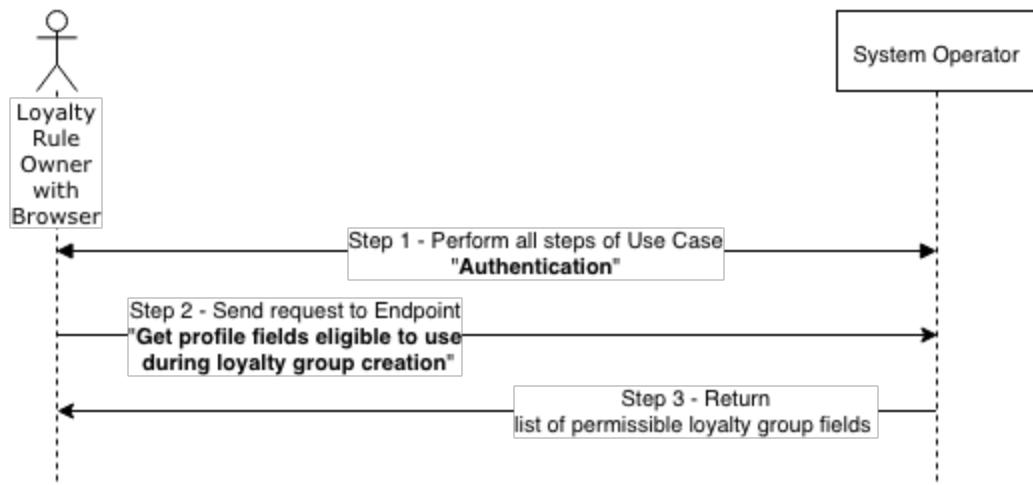
Get profile fields eligible to use during loyalty group creation scheme

Use case: Get profile fields eligible to use during loyalty group creation

Basic FLow



Optional Web UI Flow



Loyalty rule management

Add issuer to the loyalty rule description

Use Case Name

Get loyalty groups

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get loyalty groups".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to User (See Result example below).

Post Conditions

One or more loyalty groups are available.

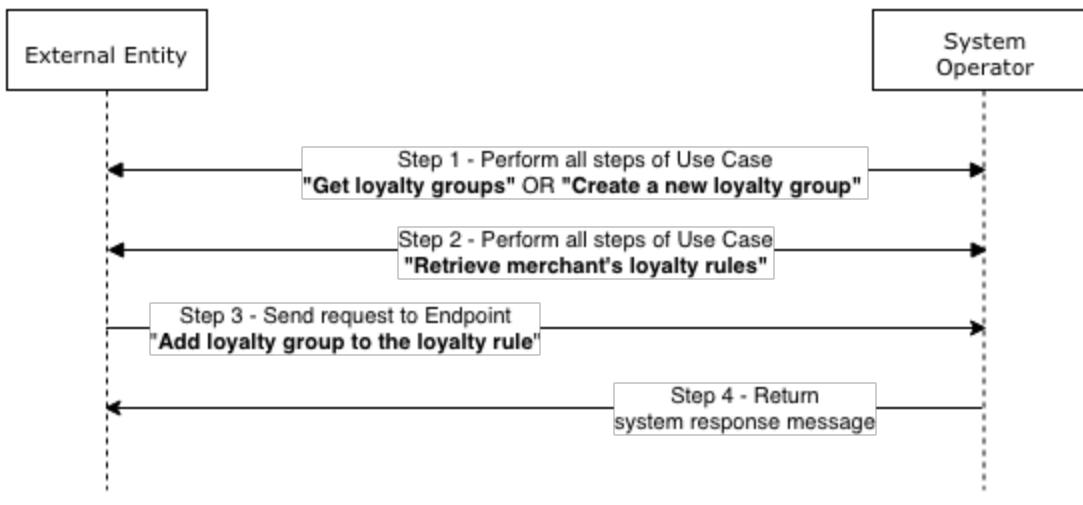
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-08-31T14:37:04.117Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

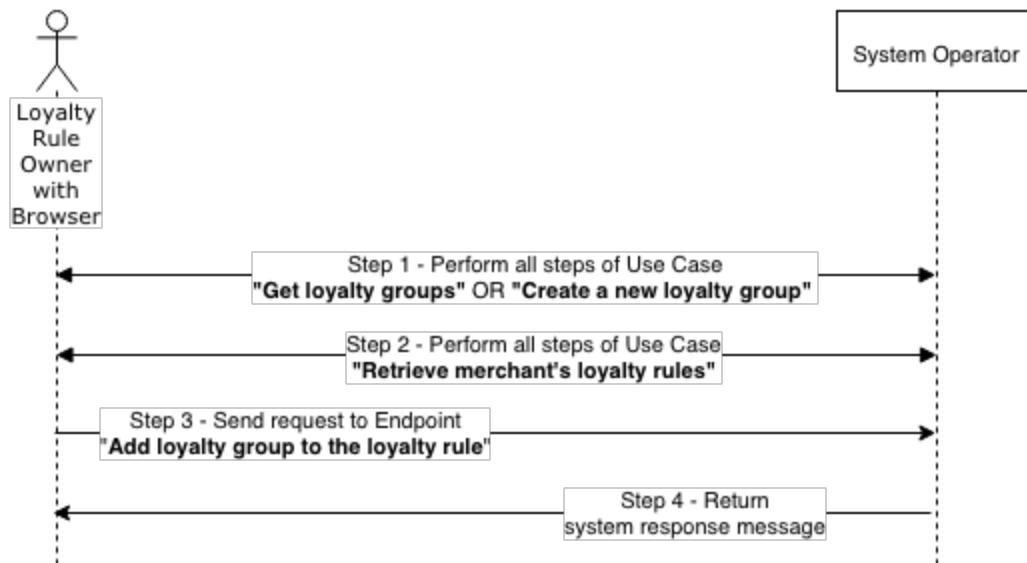
Add issuer to the loyalty rule scheme

Use case: Add loyalty group to the loyalty rule

Basic FFlow



Optional Web UI Flow



Add loyalty group to the loyalty rule description

Use Case Name

Get loyalty groups

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get loyalty groups".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to User (See Result example below).

Post Conditions

One or more loyalty groups are available.

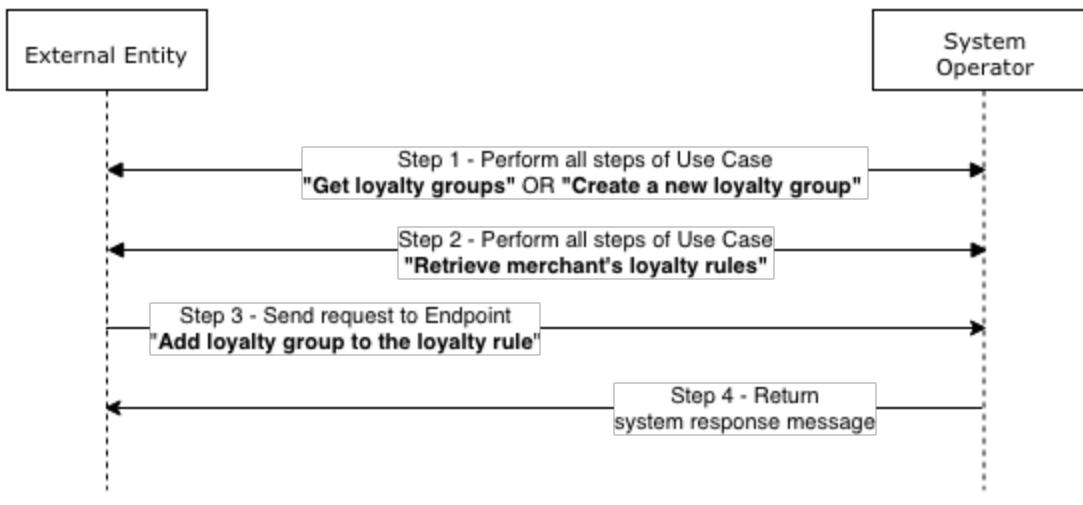
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-08-31T14:37:04.117Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

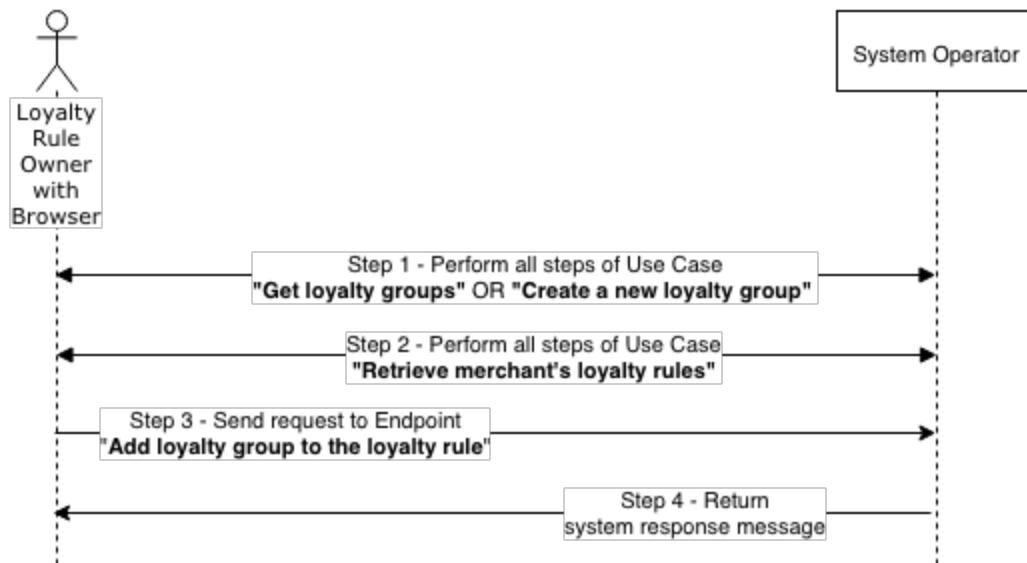
Add loyalty group to the loyalty rule scheme

Use case: Add loyalty group to the loyalty rule

Basic FFlow



Optional Web UI Flow



Add merchant product category to the loyalty rule description

Use Case Name

Add merchant product category to the loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permissions "LOYALTY_RULE_OWNER" and "MERCHANT_PRODUCT_OWNER" will go through all steps of "View product categories belonging to this merchant" OR "Create a product category" Use Case (to obtain category ID), AND "Retrieve merchant's loyalty rules" Use Case (to obtain rule ID), and then send a request to Endpoint "Add merchant product category to the loyalty rule".

Note:

"View product categories belonging to this merchant" and "Create a product category" Use Cases require "MERCHANT_PRODUCT_OWNER" role permission.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View product categories belonging to this merchant" OR "Create a product category".
2. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
3. External Entity sends a request to Endpoint "Add merchant product category to the loyalty rule".

Endpoint URL: PUT /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

Parameters: {

```
"valueType": "NOT_SPECIFIED",
"value": 0
```

1. System Operator returns information about product category added to rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View product categories belonging to this merchant" OR "Create a product category".
2. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
3. A user sends a request to Endpoint "Add merchant product category to the loyalty rule".

Endpoint URL: PUT /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

```
Parameters: {
  "valueType": "NOT_SPECIFIED",
  "value": 0
}
```

1. System Operator returns information about product category added to rule to User (See Result example below).

Post Conditions

Loyalty rule and product category are available.

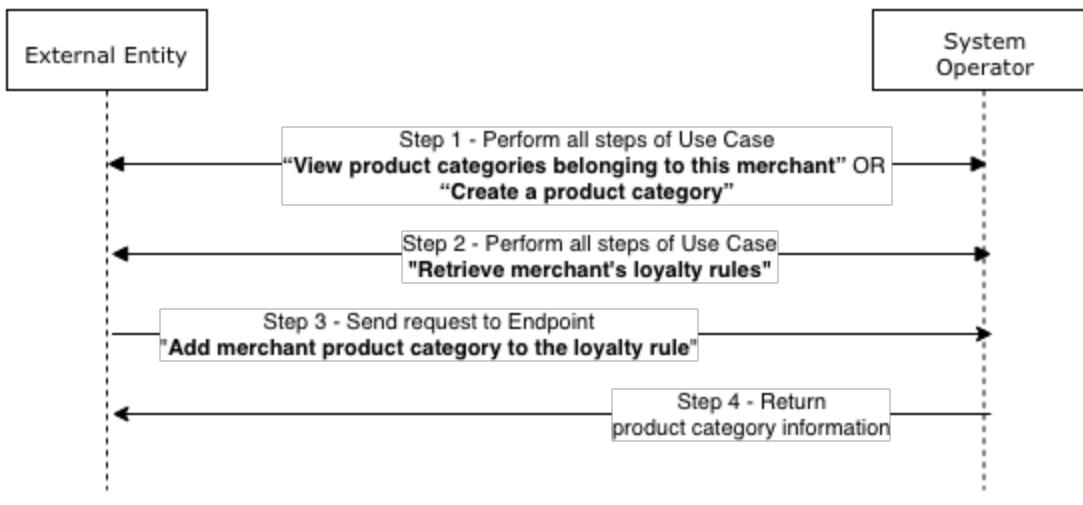
Result example

```
{
  "category": {
    "category": {
      "id": 0,
      "externalCode": "string",
      "name": "string",
      "description": "string"
    },
    "valueType": "NOT_SPECIFIED",
    "value": 0
  },
  "status": "ok",
  "message": "string"
}
```

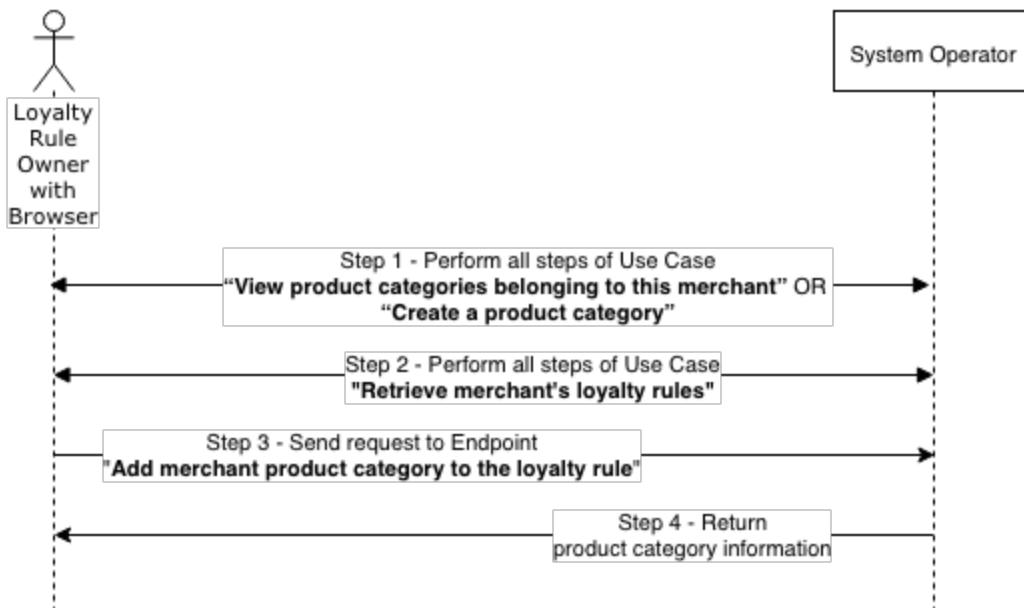
Add merchant product category to the loyalty rule scheme

Use case: Add merchant product category to the loyalty rule

Basic FFlow



Optional Web UI Flow



Add merchant product to the loyalty rule description

Use Case Name

Add merchant product to the loyalty rule

Brief Description

A User or External Entity will go through all steps of “View product list” OR “Create one product” Use Case (to obtain product ID), and then “Retrieve merchant’s loyalty rules” Use Case (to obtain loyalty rule ID). A user, with role permission LOYALTY_RULE_OWNER, will then send a request to Endpoint “Add merchant product to the loyalty rule”.

Note:

“View product list” OR “Create one product” Use Cases require a role permission of either MERCHANT_PRODUCT_VIEWER or MERCHANT_PRODUCT_OWNER.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permission: LOYALTY_RULE_OWNER, e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View product list” OR “Create one product”.
2. Perform all steps of Use Case “Retrieve merchant’s loyalty rules”.
3. External Entity sends a request to Endpoint “Add merchant product to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/merchant-products/{productId}

Parameters: {
 "valueType": "NOT_SPECIFIED",
 "value": 0
 }

1. System Operator returns information about the product added to rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View product list” OR “Create one product”.
2. Perform all steps of Use Case “Retrieve merchant’s loyalty rules”.
3. A user sends a request to Endpoint “Add merchant product to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/merchant-products/{productId}

Parameters: {
 "valueType": "NOT_SPECIFIED",
 "value": 0
 }

1. System Operator returns information about the product added to rule to User (See Result example below).

Post Conditions

Product ID and rule ID are available.

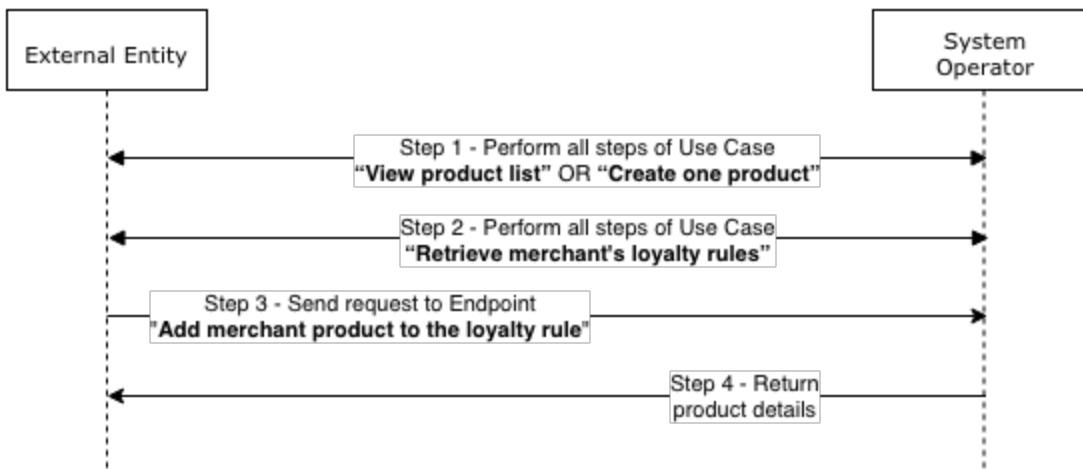
Result example

```
{
  "product": {
    "product": {
      "id": 0,
      "externalCode": "string",
      "name": "string",
      "description": "string",
      "measureUnit": {
        "id": 0,
        "externalCode": "string",
        "code": "string",
        "description": "string"
      }
    },
    "valueType": "NOT_SPECIFIED",
    "value": 0
  },
  "status": "ok",
  "message": "string"
}
```

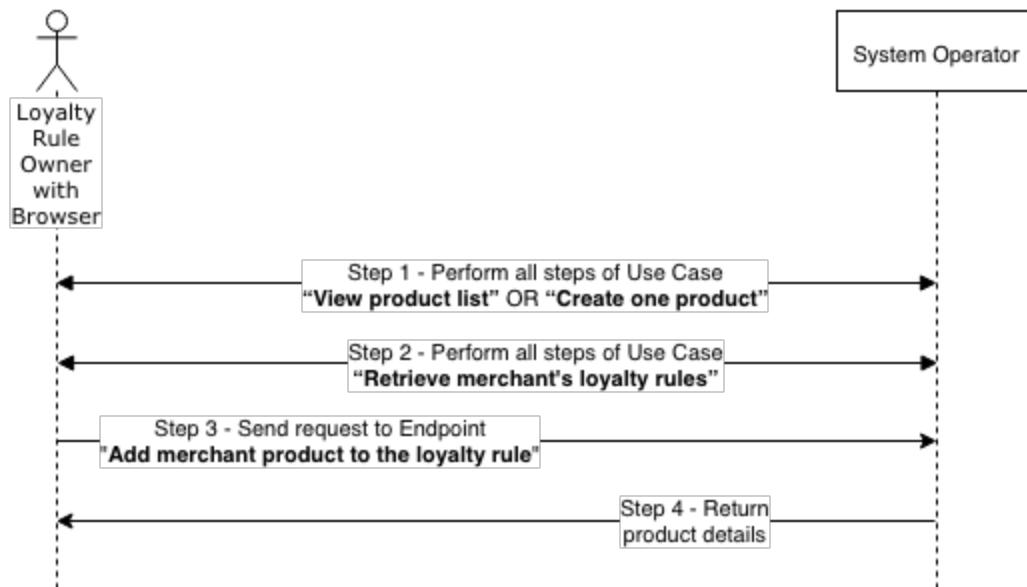
Add merchant product to the loyalty rule scheme

Use case: Add merchant product to the loyalty rule

Basic FLow



Optional Web UI Flow



Add point of sale to the loyalty rule description

Use Case Name

Add point of sale to the loyalty rule

Brief Description

A User or External Entity will go through all steps of “Get points of sale” OR “Create point of sale” Use Case (to obtain PoS ID), and then through “Retrieve merchant's loyalty rules” Use Case (to obtain rule ID). A user, with role permission "LOYALTY_RULE_OWNER", will then send a request to Endpoint “Add point of sale to the loyalty rule”.

Note:

“Get points of sale” OR “Create point of sale” require role permission "POINT_OF_SALE_OWNER".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale” OR “Create point of sale”.
2. Perform all steps of Use Case “Retrieve merchant's loyalty rules”.
3. External Entity sends a request to Endpoint “Add point of sale to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale” OR “Create point of sale”.
2. Perform all steps of Use Case “Retrieve merchant's loyalty rules”.
3. A user sends a request to Endpoint “Add point of sale to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

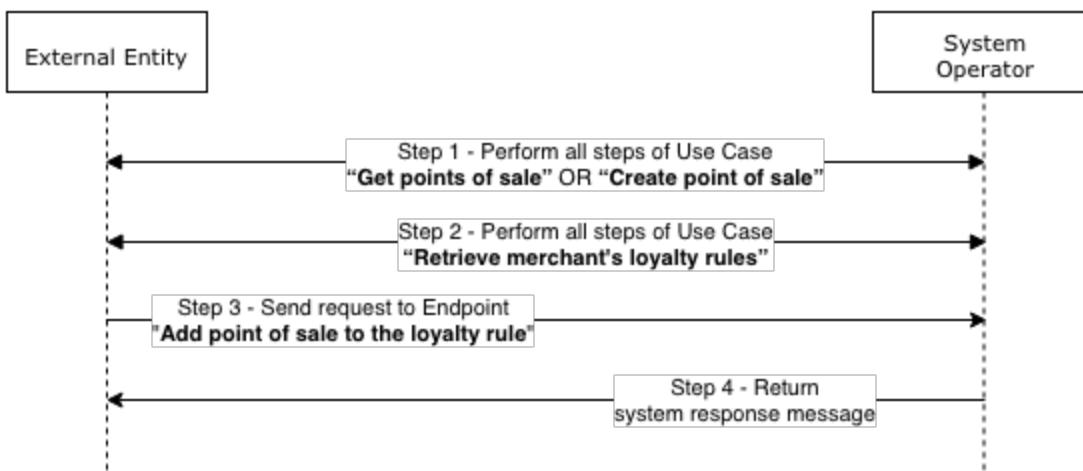
Rule ID and PoS ID are available.

Result example

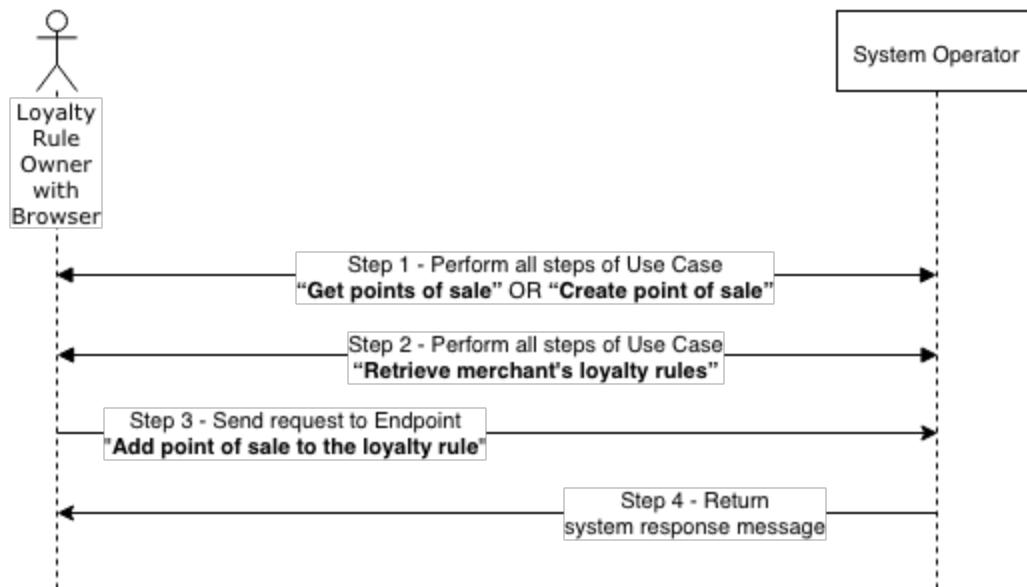
```
{  
  "status": "ok",  
  "message": "string"  
}  
Add point of sale to the loyalty rule scheme
```

Use case: Add point of sale to the loyalty rule

Basic FFlow



Optional Web UI Flow



Create a new loyalty rule description

Use Case Name

Add point of sale to the loyalty rule

Brief Description

A User or External Entity will go through all steps of “Get points of sale” OR “Create point of sale” Use Case (to obtain PoS ID), and then through “Retrieve merchant's loyalty rules” Use Case (to obtain rule ID). A user, with role permission "LOYALTY_RULE_OWNER", will then send a request to Endpoint “Add point of sale to the loyalty rule”.

Note:

“Get points of sale” OR “Create point of sale” require role permission "POINT_OF_SALE_OWNER".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get points of sale” OR “Create point of sale”.
2. Perform all steps of Use Case “Retrieve merchant's loyalty rules”.
3. External Entity sends a request to Endpoint “Add point of sale to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get points of sale” OR “Create point of sale”.
2. Perform all steps of Use Case “Retrieve merchant's loyalty rules”.
3. A user sends a request to Endpoint “Add point of sale to the loyalty rule”.

Endpoint URL: PUT /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Rule ID and PoS ID are available.

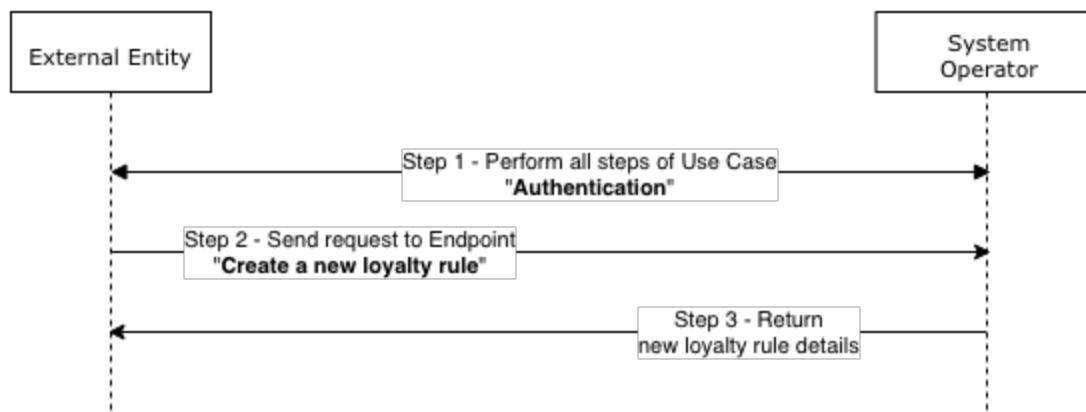
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

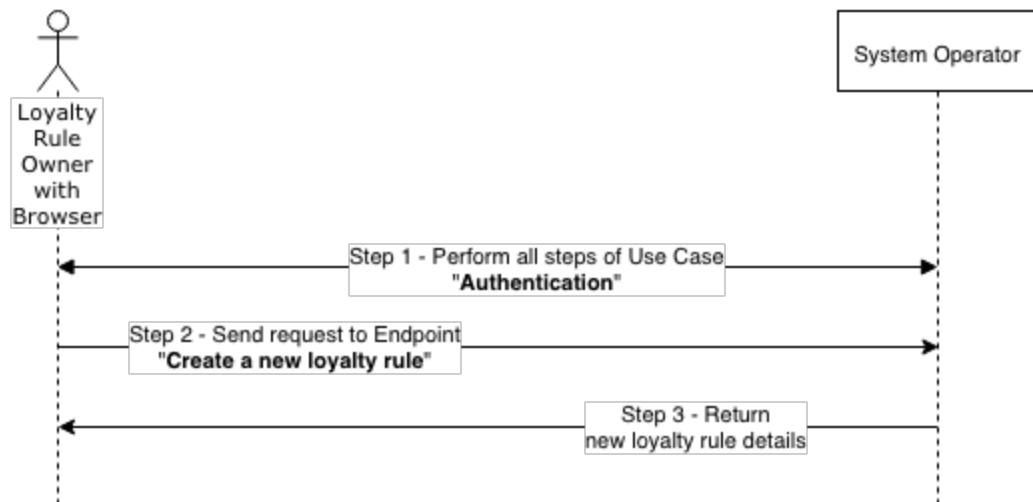
Create a new loyalty rule scheme

Use case: Create a new loyalty rule

Basic FFlow



Optional Web UI Flow



Find loyalty rules eligible for current user description

Use Case Name

Find loyalty rules eligible for current user

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_VIEWER" will go through all steps of "View points of sale used by loyalty rule" Use Case (to retrieve input parameters), and then send a request to Endpoint "Find loyalty rules eligible for current user".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_VIEWER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View points of sale used by loyalty rule".
2. External Entity sends a request to Endpoint "Find loyalty rules eligible for current user".

Endpoint URL: POST /loyalty-rules/view-eligible-rules

```
Parameters: {
  "merchantId": "string",
  "posId": "string"
}
```

1. System Operator returns list of rules eligible for current user to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View points of sale used by loyalty rule".
2. A user sends a request to Endpoint "Find loyalty rules eligible for current user".

Endpoint URL: POST /loyalty-rules/view-eligible-rules

```
Parameters: {
  "merchantId": "string",
  "posId": "string"
}
```

1. System Operator returns list of rules eligible for current user to User (See Result example below).

Post Conditions

Input parameters (merchant ID and PoS ID) are available.

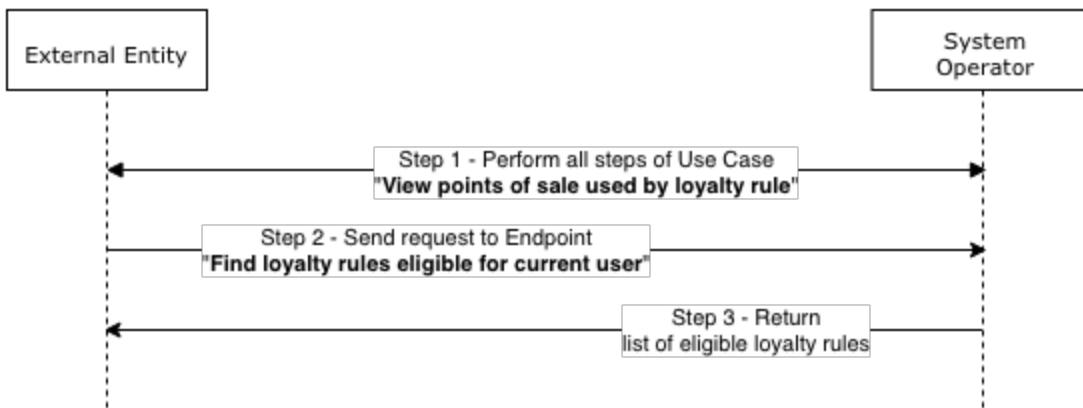
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "type": "BONUS",  
      "name": "string",  
      "description": "string",  
      "active": false,  
      "startsAt": "2018-09-13T10:50:37.386Z",  
      "endsAt": "2018-09-13T10:50:37.386Z",  
      "valueType": "NOT_SPECIFIED",  
      "value": 0,  
      "activationPolicy": "ALL_POINTS_OF_SALE"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

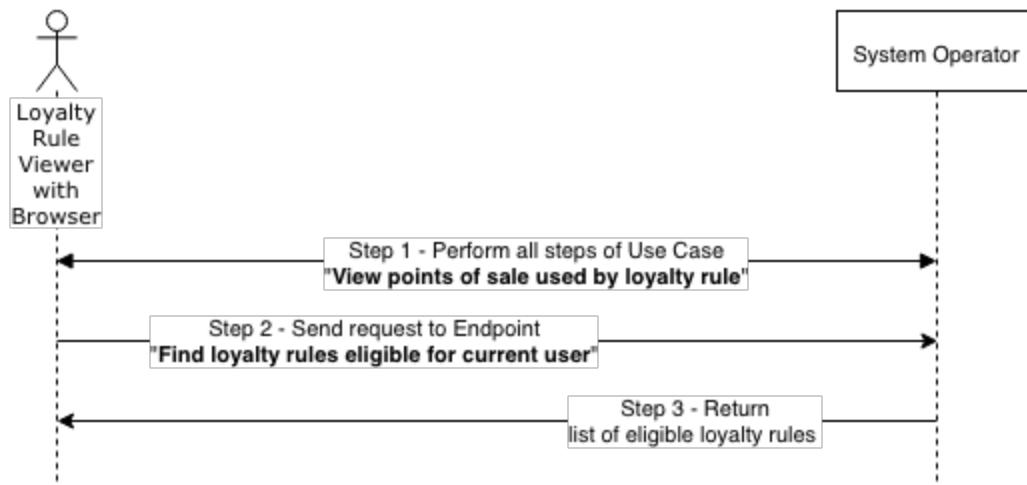
Find loyalty rules eligible for current user scheme

Use case: Find loyalty rules eligible for current user

Basic FFlow



Optional Web UI Flow



Remove issuer from loyalty rule description

Use Case Name

Remove issuer from loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "View issuers used by loyalty rule" Use Case (to obtain issuer ID and rule ID parameters), and then send a request to Endpoint "Remove issuer from loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View issuers used by loyalty rule".
2. External Entity sends a request to Endpoint "Remove issuer from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/issuers/{issuerId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View issuers used by loyalty rule".
2. A user sends a request to Endpoint "Remove issuer from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/issuers/{issuerId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

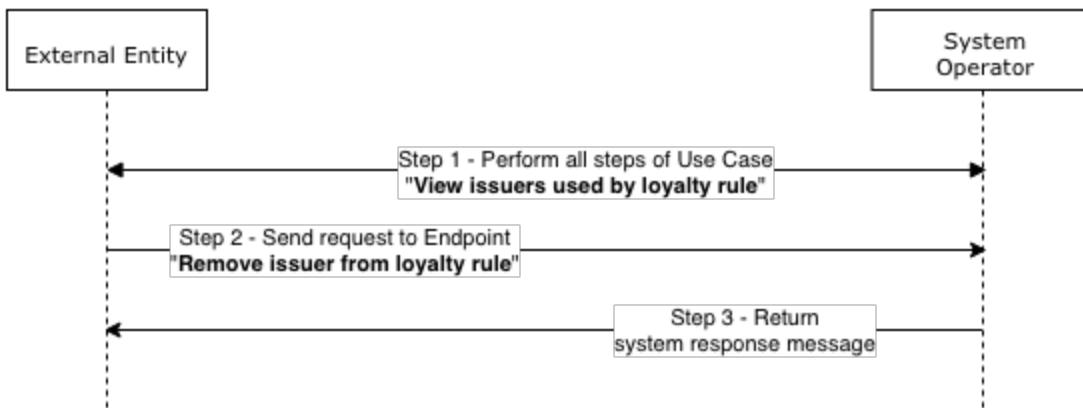
Loyalty rule is available.

Result example

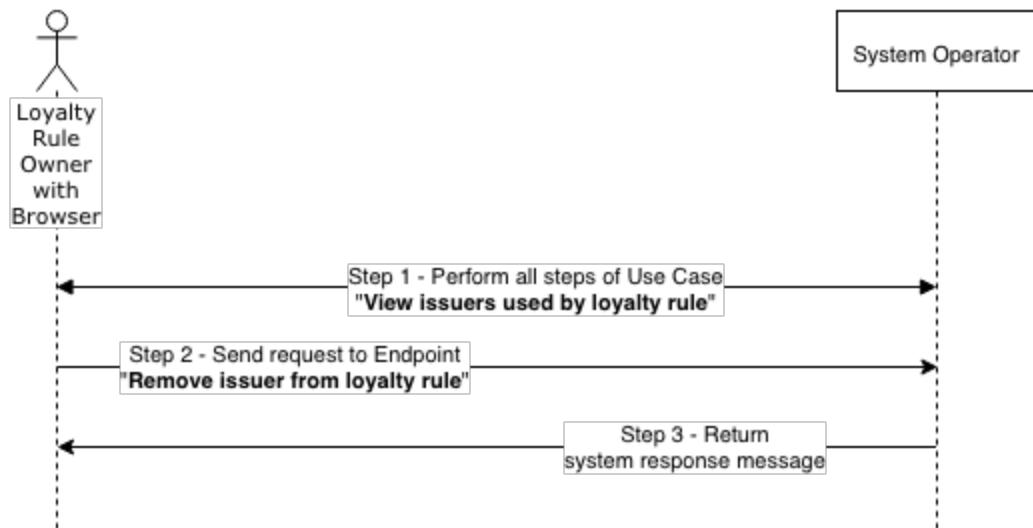
```
{  
  "status": "ok",  
  "message": "string"  
}  
Remove issuer from loyalty rule scheme
```

Use case: Remove issuer from loyalty rule

Basic FFlow



Optional Web UI Flow



Remove loyalty group from loyalty rule description

Use Case Name

Remove issuer from loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "View issuers used by loyalty rule" Use Case (to obtain issuer ID and rule ID parameters), and then send a request to Endpoint "Remove issuer from loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View issuers used by loyalty rule".
2. External Entity sends a request to Endpoint "Remove issuer from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/issuers/{issuerId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View issuers used by loyalty rule".
2. A user sends a request to Endpoint "Remove issuer from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/issuers/{issuerId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Loyalty rule is available.

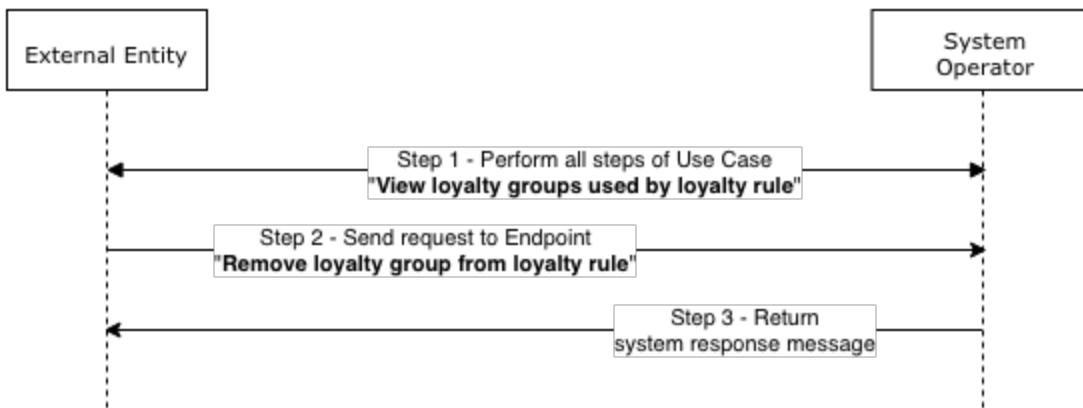
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

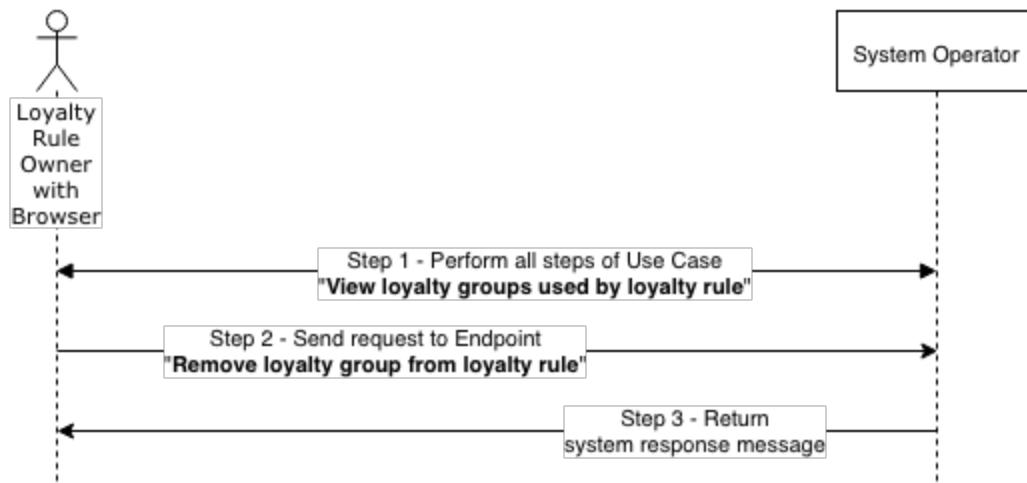
Remove loyalty group from loyalty rule scheme

Use case: Remove loyalty group from loyalty rule

Basic FFlow



Optional Web UI Flow



Remove merchant product category from loyalty rule description

Use Case Name

Remove merchant product category from loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "View merchant products categories used by loyalty rule" Use Case (to obtain rule and product category IDs), and then send a request to Endpoint "Remove merchant product category from loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View merchant products categories used by loyalty rule".
2. External Entity sends a request to Endpoint "Remove merchant product category from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View merchant products categories used by loyalty rule".
2. A user sends a request to Endpoint "Remove merchant product category from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

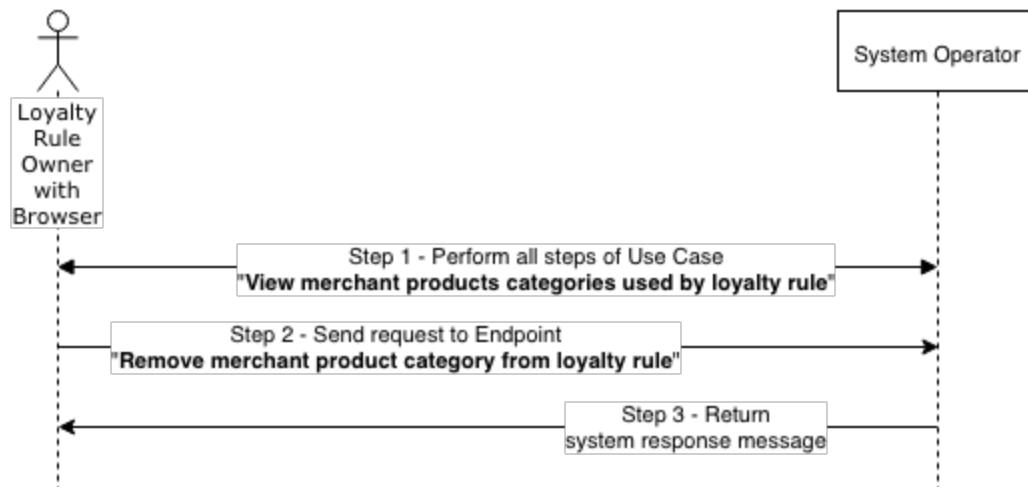
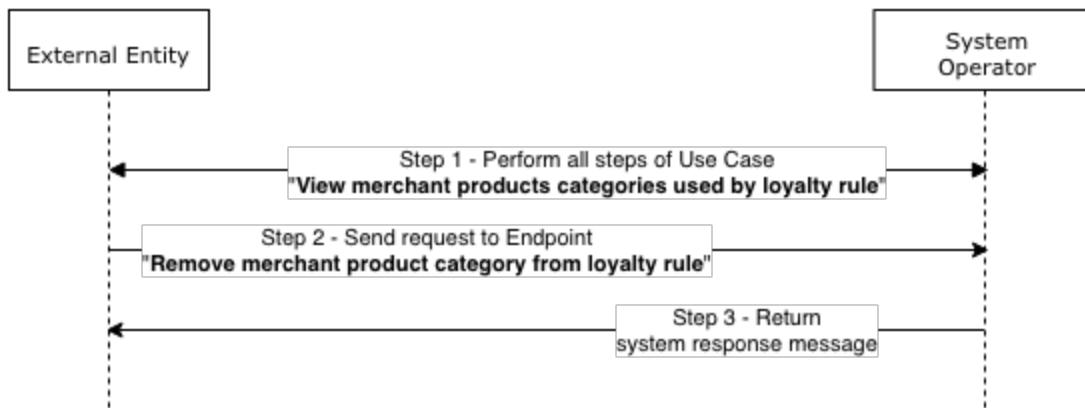
Loyalty rule is available.

Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

Remove merchant product category from loyalty rule scheme

Use case: Remove merchant product category from loyalty rule



Remove merchant product from loyalty rule description

Use Case Name

Remove merchant product category from loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "View merchant products categories used by loyalty rule" Use Case (to obtain rule and product category IDs), and then send a request to Endpoint "Remove merchant product category from loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View merchant products categories used by loyalty rule".
2. External Entity sends a request to Endpoint "Remove merchant product category from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View merchant products categories used by loyalty rule".
2. A user sends a request to Endpoint "Remove merchant product category from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/merchant-product-categories/{categoryId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

Loyalty rule is available.

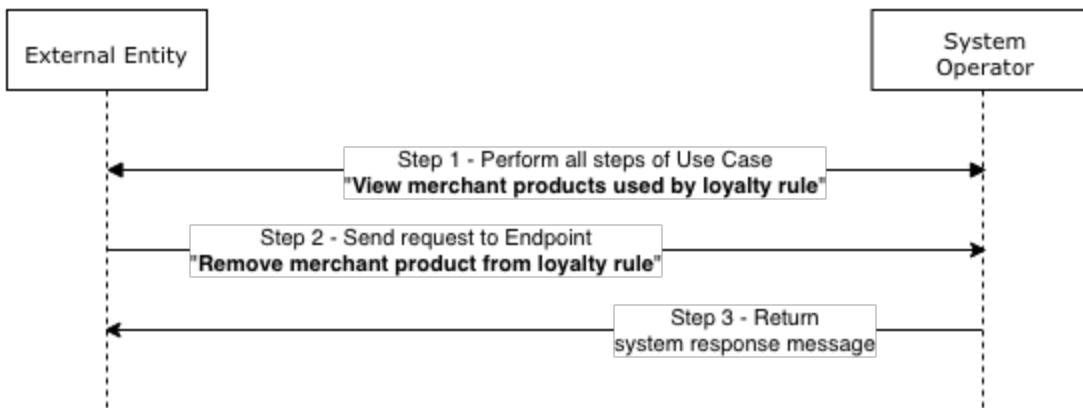
Result example

```
{  
  "status": "ok",  
  "message": "string"  
}
```

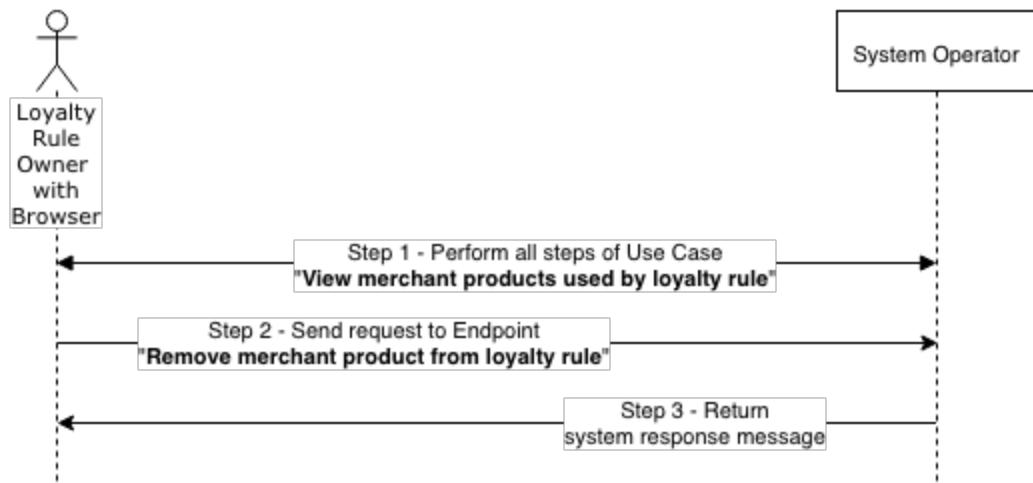
Remove merchant product from loyalty rule scheme

Use case: Remove merchant product from loyalty rule

Basic FFlow



Optional Web UI Flow



Remove point of sale from loyalty rule description

Use Case Name

Remove point of sale from loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "View points of sale used by loyalty rule" Use Case (to retrieve rule and PoS IDs), and then send a request to Endpoint "Remove point of sale from loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View points of sale used by loyalty rule".
2. External Entity sends a request to Endpoint "Remove point of sale from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View points of sale used by loyalty rule".
2. A user sends a request to Endpoint "Remove point of sale from loyalty rule".

Endpoint URL: DELETE /loyalty-rules/{ruleId}/points-of-sale/{posId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns system response message to User (See Result example below).

Post Conditions

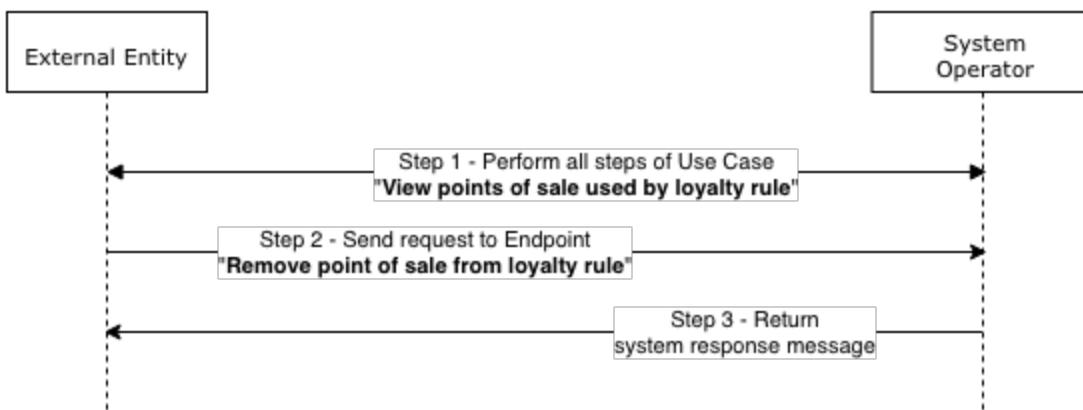
Loyalty rule is available.

Result example

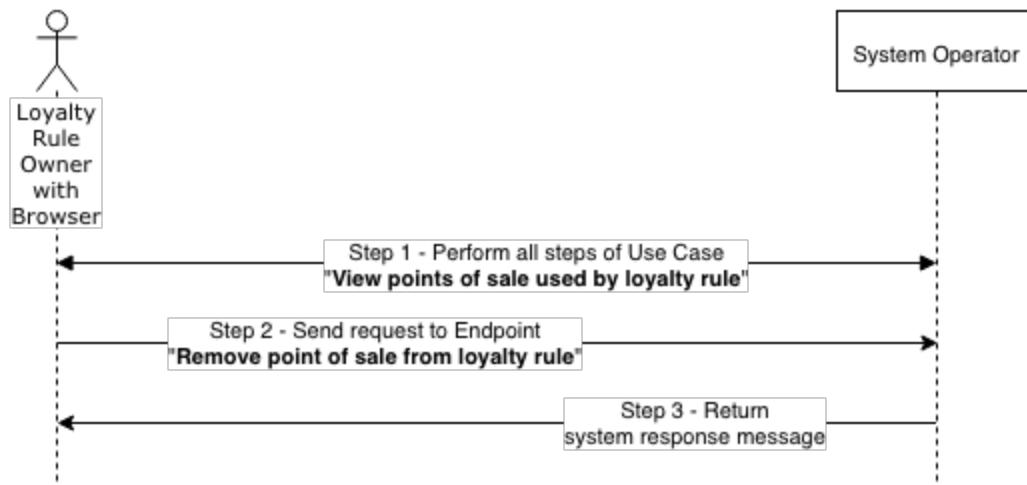
```
{  
  "status": "ok",  
  "message": "string"  
}  
Remove point of sale from loyalty rule scheme
```

Use case: Remove point of sale from loyalty rule

Basic FLow



Optional Web UI Flow



Retrieve media files attached to the loyalty rule description

Use Case Name

Retrieve media files attached to the loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_VIEWER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to obtain rule ID), and then send a request to Endpoint "Retrieve media files attached to the loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_VIEWER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "Retrieve media files attached to the loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/media-files

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of media files attached to a rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "Retrieve media files attached to the loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/media-files

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of media files attached to a rule to User (See Result example below).

Post Conditions

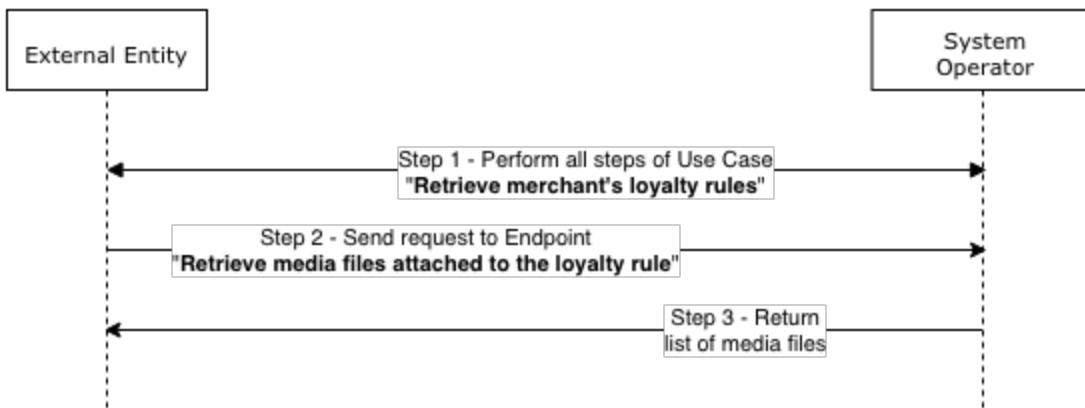
Loyalty rule is available.

Result example

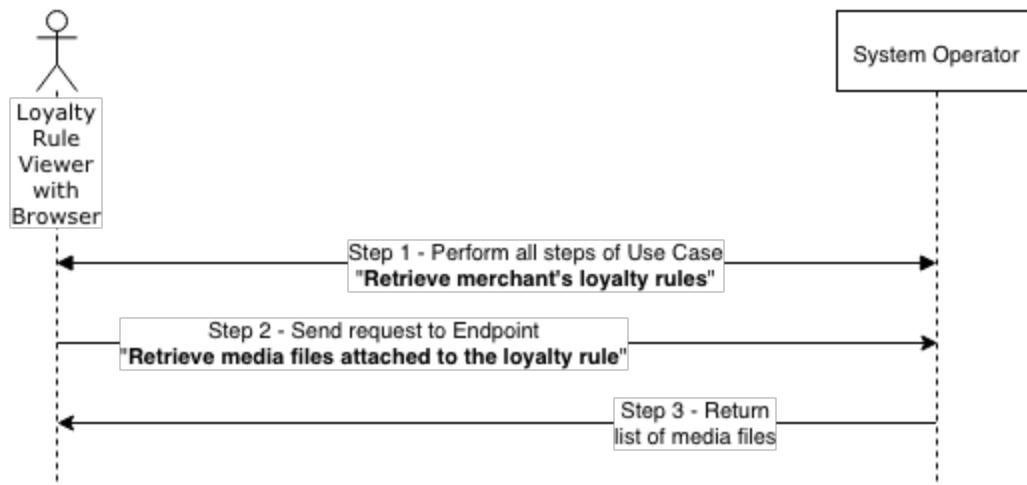
```
{  
  "records": [  
    {  
      "id": "string",  
      "ownerId": "string",  
      "mediaType": "string",  
      "name": "string",  
      "url": "string",  
      "md5": "string",  
      "sha1": "string",  
      "size": 0,  
      "used": false,  
      "createdAt": "2018-09-13T10:50:37.714Z",  
      "expiresAt": "2018-09-13T10:50:37.714Z",  
      "tag": "string"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}  
Retrieve media files attached to the loyalty rule scheme
```

Use case: Retrieve media files attached to the loyalty rule

Basic FLow



Optional Web UI Flow



Retrieve merchant's loyalty rules description

Use Case Name

Retrieve merchant's loyalty rules

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Retrieve merchant's loyalty rules".

Note: You can retrieve a filtered and sorted list of loyalty rules using "View merchant's loyalty rules" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Retrieve merchant's loyalty rules".

Endpoint URL: GET /loyalty-rules

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty rules for logged in merchant to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Retrieve merchant's loyalty rules".

Endpoint URL: GET /loyalty-rules

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty rules for logged in merchant to User (See Result example below).

Post Conditions

One or more loyalty rules are available.

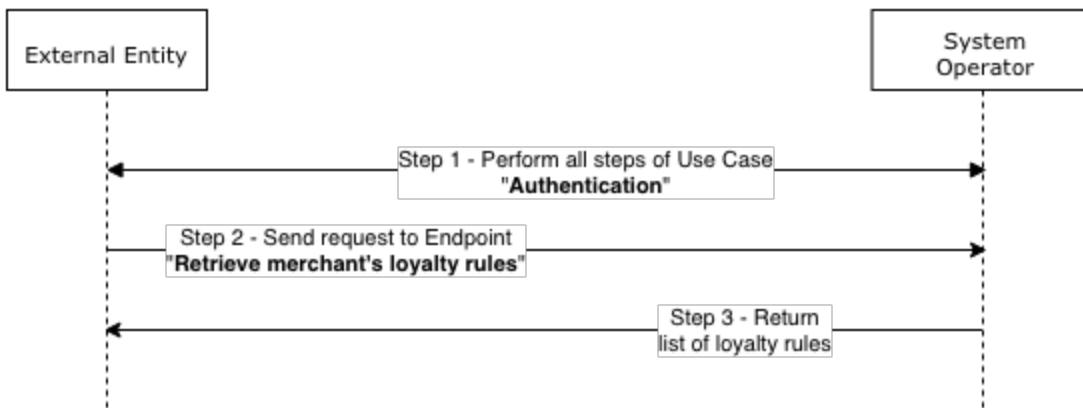
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "type": "BONUS",  
      "name": "string",  
      "description": "string",  
      "active": false,  
      "startsAt": "2018-09-13T10:50:37.363Z",  
      "endsAt": "2018-09-13T10:50:37.363Z",  
      "valueType": "NOT_SPECIFIED",  
      "value": 0,  
      "activationPolicy": "ALL_POINTS_OF_SALE"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

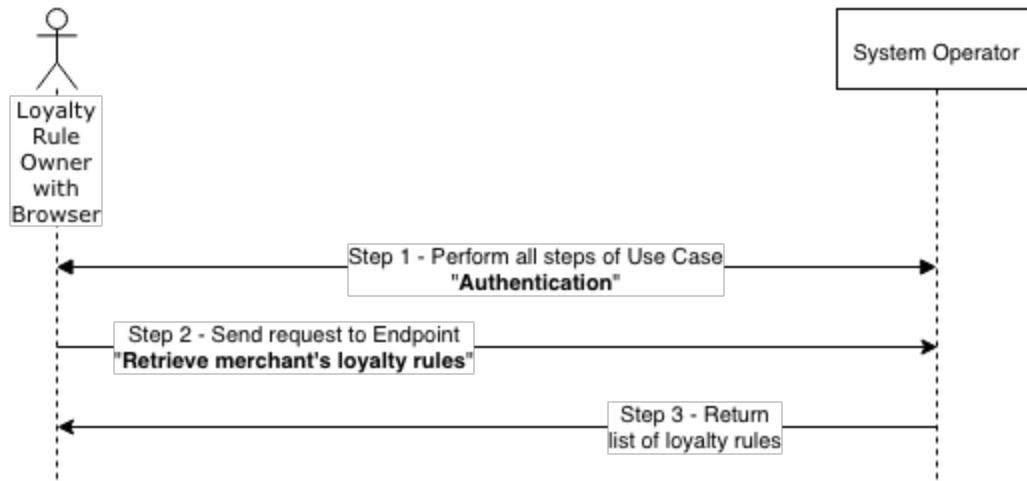
Retrieve merchant's loyalty rules scheme

Use case: Retrieve merchant's loyalty rules

Basic FFlow



Optional Web UI Flow



Update an existing loyalty rule description

Use Case Name

Retrieve merchant's loyalty rules

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Retrieve merchant's loyalty rules".

Note: You can retrieve a filtered and sorted list of loyalty rules using "View merchant's loyalty rules" Use Case.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Retrieve merchant's loyalty rules".

Endpoint URL: GET /loyalty-rules

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty rules for logged in merchant to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Retrieve merchant's loyalty rules".

Endpoint URL: GET /loyalty-rules

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty rules for logged in merchant to User (See Result example below).

Post Conditions

One or more loyalty rules are available.

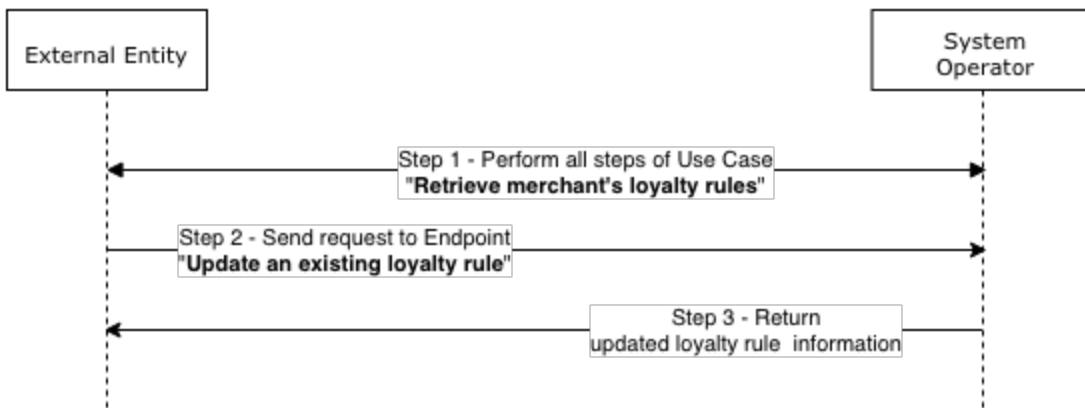
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "type": "BONUS",  
      "name": "string",  
      "description": "string",  
      "active": false,  
      "startsAt": "2018-09-13T10:50:37.363Z",  
      "endsAt": "2018-09-13T10:50:37.363Z",  
      "valueType": "NOT_SPECIFIED",  
      "value": 0,  
      "activationPolicy": "ALL_POINTS_OF_SALE"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

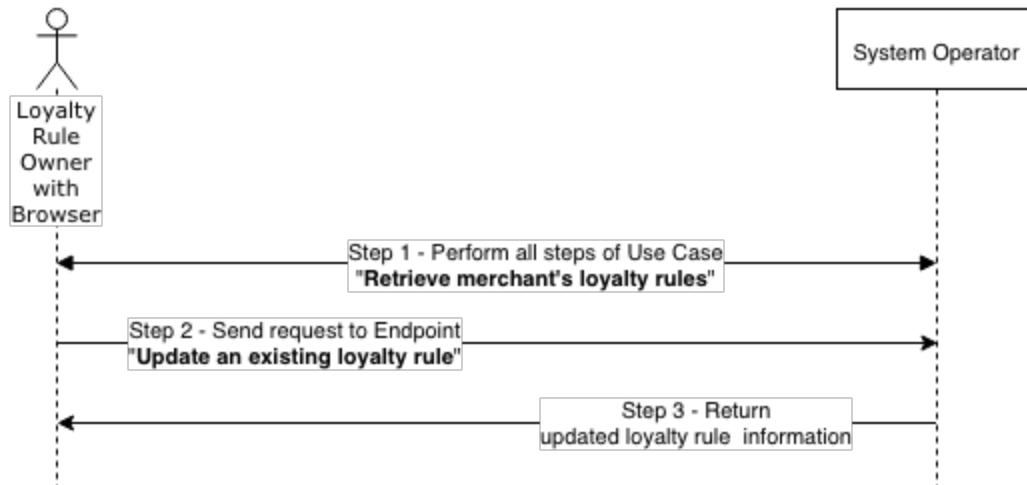
Update an existing loyalty rule scheme

Use case: Update an existing loyalty rule

Basic FFlow



Optional Web UI Flow



View issuers used by loyalty rule description

Use Case Name

View issuers used by loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to obtain ruleId parameter), and then send a request to Endpoint "View issuers used by loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "View issuers used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/issuers

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of issuers for merchant's rule ID to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "View issuers used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/issuers

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of issuers for merchant's rule ID to User (See Result example below).

Post Conditions

Rule ID is available.

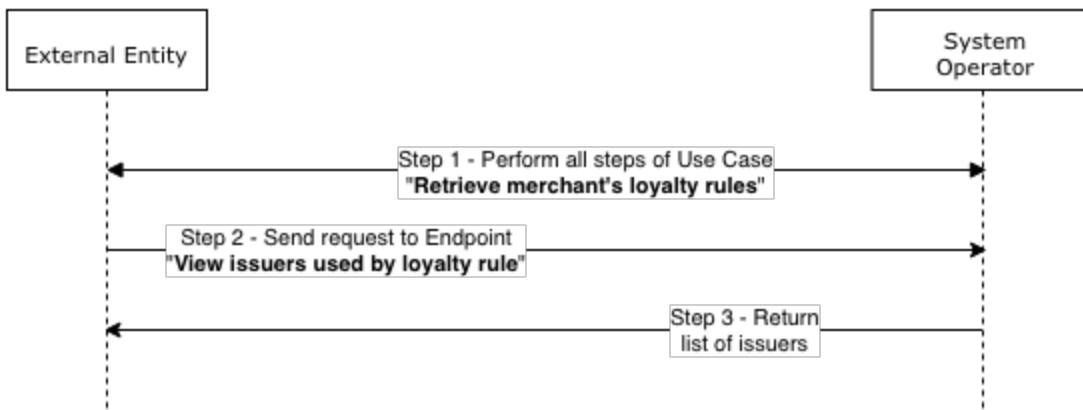
Result example

```
{  
  "issuers": [  
    {  
      "issuer": {  
        "id": "string",  
        "sn": "string",  
        "currency": "string"  
      },  
      "restrictions": {  
        "min": 0,  
        "max": 0  
      }  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

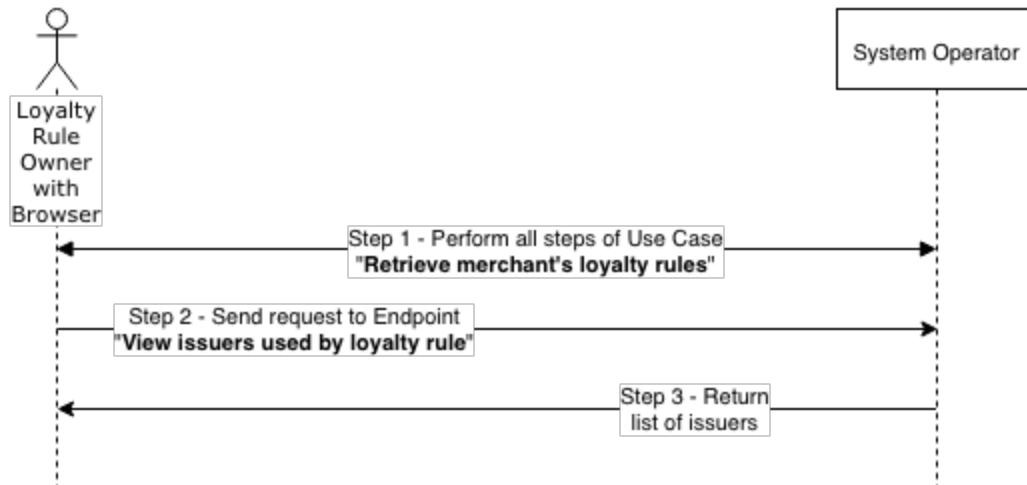
View issuers used by loyalty rule scheme

Use case: View issuers used by loyalty rule

Basic FFlow



Optional Web UI Flow



View loyalty groups used by loyalty rule description

Use Case Name

View loyalty groups used by loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to retrieve rule ID), and then send a request to Endpoint "View loyalty groups used by loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "View loyalty groups used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups used by loyalty rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "View loyalty groups used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups used by loyalty rule to User (See Result example below).

Post Conditions

Loyalty rule is available.

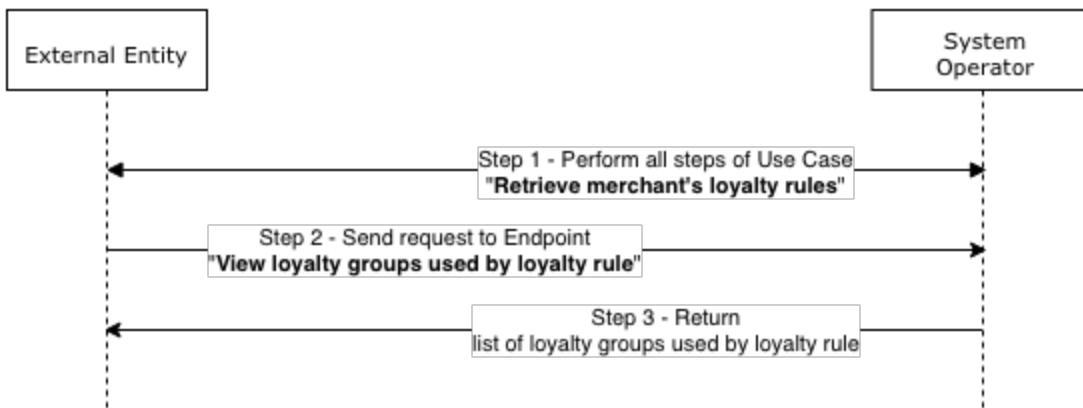
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-09-13T10:50:37.686Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

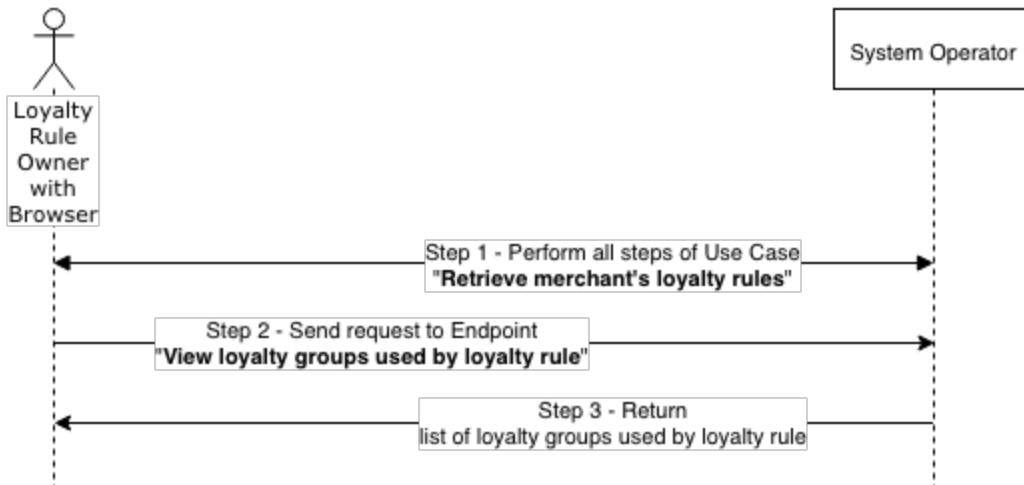
View loyalty groups used by loyalty rule scheme

Use case: View loyalty groups used by loyalty rule

Basic FFlow



Optional Web UI Flow



View merchant products categories used by loyalty rule description

Use Case Name

View merchant products categories used by loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_VIEWER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to retrieve rule ID), and then send a request to Endpoint "View merchant products categories used by loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_VIEWER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "View merchant products categories used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product categories for a rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "View merchant products categories used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product categories for a rule to User (See Result example below).

Post Conditions

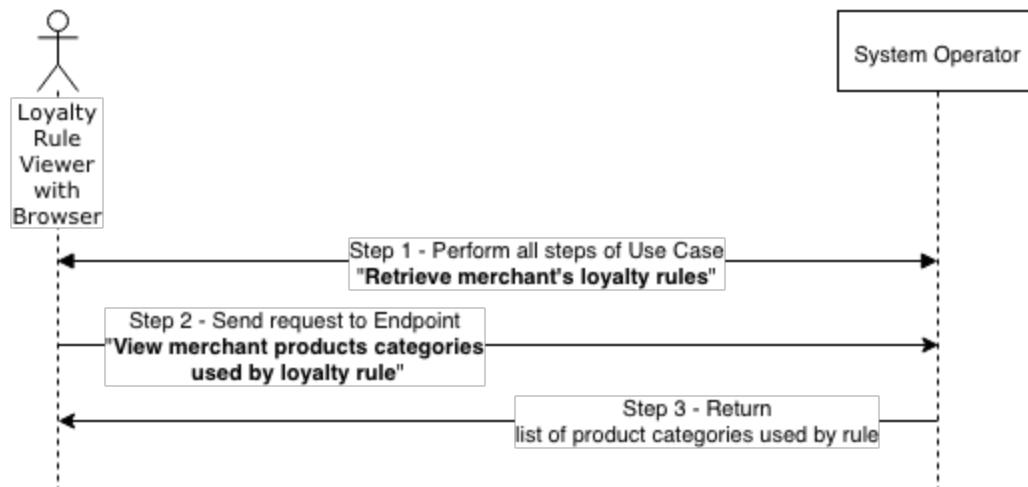
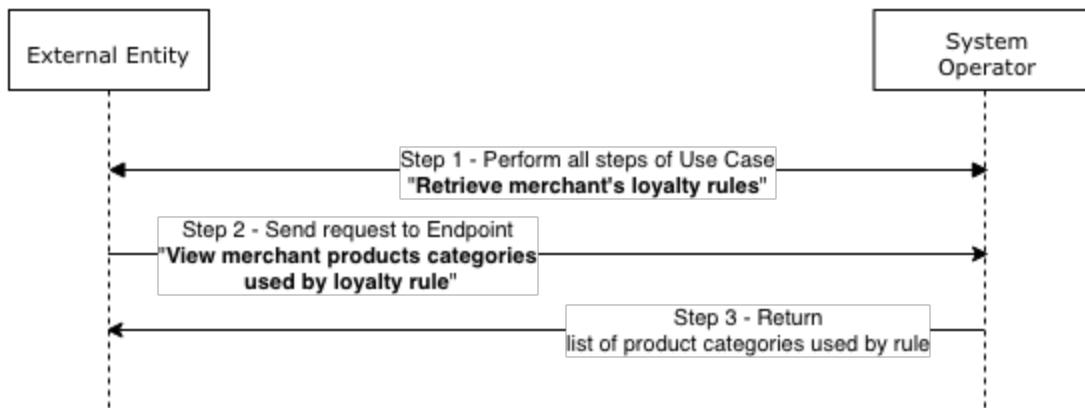
One or more product categories are available.

Result example

```
{  
  "categories": [  
    {  
      "category": {  
        "id": 0,  
        "externalCode": "string",  
        "name": "string",  
        "description": "string"  
      },  
      "valueType": "NOT_SPECIFIED",  
      "value": 0  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

View merchant products categories used by loyalty rule scheme

Use case: View merchant products categories used by loyalty rule



View merchant products used by loyalty rule description

Use Case Name

View merchant products categories used by loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_VIEWER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to retrieve rule ID), and then send a request to Endpoint "View merchant products categories used by loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_VIEWER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "View merchant products categories used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product categories for a rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "View merchant products categories used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/merchant-product-categories

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of product categories for a rule to User (See Result example below).

Post Conditions

One or more product categories are available.

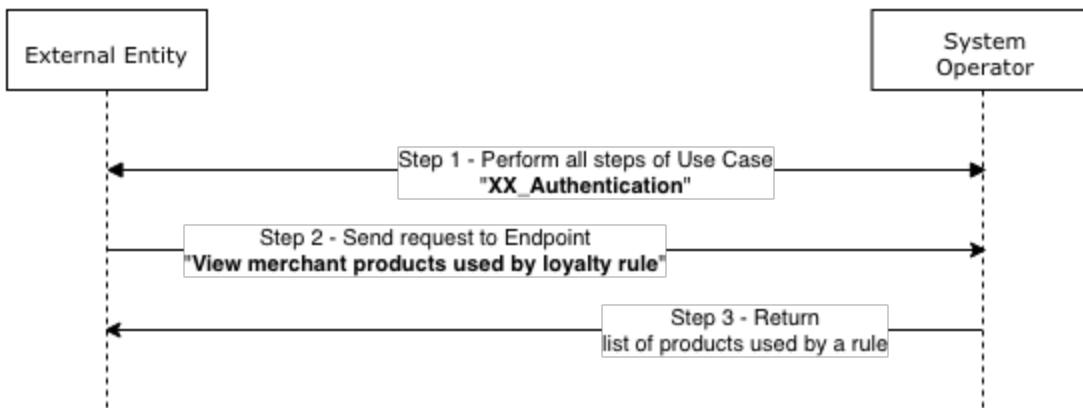
Result example

```
{  
  "categories": [  
    {  
      "category": {  
        "id": 0,  
        "externalCode": "string",  
        "name": "string",  
        "description": "string"  
      },  
      "valueType": "NOT_SPECIFIED",  
      "value": 0  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

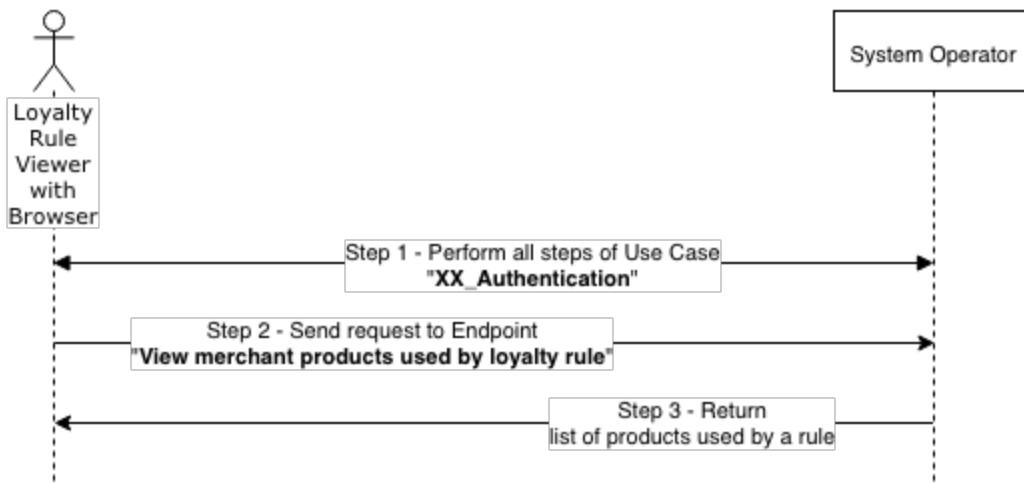
View merchant products used by loyalty rule scheme

Use case: View merchant products used by loyalty rule

Basic FFlow



Optional Web UI Flow



View merchant's loyalty rules description

Use Case Name

View merchant's loyalty rules

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View merchant's loyalty rules". This endpoint allows you to filter a list of loyalty rules, e.g. by whether it is active or not for a merchant.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View merchant's loyalty rules".

Endpoint URL: POST /loyalty-rules/view

Parameters: {

```
"filter": {
  "active": false
},
"sort": {
  "date": "asc",
  "startsAt": "asc",
  "endsAt": "asc"
},
"pageNumber": 0,
"pageSize": 0
}
```

1. System Operator returns filtered list of loyalty rules for logged in merchant ID to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".

2. A user sends a request to Endpoint “View merchant's loyalty rules”.

Endpoint URL: POST /loyalty-rules/view

Parameters: {

```

"filter": {
  "active": false
},
"sort": {
  "date": "asc",
  "startsAt": "asc",
  "endsAt": "asc"
},
"pageNumber": 0,
"pageSize": 0
}

```

1. System Operator returns filtered list of loyalty rules for logged in merchant ID to User (See Result example below).

Post Conditions

One or more loyalty rules are available.

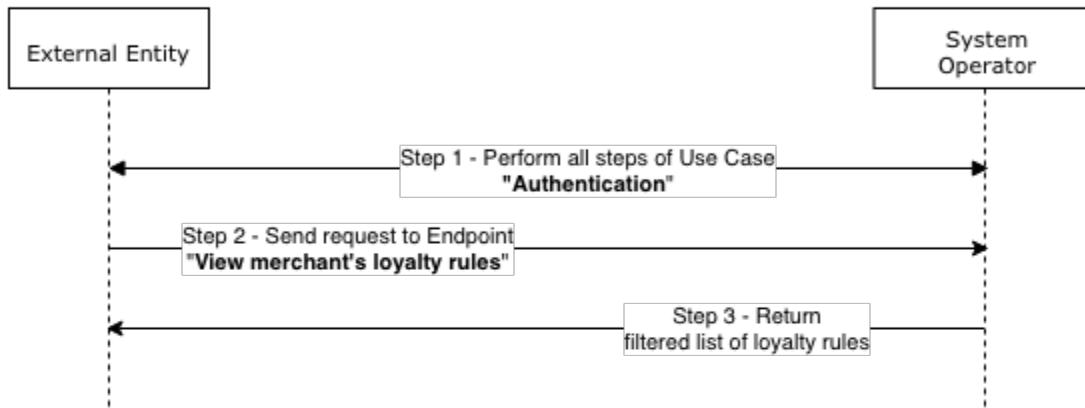
Result example

```
{
  "records": [
    {
      "id": "string",
      "type": "BONUS",
      "name": "string",
      "description": "string",
      "active": false,
      "startsAt": "2018-09-13T10:50:37.371Z",
      "endsAt": "2018-09-13T10:50:37.371Z",
      "valueType": "NOT_SPECIFIED",
      "value": 0,
      "activationPolicy": "ALL_POINTS_OF_SALE"
    }
  ],
  "status": "ok",
  "message": "string"
}
```

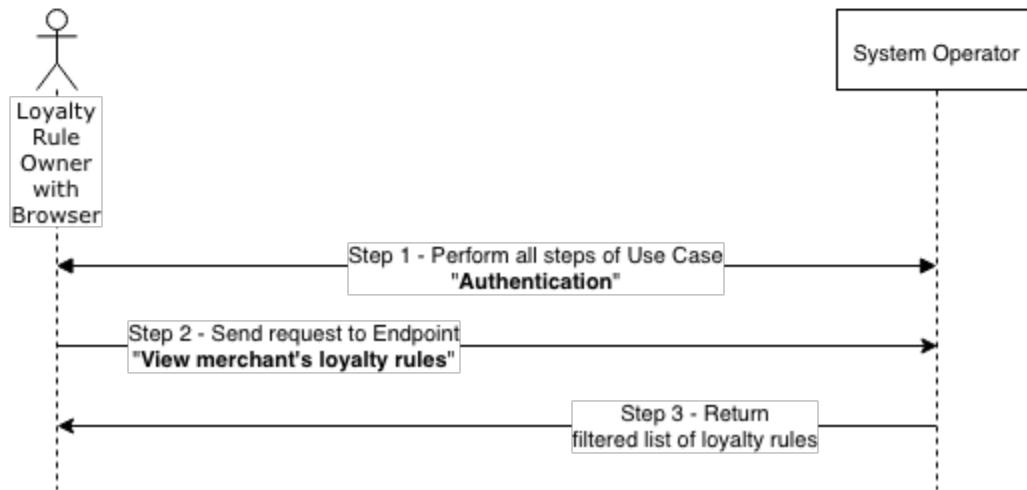
View merchant's loyalty rules scheme

Use case: View merchant's loyalty rules

Basic FLow



Optional Web UI Flow



View points of sale used by loyalty rule description

Use Case Name

View points of sale used by loyalty rule

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_VIEWER" will go through all steps of "Retrieve merchant's loyalty rules" Use Case (to retrieve loyalty rule ID), and then send a request to Endpoint "View points of sale used by loyalty rule".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_VIEWER", e.g. individual, merchant, payroll specialist, payroll manager or exchange manager.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. External Entity sends a request to Endpoint "View points of sale used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/points-of-sale

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of PoS devices used by a loyalty rule to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Retrieve merchant's loyalty rules".
2. A user sends a request to Endpoint "View points of sale used by loyalty rule".

Endpoint URL: GET /loyalty-rules/{ruleId}/points-of-sale

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of PoS devices used by a loyalty rule to User (See Result example below).

Post Conditions

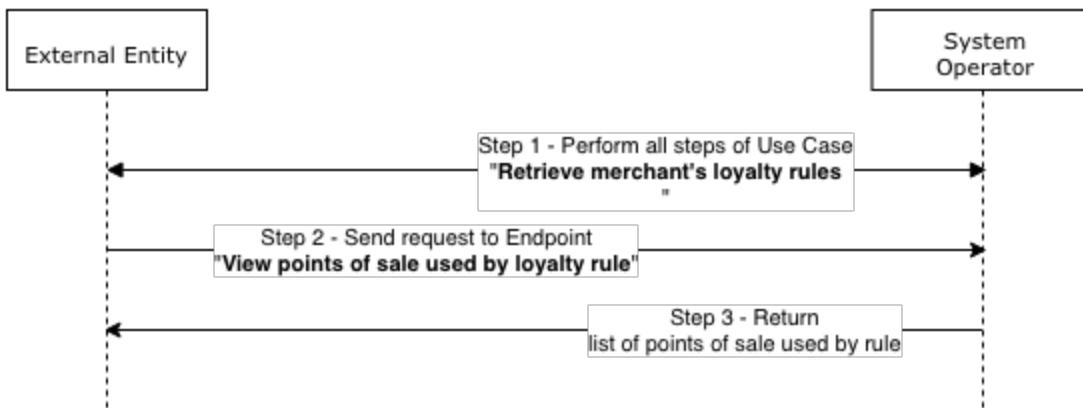
Loyalty rule is available.

Result example

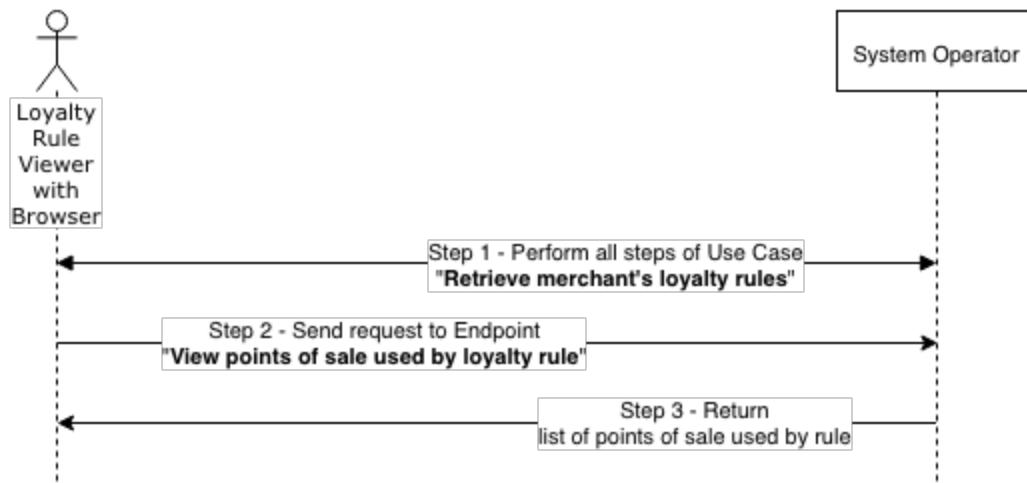
```
{  
  "records": [  
    {  
      "id": "string",  
      "name": "string",  
      "description": "string",  
      "website": "string",  
      "active": false  
    }  
,  
    "  
  "status": "ok",  
  "message": "string"  
}  
View points of sale used by loyalty rule scheme
```

Use case: View points of sale used by loyalty rule

Basic FFlow



Optional Web UI Flow



Loyalty rule activations

Get activations of the loyalty rules occurred during merchant payment description

Use Case Name

Get loyalty groups

Brief Description

A User or External Entity on behalf of a User with role permission "LOYALTY_RULE_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Get loyalty groups".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "LOYALTY_RULE_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Get loyalty groups".

Endpoint URL: GET /loyalty-groups

Parameters: TOKEN - identifies authenticated user

1. System Operator returns list of loyalty groups to User (See Result example below).

Post Conditions

One or more loyalty groups are available.

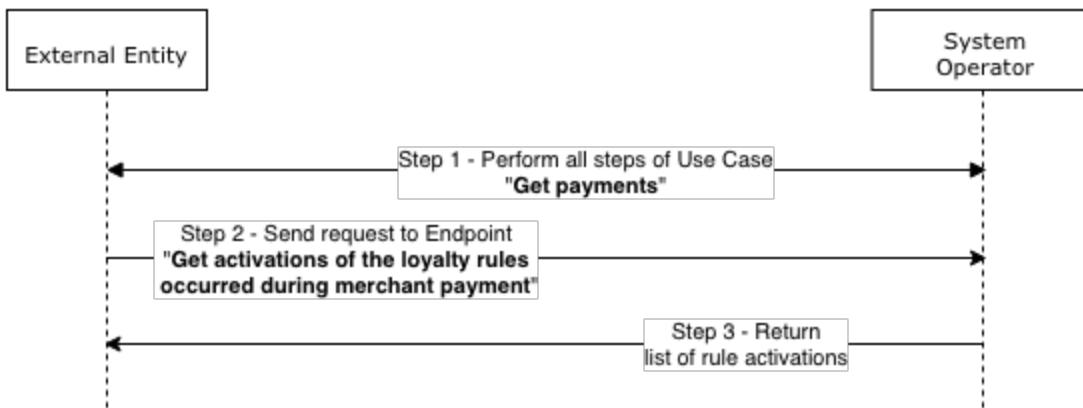
Result example

```
{  
  "records": [  
    {  
      "id": "string",  
      "createdAt": "2018-08-31T14:37:04.117Z",  
      "name": "string",  
      "conditions": {}  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

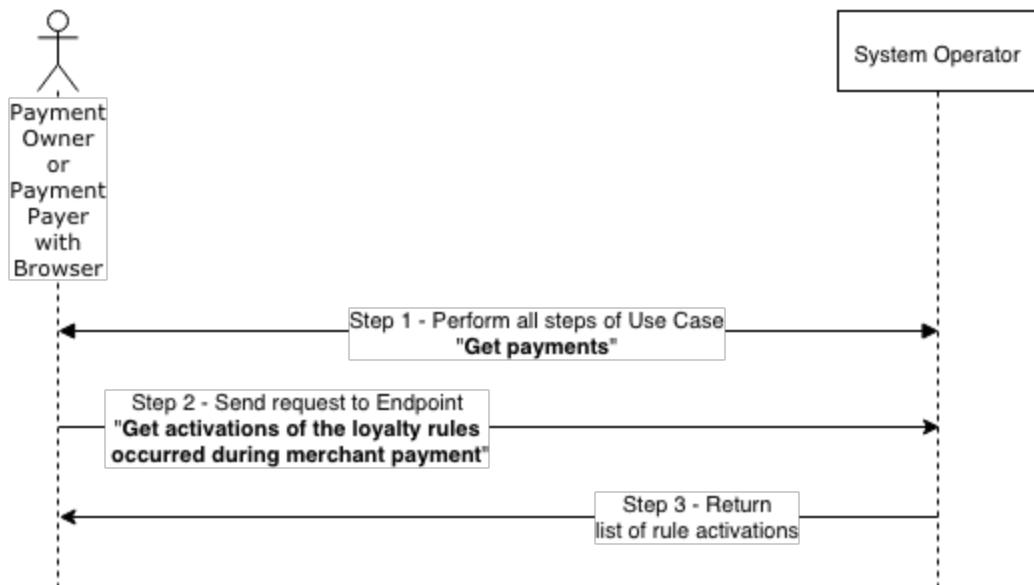
Get activations of the loyalty rules occurred during merchant payment scheme

Use case: Get activations of the loyalty rules occurred during merchant payment

Basic FFlow



Optional Web UI Flow



I18n - management

Create a new I18n record description

Use Case Name

Create a new I18n record

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Create a new I18n record”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
I18N_RECORD_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Create a new I18n record”.

Endpoint URL: POST /i18n-records

Parameters:

```
{
  "key": "err.invalid.entry",
  "values": [
    {
      "locale": "en",
      "value": "Your entry is invalid"
    }
  ]
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Create a new I18n record”.

Endpoint URL: POST /i18n-records

Parameters:

```
{  
    "key": "err.invalid.entry",  
    "values": [  
        {  
            "locale": "en",  
            "value": "Your entry is invalid"  
        }  
    ]  
}
```

3. System Operator returns result information to User (See Result example below).

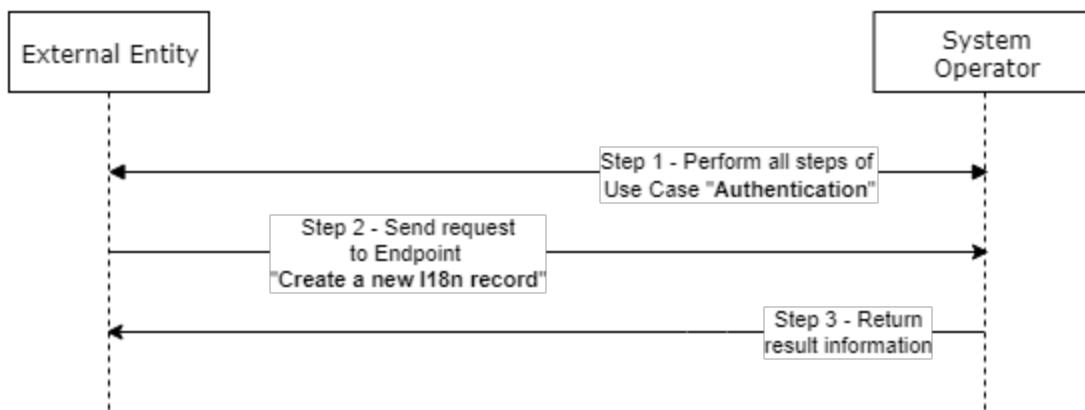
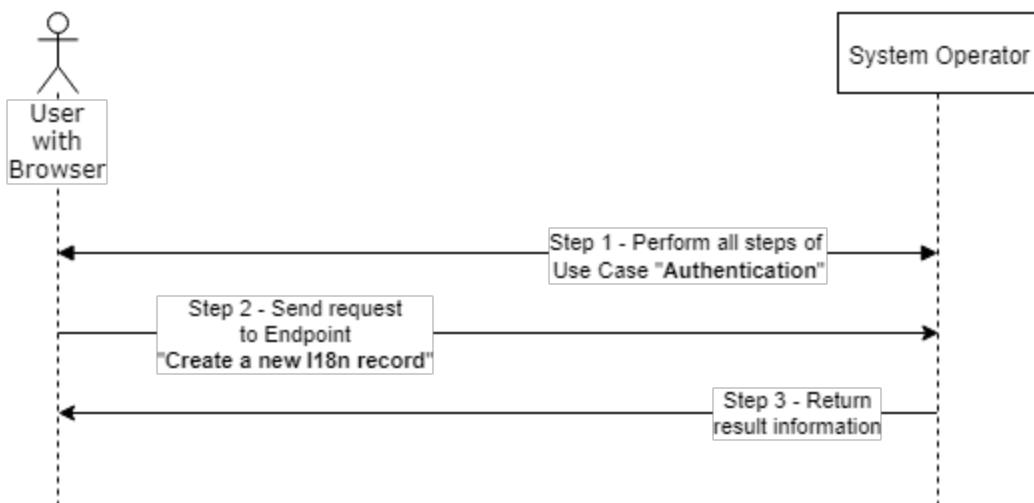
Post Conditions

A new entry is available to include in code.

Result example

```
{  
    "property": {  
        "id": 0,  
        "key": "string",  
        "values": [  
            {  
                "locale": "en",  
                "value": "translated_string_value"  
            }  
        ]  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Create a new I18n record scheme

Use case: Create a new I18n record**Basic FFlow****Optional Web UI Flow**

Delete a I18n record description

Use Case Name

Delete an I18n record

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “View I18n records” Use Case, and then send a request to Endpoint “Delete an I18n record”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View I18n records”.

2. External Entity sends a request to Endpoint “Delete an I18n record”.

Endpoint URL: DELETE /i18n-records/{id}

Parameters: Security TOKEN

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View I18n records”.

2. A user sends a request to Endpoint “Delete an I18n record”.

Endpoint URL: DELETE /i18n-records/{id}

Parameters: Security TOKEN

3. System Operator returns xx result information to User (See Result example below).

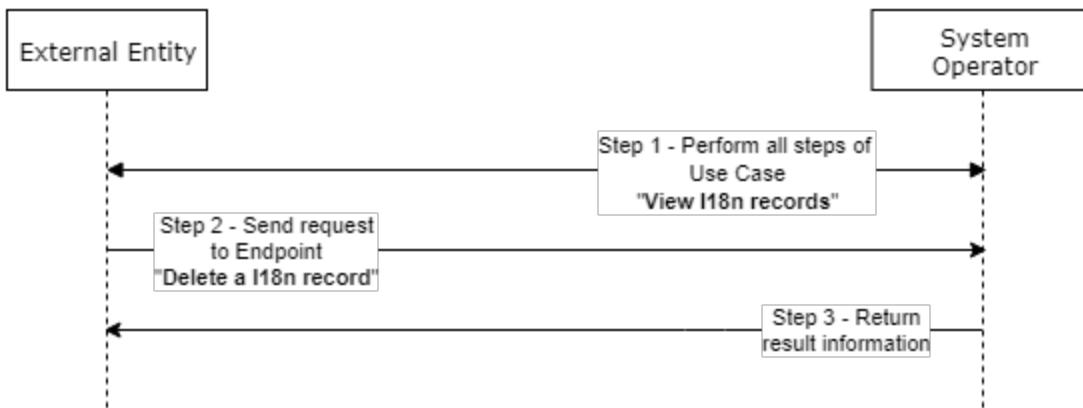
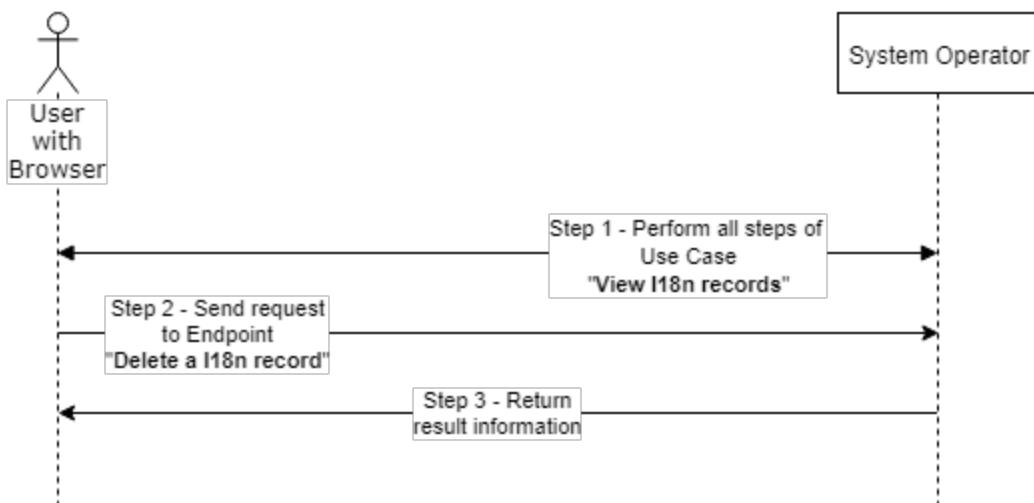
Post Conditions

The item is removed.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Delete a I18n record scheme

Use case: Delete a I18n record**Basic FFlow****Optional Web UI Flow**

Download all properties in excel file description

Use Case Name

Download all properties in excel file

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Download all properties in excel file”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: XX
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Download all properties in excel file”.

Endpoint URL: POST /i18n-records/export

Parameters: Security TOKEN

3. System Operator returns download URL and the result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Download all properties in excel file”.

Endpoint URL: POST /i18n-records/export

Parameters: Security TOKEN

3. System Operator returns download dialog and the result information to User (See Result example below).

Post Conditions

Excel file is available.

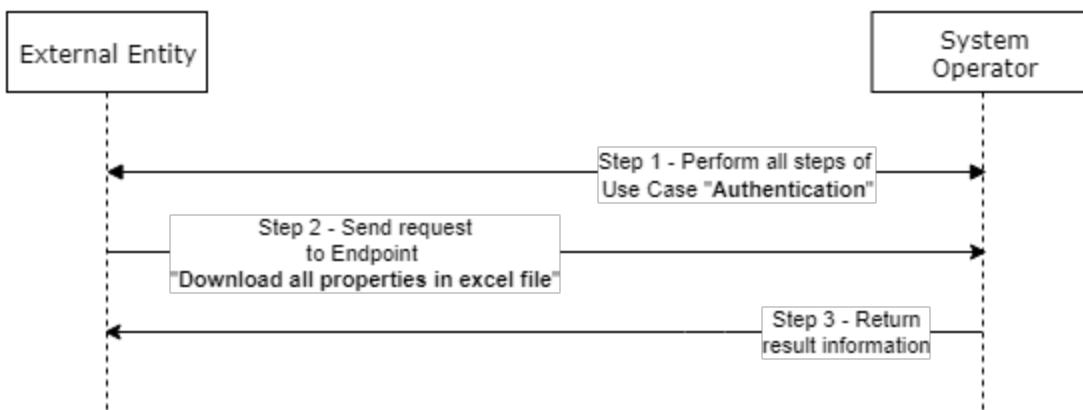
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

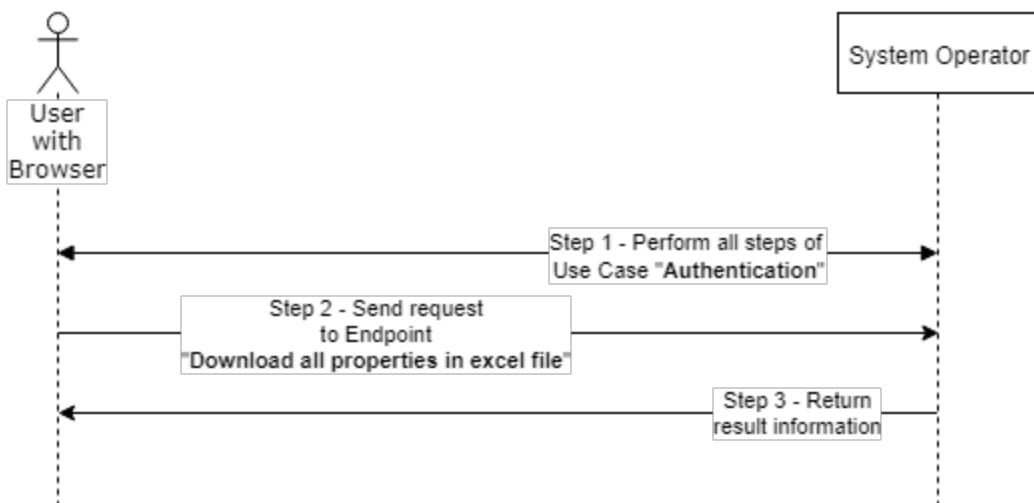
Download all properties in excel file scheme

Use case: Download all properties in excel file

Basic FFlow



Optional Web UI Flow



Update a I18n record description

Use Case Name

Update an I18n record

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “View I18n records” Use Case, and then send a request to Endpoint “Update a I18n record”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Update an I18n record”.
2. External Entity sends a request to Endpoint “Delete an I18n record”.

Endpoint URL: PATCH /i18n-records/{id}

Parameters: Security TOKEN

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Update an I18n record”.
2. A user sends a request to Endpoint “Delete an I18n record”.

Endpoint URL: PATCH /i18n-records/{id}

Parameters: Security TOKEN

3. System Operator returns xx result information to User (See Result example below).

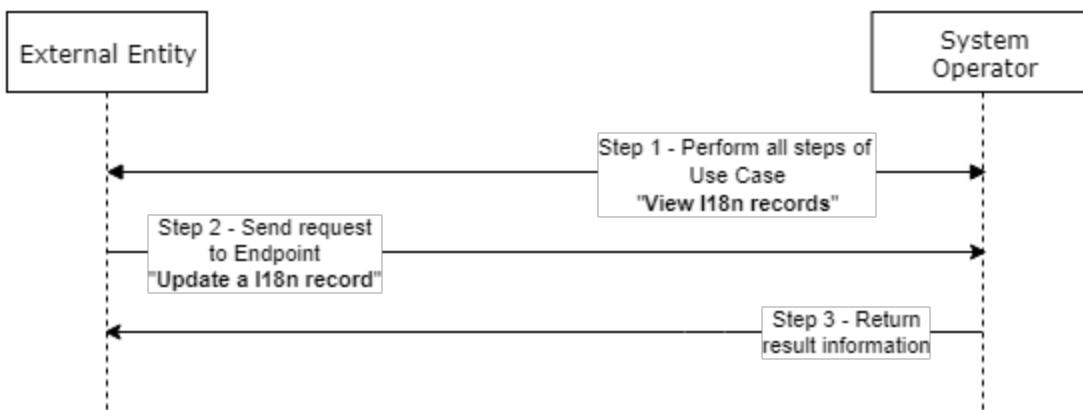
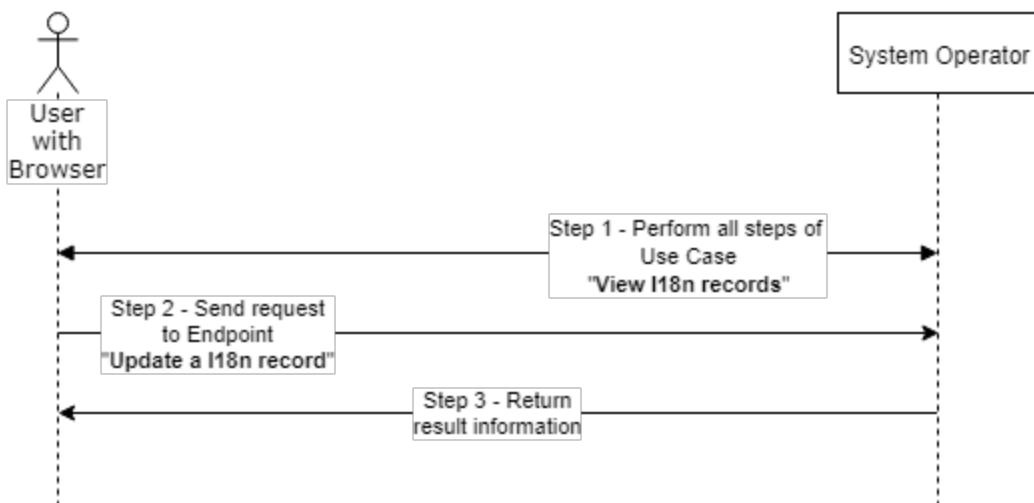
Post Conditions

The item is removed.

Result example

```
{
  "status": "ok",
  "message": "string"
}
```

Update a I18n record scheme

Use case: Update a I18n record**Basic FFlow****Optional Web UI Flow**

Upload all properties from excel file description

Use Case Name

Upload all properties from excel file

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Upload all properties from excel file”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Upload all properties from excel file”.

Endpoint URL: POST /i18n-records/import

Parameters: Path to a file to upload

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Upload all properties from excel file”.

Endpoint URL: POST /i18n-records/import

Parameters: Path to a file to upload

3. System Operator returns result information to User (See Result example below).

Post Conditions

Items from the uploaded file are available.

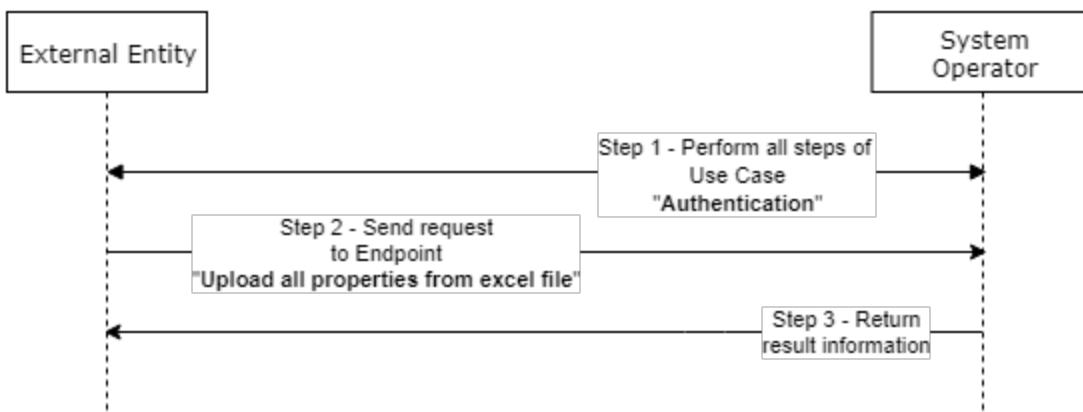
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

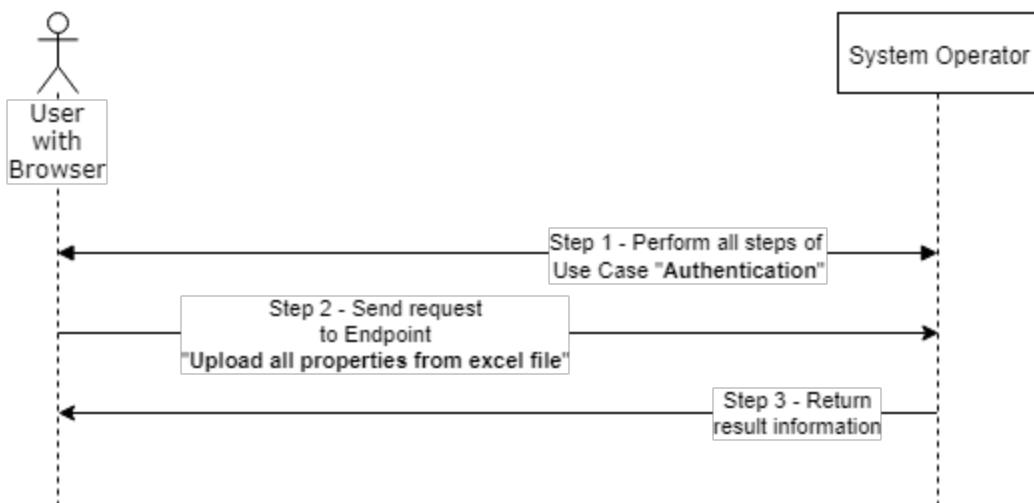
Upload all properties from excel file scheme

Use case: Upload all properties from excel file

Basic FFlow



Optional Web UI Flow



View all I18n records by locale description

Use Case Name

View all I18n records by locale

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “View all locales” Use Case, and then send a request to Endpoint “View all I18n records by locale”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “View all locales”.
2. External Entity sends a request to Endpoint “View all I18n records by locale”.

Endpoint URL: GET /i18n-records/{locale}

Parameters: Security TOKEN

3. System Operator returns xx result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “View all locales”.
2. A user sends a request to Endpoint “View all I18n records by locale”.

Endpoint URL: GET /i18n-records/{locale}

Parameters: Security TOKEN

3. System Operator returns xx result information to User (See Result example below).

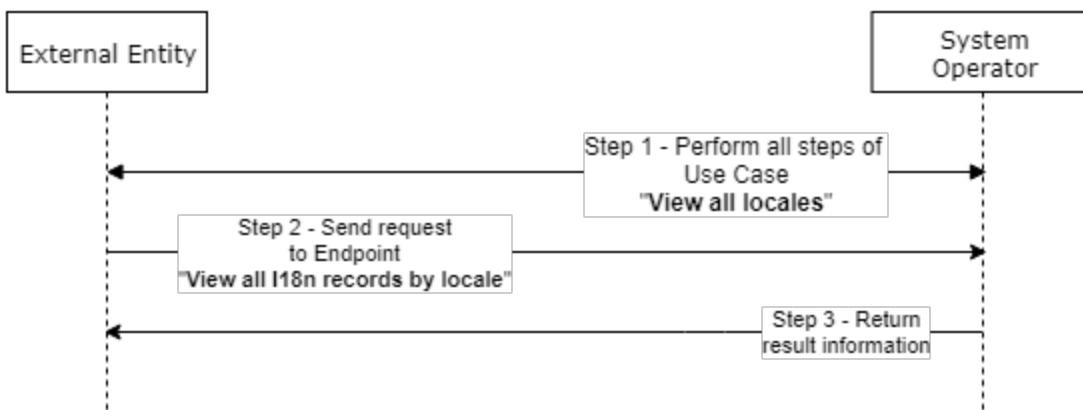
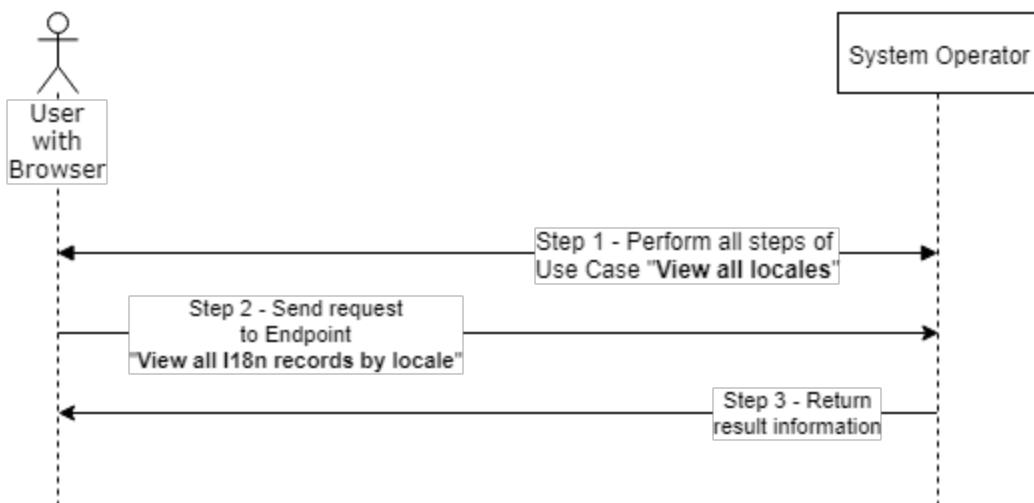
Post Conditions

Only locale-specific items are available.

Result example

```
{  
    "records": [  
        {  
            "id": 0,  
            "key": "string",  
            "values": [  
                {  
                    "locale": "en",  
                    "value": "translated_string_value"  
                }  
            ]  
        },  
        {"status": "ok",  
         "message": "string"}  
    ]  
}
```

View all I18n records by locale scheme

Use case: View all I18n records by locale**Basic FFlow****Optional Web UI Flow**

View all I18n records description

Use Case Name

View all I18n records

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View all I18n records”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions:
I18N_RECORD_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.

2. External Entity sends a request to Endpoint “View all I18n records”.

Endpoint URL: GET /i18n-records

Parameters: Security TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.

2. A user sends a request to Endpoint “View all I18n records”.

Endpoint URL: GET /i18n-records

Parameters: Security TOKEN

3. System Operator returns result information to User (See Result example below).

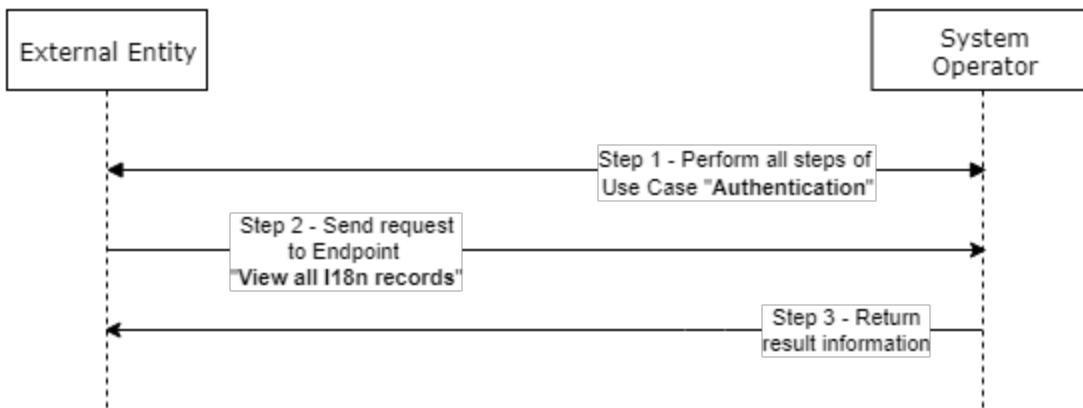
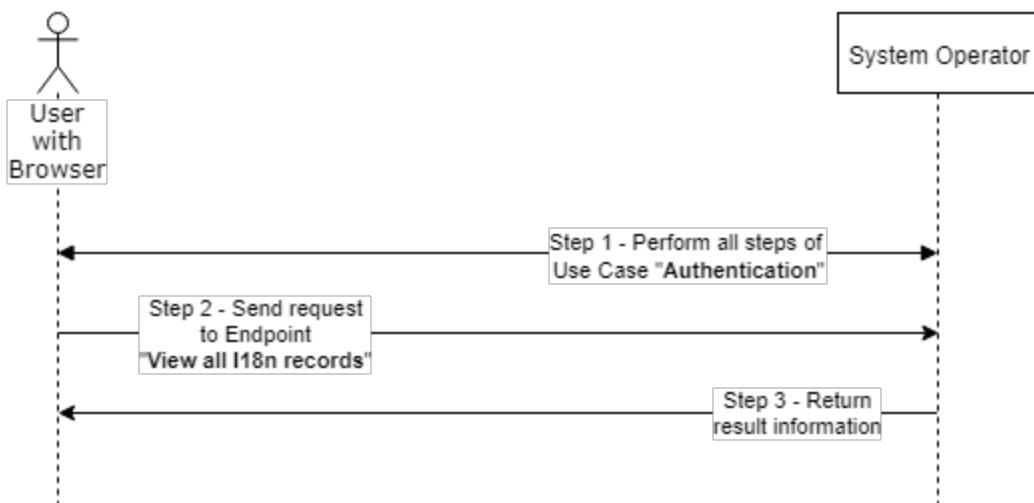
Post Conditions

List is available

Result example

```
{  
    "records": [  
        {  
            "id": 0,  
            "key": "string",  
            "values": [  
                {  
                    "locale": "en",  
                    "value": "translated_string_value"  
                }  
            ]  
        },  
        {"status": "ok",  
         "message": "string"}  
    ]  
}
```

[View all I18n records scheme](#)

Use case: View all I18n records**Basic FFlow****Optional Web UI Flow**

|
View all locales description

Use Case Name

View all locales

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View all locales”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View all locales”.

Endpoint URL: POST /i18n-records/view-locales

Parameters:

```
{
  "records": [
    "string"
  ],
  "status": "ok",
  "message": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View all locales”.

Endpoint URL: POST /i18n-records/view-locales

Parameters:

```
{  
    "records": [  
        "string"  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

3. System Operator returns xx result information to User (See Result example below).

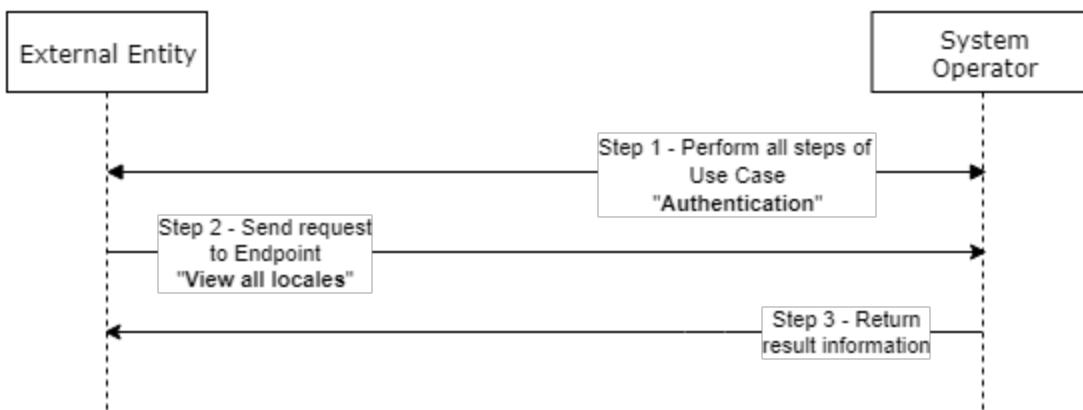
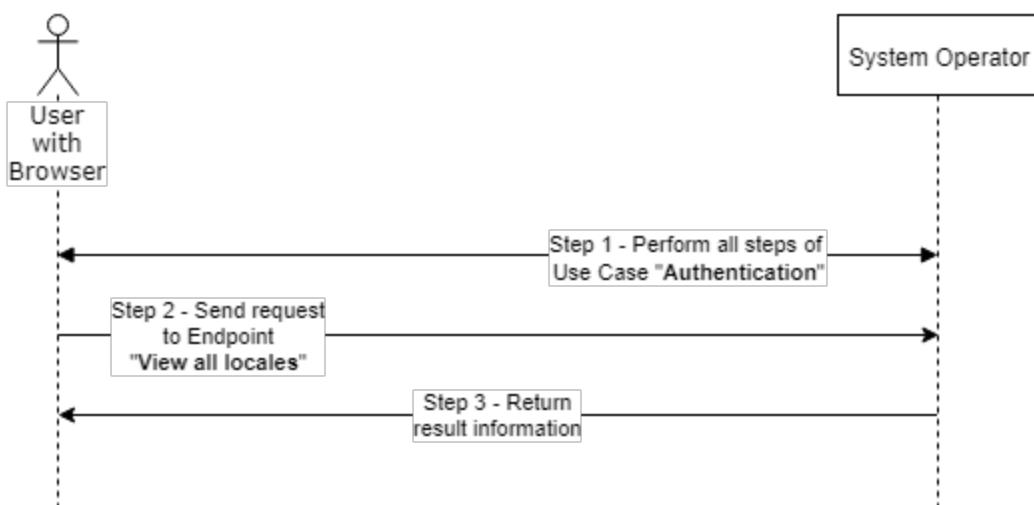
Post Conditions

List of Locales is available.

Result example

```
{  
    "records": [  
        "string"  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

[View all locales description scheme](#)

Use case: View all locales**Basic FFlow****Optional Web UI Flow**

View I18n records description

Use Case Name

View I18n records

Brief Description

A User or External Entity on behalf of a User with role permission I18N_RECORD_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “View I18n records”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: I18N_RECORD_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “View I18n records”.

Endpoint URL: POST /i18n-records/view

Parameters:

```
{
  "filter": {
    "ids": [
      0
    ],
    "key": "string",
    "value": "string"
  },
  "sort": {
    "key": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “View I18n records”.

Endpoint URL: POST /i18n-records/view

Parameters:

```
{
  "filter": {
    "ids": [
      0
    ],
    "key": "string",
    "value": "string"
  },
  "sort": {
    "key": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns result information to User (See Result example below).

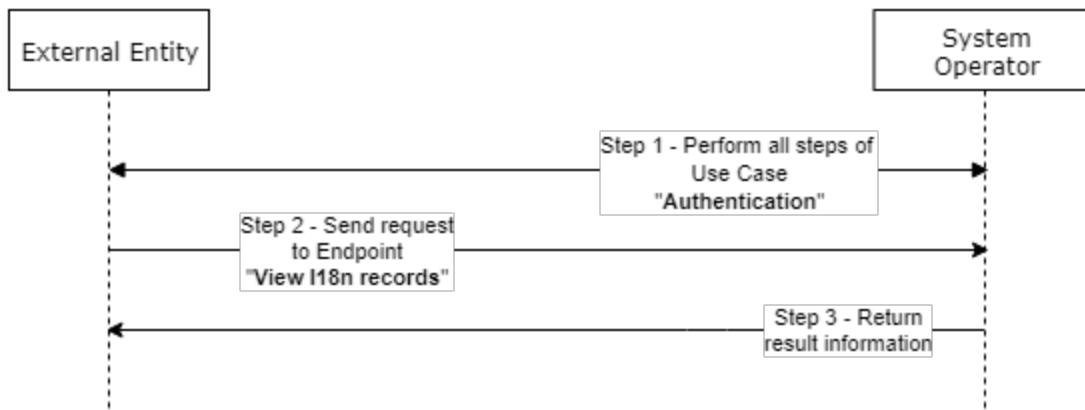
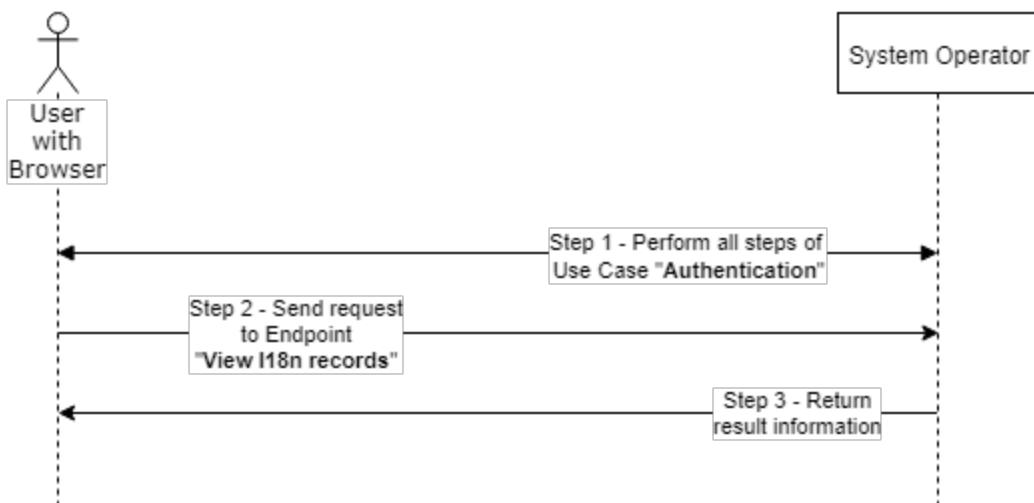
Post Conditions

Requested list is available.

Result example

```
{
  "records": [
    {
      "id": 0,
      "key": "string",
      "values": [
        {
          "locale": "en",
          "value": "translated_string_value"
        }
      ]
    }
  ],
  "status": "ok",
  "message": "string"
}
```

[View I18n records description scheme](#)

Use case: View I18n records**Basic FFlow****Optional Web UI Flow**

Support

Create a ticket to customer support description

Use Case Name

Create a ticket to customer support

Brief Description

A User or External Entity on behalf of a User with role permissions TICKET_OWNER or TICKET_MANAGER will go through all steps of “Retrieve the list of tickets as customer support manager” Use Case, and then send a request to Endpoint “Create a ticket to customer support”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TICKET_OWNER or TICKET_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient permission.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Retrieve the list of tickets as customer support manager”.
2. External Entity sends a request to Endpoint “Create a ticket to customer support”.

Endpoint URL: POST /tickets

Parameter:

```
{
  "category": "GENERAL",
  "topic": "string",
  "description": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Retrieve the list of tickets as customer support manager”.
2. A user sends a request to Endpoint “Create a ticket to customer support”.

Endpoint URL: POST /tickets

Parameter:

```
{
  "category": "GENERAL",
  "topic": "string",
  "description": "string"
}
```

3. System Operator returns result information to User (See Result example below).

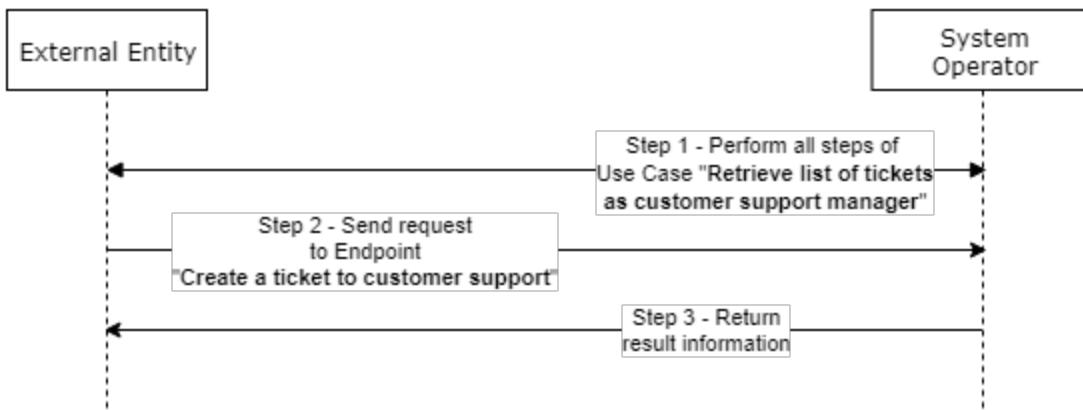
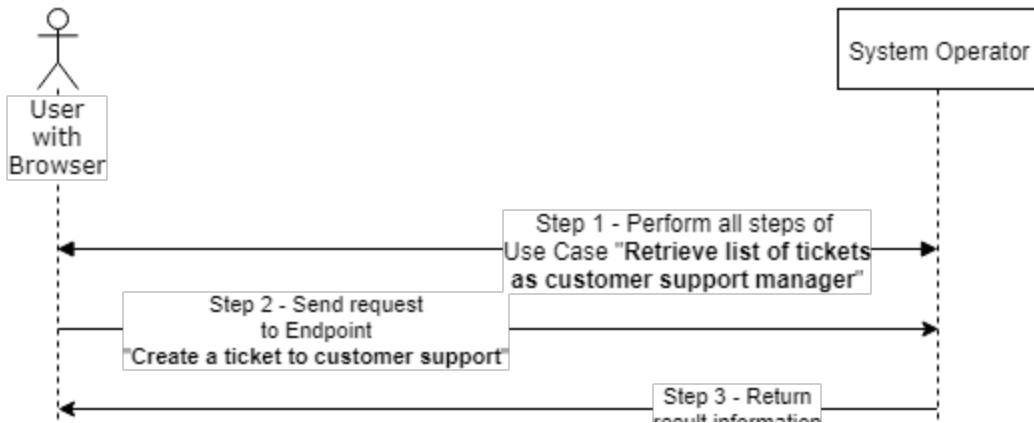
Post Conditions

The new ticket is visible on the ticket list.

Result example

```
{  
    "ticket": {  
        "id": "string",  
        "category": "GENERAL",  
        "conversation": {  
            "id": "string",  
            "name": "string",  
            "createdAt": "2018-07-25T12:12:34.533Z",  
            "initiator": {  
                "id": "string",  
                "type": "ORGANIZATION",  
                "entityId": "string"  
            }  
        }  
    },  
    "status": "ok",  
    "message": "string"  
}
```

Create a ticket to customer support scheme

Use case: Create a ticket to customer support**Basic FLow****Optional Web UI Flow**

Retrieve list of tickets as customer support manager description

Use Case Name

Retrieve the list of tickets as customer support manager

Brief Description

A User or External Entity on behalf of a User with role permission TICKET_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Retrieve list of tickets as customer support manager”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: TICKET_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Retrieve the list of tickets as customer support manager”.

Endpoint URL: POST /tickets/view

Parameter:

```
{
  "filter": {
    "id": "string",
    "conversationId": "string",
    "categories": [
      "GENERAL"
    ],
    "text": "string",
    "initiatorType": "ORGANIZATION",
    "initiatorEntityId": "string",
    "dateFrom": "2018-07-25T12:12:32.451Z",
    "dateTo": "2018-07-25T12:12:32.451Z"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns List of tickets to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Retrieve list of tickets as customer support manager”.

Endpoint URL: POST /tickets/view

Parameter:

```
{
  "filter": {
    "id": "string",
    "conversationId": "string",
    "categories": [
      "GENERAL"
    ],
    "text": "string",
    "initiatorType": "ORGANIZATION",
    "initiatorEntityId": "string",
    "dateFrom": "2018-07-25T12:12:32.451Z",
    "dateTo": "2018-07-25T12:12:32.451Z"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns List of tickets to External Entity. (See Result example below)

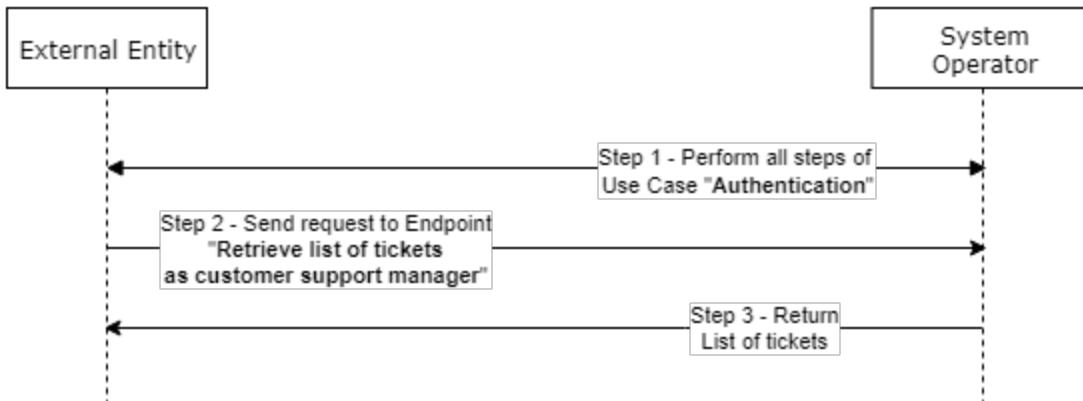
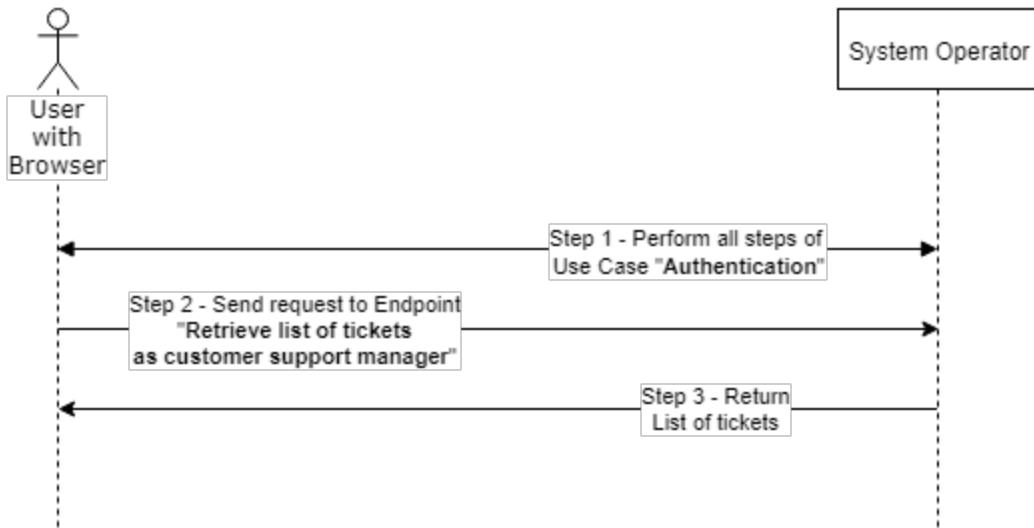
Post Conditions

List of tickets is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "name": "string",  
            "broadcast": false,  
            "unread": false,  
            "createdAt": "2018-07-25T12:12:34.539Z",  
            "lastMessageCreatedAt": "2018-07-25T12:12:34.539Z",  
            "initiator": {  
                "id": "string",  
                "type": "ORGANIZATION",  
                "entityId": "string"  
            },  
            "participants": [  
                {  
                    "id": "string",  
                    "type": "ORGANIZATION",  
                    "entityId": "string"  
                }  
            ]  
        }  
    ],  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0  
}
```

Retrieve list of tickets as customer support manager scheme

Use case: Retrieve list of tickets as customer support manager**Basic FLow****Optional Web UI Flow**

User message API provider

Create and send message to user description

Use Case Name

Create and send message to user

Brief Description

A User or External Entity on behalf of a User with role permission USER_MESSAGE_MANAGER will go through all steps of “Get all messages to user” Use Case (to obtain user ID and, if required, all messages), and then send a request to Endpoint “Create and send message to user”. Alternatively, “Get users” Use Case can be called directly to obtain user ID.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MESSAGE_MANAGER, e.g. head of compliance, administrator, compliance manager, or compliance specialist
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get all messages to user”.
2. External Entity sends a request to Endpoint “Create and send message to user”.

Endpoint URL: POST /users/message

```
Parameters: {
  "userId": "string",
  "message": "string"
}
```

1. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get all messages to user”.
2. A user sends a request to Endpoint “Create and send message to user”.

Endpoint URL: POST /users/message

```
Parameters: {
```

```
"userId": "string",
"message": "string"
}
```

1. System Operator returns result information to User (See Result example below).

Post Conditions

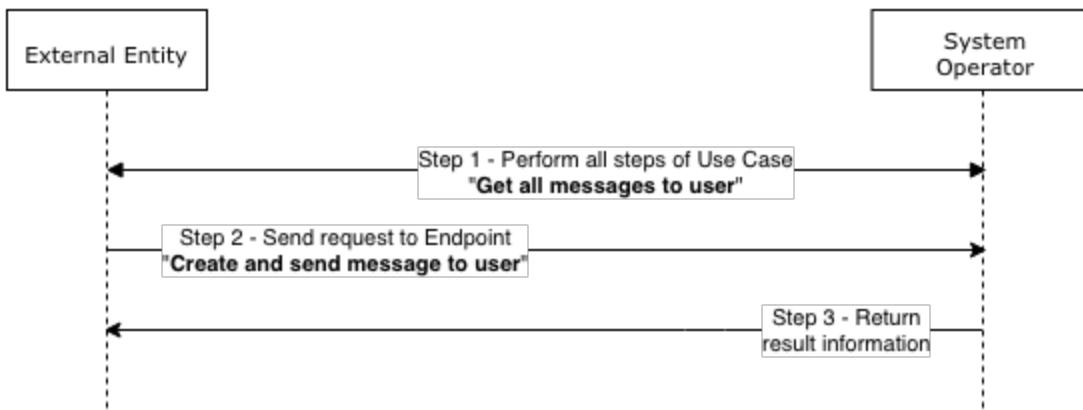
User ID is available.

Result example

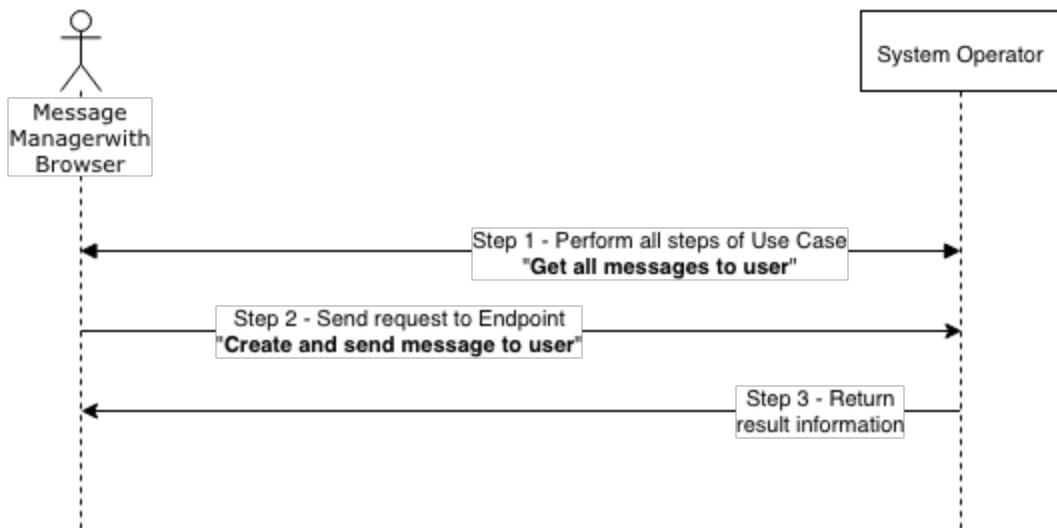
```
{
  "status": "ok",
  "message": "string"
}
Create and send message to user scheme
```

Use case: Create and send message to user

Basic FFlow



Optional Web UI Flow



Get all messages to user description

Use Case Name

Get all messages to user

Brief Description

A User or External Entity on behalf of a User with role permission USER_MESSAGE_MANAGER will go through all steps of “Get users” Use Case (to obtain user ID), and then send a request to Endpoint “Get all messages to user”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: USER_MESSAGE_MANAGER, e.g. head of compliance, administrator, compliance manager, or compliance specialist.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Get users”.
2. External Entity sends a request to Endpoint “Get all messages to user”.

Endpoint URL: GET /users/message/{userId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns messages to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Get users”.
2. A user sends a request to Endpoint “Get all messages to user”.

Endpoint URL: GET /users/message/{userId}

Parameters: TOKEN - identifies authenticated user

1. System Operator returns messages to User (See Result example below).

Post Conditions

User ID is available.

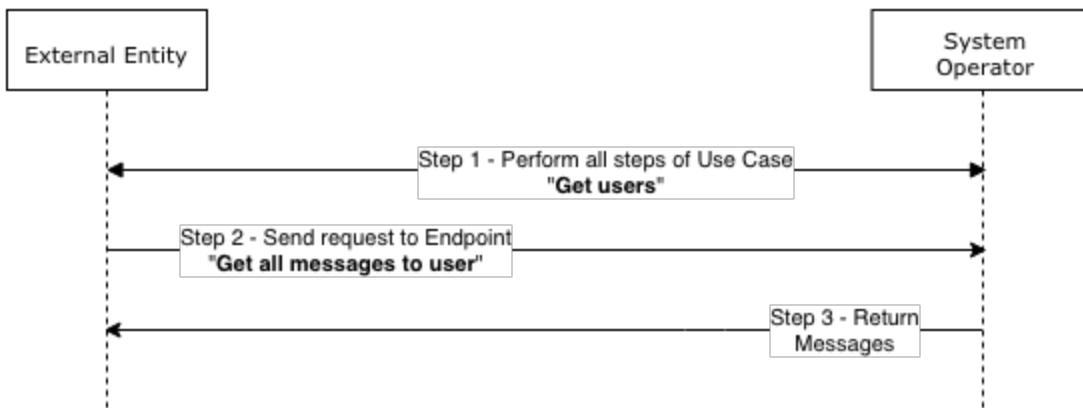
Result example

```
{  
  "messages": [  
    {  
      "userId": "string",  
      "message": "string",  
      "email": "string",  
      "phone": "string",  
      "createdAt": "2018-08-21T09:45:22.626Z"  
    }  
  ]  
}
```

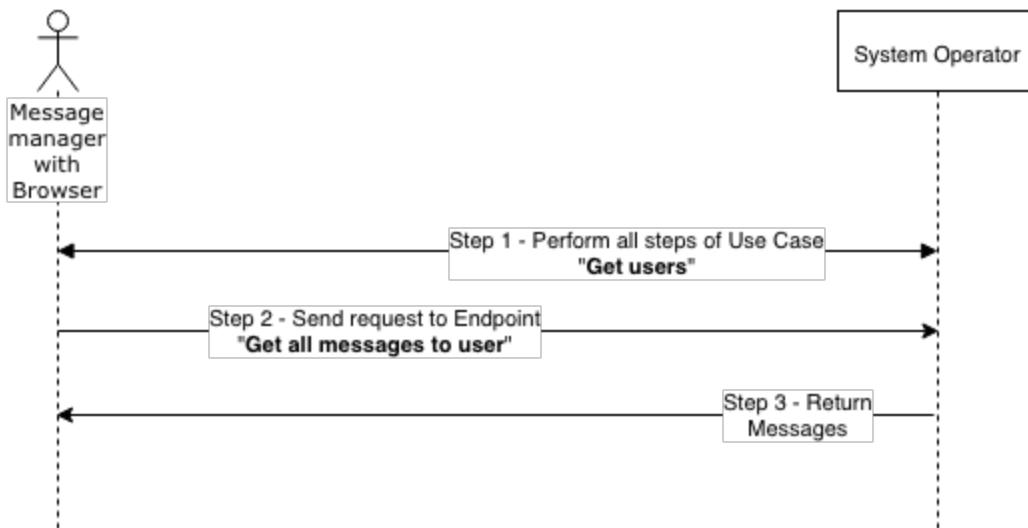
Get all messages to user scheme

Use case: Get all messages to user

Basic FFlow



Optional Web UI Flow



Callback settings for non-merchant operations

Create or update callback settings description

Use Case Name

Create or update callback settings

Brief Description

A User or External Entity on behalf of a User with role permission "CALLBACK_SETTINGS_OWNER" will go through all steps of "View callback settings" (to obtain trustAllCertificates and callbackUrl parameters where relevant) Use Case, and then send a request to Endpoint "Create or update callback settings". These settings are exclusive to the logged in merchant.

Note:

If a new callback settings record is being created, you may call "Authentication" EP first instead of "View callback settings" as long as you have the required parameters, e.g. callbackUrl, to create a new record. To change the callback settings secretCode, you can call "View callback settings" EP to change it.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CALLBACK_SETTINGS_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. User must have an System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "View callback settings".
2. External Entity sends a request to Endpoint "Create or update callback settings".

Endpoint URL: PUT /profiles/my/callback-settings

Parameters:

```
{
  "trustAllCertificates": false,
  "callbackUrl": "string",
  "active": false
}
```

3. System Operator returns details of new callback settings, including secretCode, to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "View callback settings".
2. A user sends a request to Endpoint "Create or update callback settings".

Endpoint URL: PUT /profiles/my/callback-settings

Parameters:

```
{  
    "trustAllCertificates": false,  
    "callbackUrl": "string",  
    "active": false  
}
```

3. System Operator returns details of new or updated callback settings, including secretCode, to User (See Result example below).

Post Conditions

New callback settings or those to be updated, e.g. callbackUrl, are available.

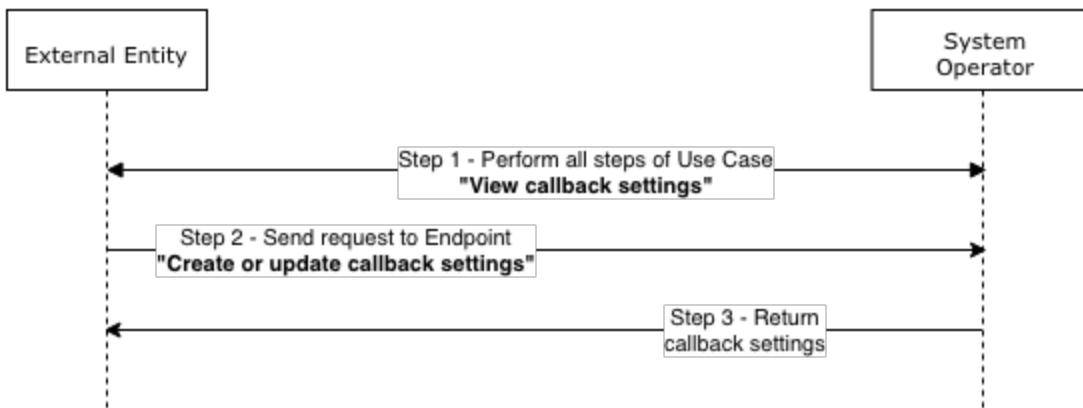
Result example

```
{  
    "callbackSettings": {  
        "secretCode": "string",  
        "trustAllCertificates": false,  
        "callbackUrl": "string",  
        "active": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

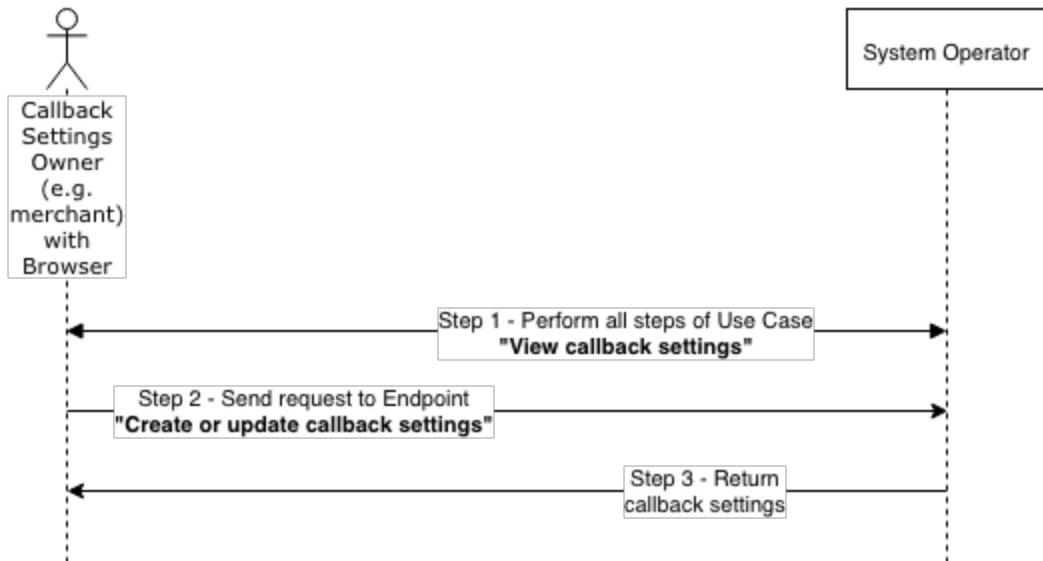
Create or update callback settings scheme

Use case: Create or update callback settings

Basic FLow



Optional Web UI Flow



Generate secret code for callback settings description

Use Case Name

Generate secret code for callback settings

Brief Description

A User or External Entity on behalf of a User with role permission "CALLBACK_SETTINGS_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "Generate secret code for callback settings". This EP will automatically update the user's callback settings with a new secret code.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CALLBACK_SETTINGS_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "Generate secret code for callback settings".

Endpoint URL: POST /profiles/my/callback-settings/generate-secret-code

Parameters: TOKEN - identifies authenticated user

3. System Operator returns call settings information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "Generate secret code for callback settings".

Endpoint URL: POST /profiles/my/callback-settings/generate-secret-code

Parameters: TOKEN - identifies authenticated user

3. System Operator returns call settings information to User (See Result example below).

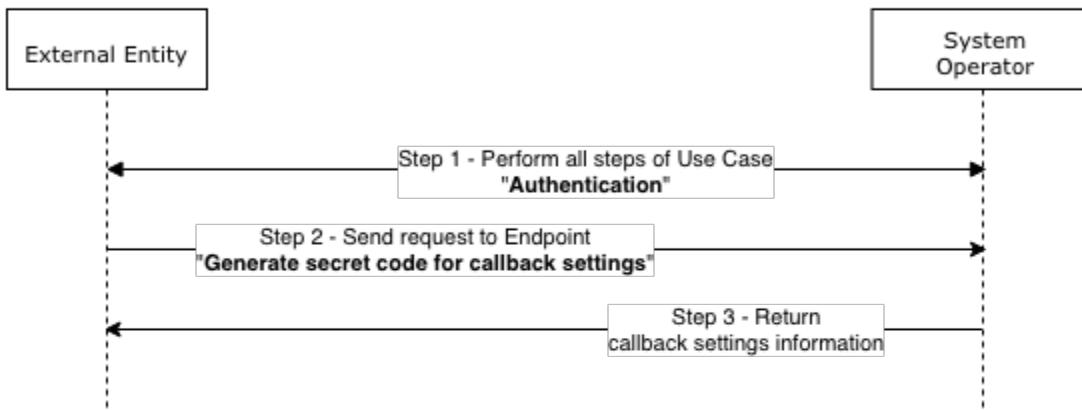
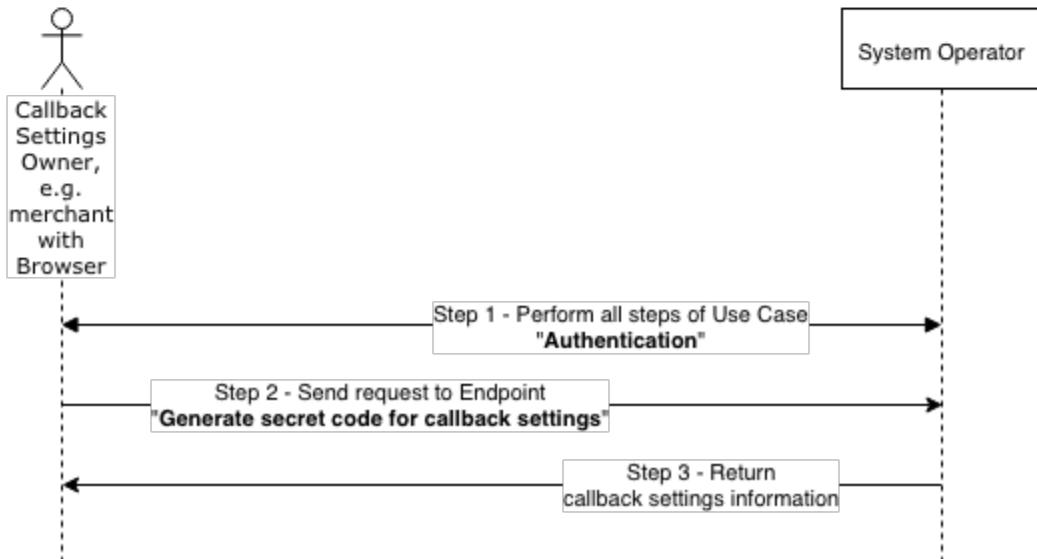
Post Conditions

Callback settings record is available.

Result example

```
{
  "callbackSettings": {
    "secretCode": "string",
    "trustAllCertificates": false,
    "callbackUrl": "string",
    "active": false
  },
  "status": "ok",
  "message": "string"
}
```

Generate secret code for callback settings scheme

Use case: Generate secret code for callback settings**Basic Flow****Optional Web UI Flow**

[View callback settings description](#)

Use Case Name

[View callback settings](#)

Brief Description

A User or External Entity on behalf of a User with role permission "CALLBACK_SETTINGS_OWNER" will go through all steps of "Authentication" Use Case, and then send a request to Endpoint "View callback settings".

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: "CALLBACK_SETTINGS_OWNER", e.g. merchant.
2. System Operator running "SDK.Finance" software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case "Authentication".
2. External Entity sends a request to Endpoint "View callback settings".

Endpoint URL: GET /profiles/my/callback-settings

Parameters: TOKEN - identifies authenticated user

3. System Operator returns callback settings to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case "Authentication".
2. A user sends a request to Endpoint "View callback settings".

Endpoint URL: GET /profiles/my/callback-settings

Parameters: TOKEN - identifies authenticated user

1. System Operator returns callback settings to User (See Result example below).

Post Conditions

The token is available.

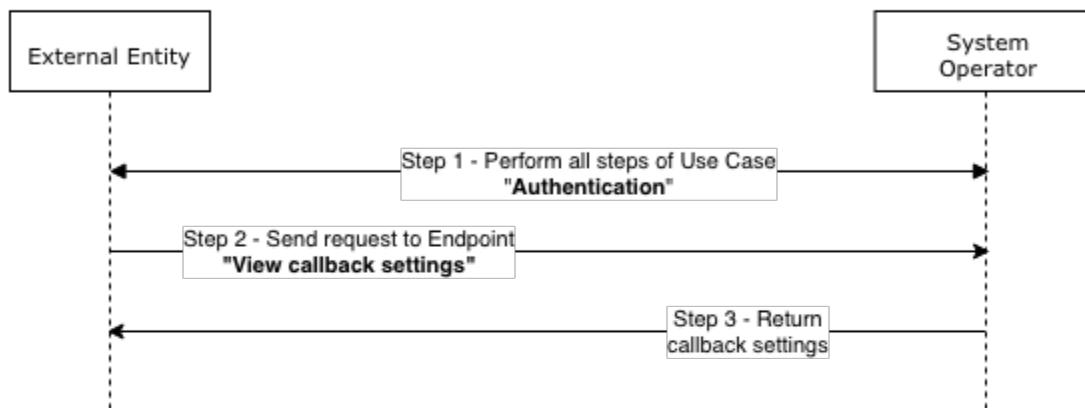
Result example

```
{
  "callbackSettings": {
    "secretCode": "string",
    "trustAllCertificates": false,
    "callbackUrl": "string",
    "active": false
  },
  "status": "ok",
  "message": "string"
}
```

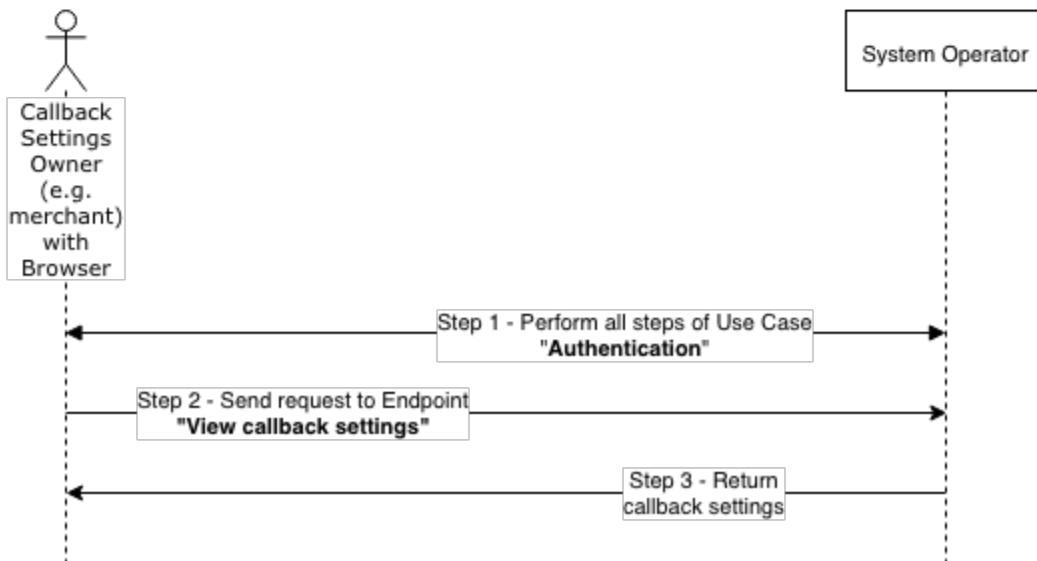
View callback settings scheme

Use case: View callback settings

Basic FFlow



Optional Web UI Flow



In-system communication

Create system broadcasts description

Use Case Name

Create system broadcasts

Brief Description

A User or External Entity on behalf of a User with role permission SYSTEM_BROADCAST_CREATION_EXECUTOR will go through all steps of one of these Use Cases: “Get users”, “View organizations” which will provide IDs for organizations or users. These IDs will be used to specify the audience for the broadcasted message produced by sending a request to Endpoint “Create system broadcasts”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: SYSTEM_BROADCAST_CREATION_EXECUTOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of one of these Use Cases: “Get users”, “View organizations”.
2. External Entity sends a request to Endpoint “Create system broadcasts”.

Endpoint URL: POST /conversations/create-system-broadcasts

Parameters:

```
{
  "mode": "ALL_USERS_WITH_IDS",
  "organizationTypes": "[\"individual\"]",
  "ids": [
    "string"
  ],
  "title": "string",
  "creationPolicy": "CREATE_NEW",
  "conversationId": "string",
  "text": "string",
  "fileIds": [
    "string"
  ]
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of one of these Use Cases: “Get users”, “View organizations”.
2. A user sends a request to Endpoint “Create system broadcasts”.

Endpoint URL: POST /conversations/create-system-broadcasts

Parameters:

```
{
  "mode": "ALL_USERS_WITH_IDS",
  "organizationTypes": "[\"individual\"]",
  "ids": [
    "string"
  ],
  "title": "string",
  "creationPolicy": "CREATE_NEW",
  "conversationId": "string",
  "text": "string",
  "fileIds": [
    "string"
  ]
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

System broadcast is created.

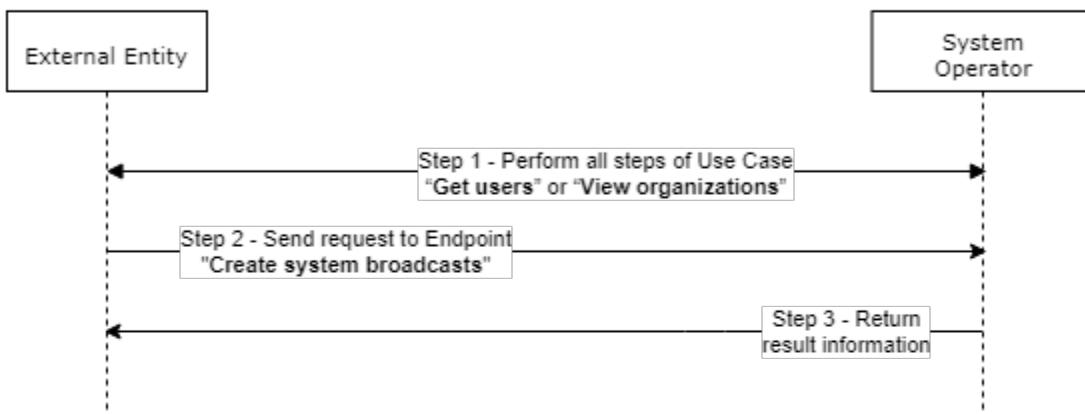
Result example

```
{
  "conversation": {
    "id": "string",
    "name": "string",
    "broadcast": false,
    "unread": false,
    "createdAt": "2018-08-10T14:07:38.781Z",
    "lastMessageCreatedAt": "2018-08-10T14:07:38.781Z",
    "initiator": {
      "id": "string",
      "type": "ORGANIZATION",
      "entityId": "string"
    },
    "participants": [
      {
        "id": "string",
        "type": "ORGANIZATION",
        "entityId": "string"
      }
    ]
  },
  "status": "ok",
  "message": "string"
}
```

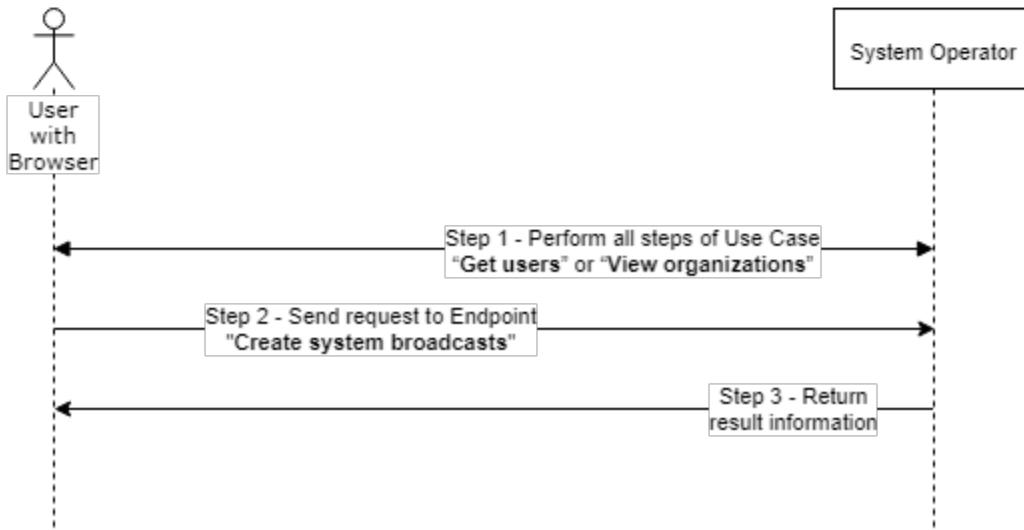
Create system broadcasts scheme

Use case: Create system broadcasts

Basic FLow



Optional Web UI Flow



Delete a conversation description

Use Case Name**Delete a conversation****Brief Description**

A User or External Entity on behalf of a User with role permission CONVERSATION_DELETION_EXECUTOR will go through all steps of “Retrieve the list of conversations by specified filter” Use Case, and then send a request to Endpoint “Delete a conversation”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_DELETION_EXECUTOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Retrieve the list of conversations by specified filter”.
2. External Entity sends a request to Endpoint “Delete a conversation”.

Endpoint URL: DELETE /conversations/{conversationId}

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Retrieve the list of conversations by specified filter”.
2. A user sends a request to Endpoint “Delete a conversation”.

Endpoint URL: DELETE /conversations/{conversationId}

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

Conversation is deleted.

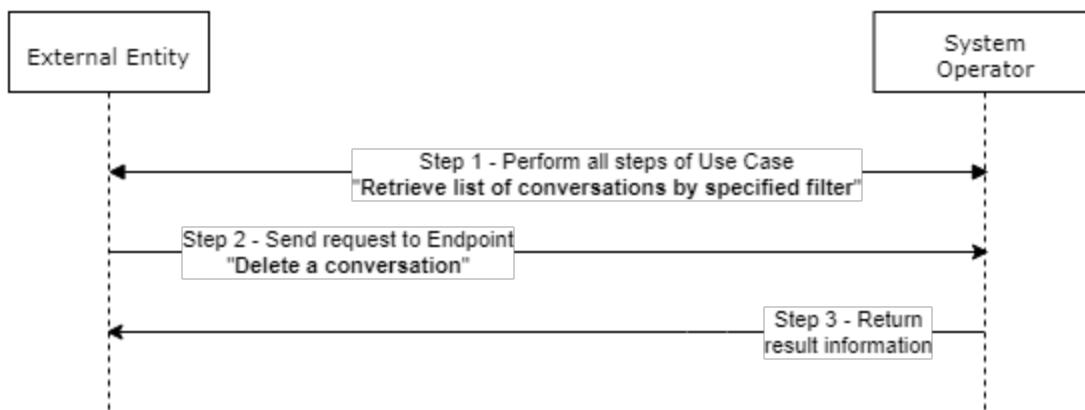
Result example

```
{
  "status": "ok",
  "message": "string"
}
```

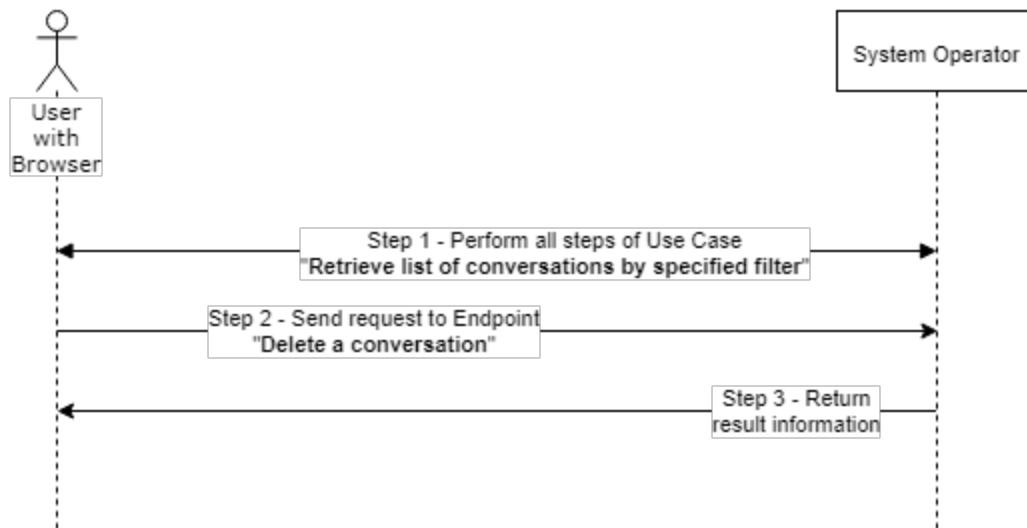
Delete a conversation scheme

Use case: Delete a conversation

Basic FFlow



Optional Web UI Flow



Get files attached to the specified message description

Use Case Name

Get files attached to the specified message

Brief Description

A User or External Entity on behalf of a User with role permission CONVERSATION_VIEWER will go through all steps of “Retrieve the list of messages for specified conversation” Use Case, and then send a request to Endpoint “Get files attached to the specified message”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_VIEWER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Retrieve the list of messages for specified conversation”.
 2. External Entity sends a request to Endpoint “Get files attached to the specified message”.
- Endpoint URL: GET /conversations/{conversationId}/messages/{ messageId }/media-files
Parameters: TOKEN
3. System Operator returns List of files to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Retrieve the list of messages for specified conversation”.
 2. A user sends a request to Endpoint “Get files attached to the specified message”.
- Endpoint URL: GET /conversations/{conversationId}/messages/{ messageId }/media-files
Parameters: TOKEN
3. System Operator returns List of files to User (See Result example below).

Post Conditions

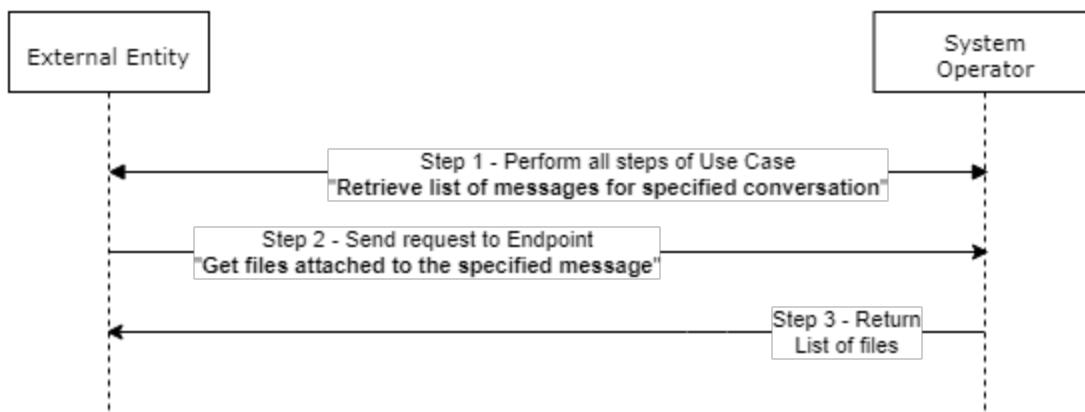
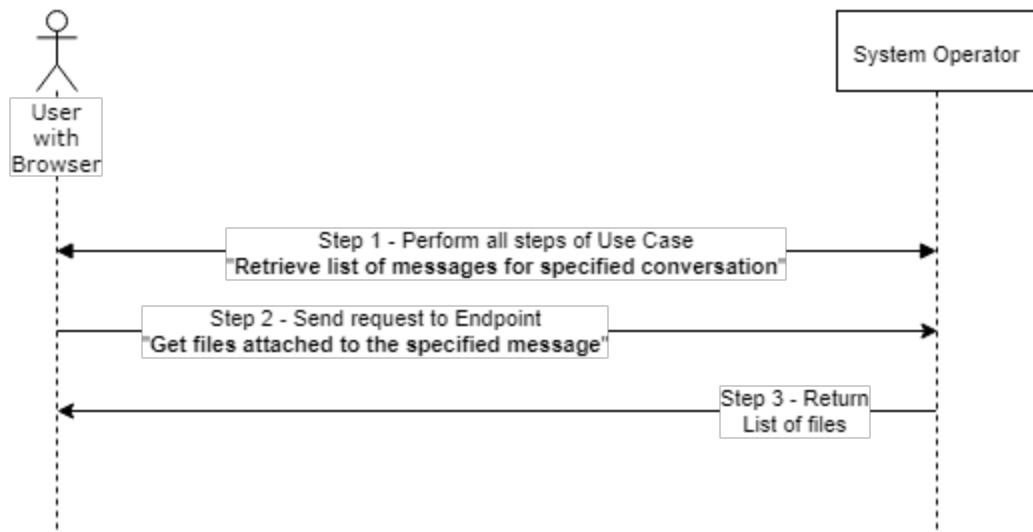
Files are available.

Result example

```
{  
  "files": [  
    {  
      "id": "string",  
      "ownerId": "string",  
      "mediaType": "string",  
      "name": "string",  
      "url": "string",  
      "md5": "string",  
      "sha1": "string",  
      "size": 0,  
      "used": false,  
      "createdAt": "2018-08-10T14:07:38.824Z",  
      "expiresAt": "2018-08-10T14:07:38.824Z",  
      "tag": "string"  
    }  
  ],  
  "status": "ok",  
  "message": "string"  
}
```

Get files attached to the specified message scheme

Use case: Get files attached to the specified message

Basic FLow**Optional Web UI Flow**

Mark message as read description

Use Case Name

Mark message as read

Brief Description

A User or External Entity on behalf of a User with role permission CONVERSATION_VIEWER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Mark message as read”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_VIEWER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Mark message as read”.

Endpoint URL: POST

/conversations/{conversationId}/messages/{messageId}/mark-as-read

Parameters: TOKEN

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Mark message as read”.

Endpoint URL: POST

/conversations/{conversationId}/messages/{messageId}/mark-as-read

Parameters: TOKEN

3. System Operator returns result information to User (See Result example below).

Post Conditions

The message is marked.

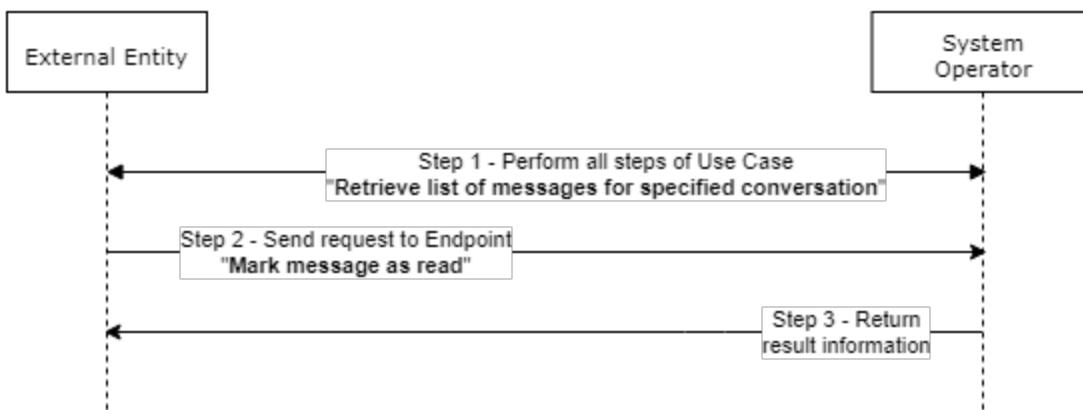
Result example

```
{  
    "value": {  
        "id": "string",  
        "sender": {  
            "id": "string",  
            "type": "ORGANIZATION",  
            "entityId": "string"  
        },  
        "text": "string",  
        "createdAt": "2018-08-10T14:07:38.817Z",  
        "updatedAt": "2018-08-10T14:07:38.817Z",  
        "read": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

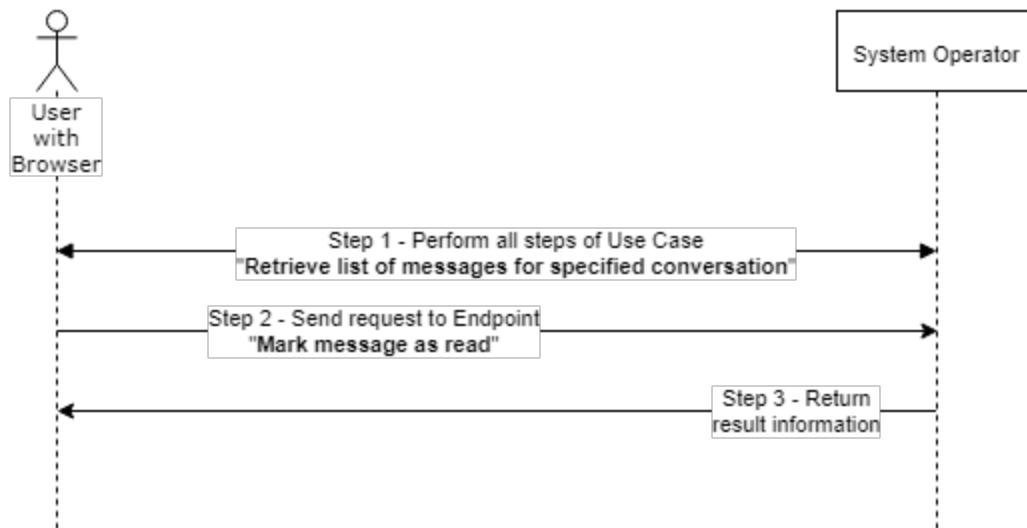
Mark message as read scheme

Use case: Mark message as read

Basic FFlow



Optional Web UI Flow



Retrieve list of conversations by specified filter description

Use Case Name

Retrieve the list of conversations by the specified filter

Brief Description

A User or External Entity on behalf of a User with role permission CONVERSATION_VIEWER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Retrieve the list of conversations by specified filter”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_VIEWER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Retrieve the list of conversations by specified filter”.

Endpoint URL: POST /conversations/view

Parameters:

```
{
  "filter": {
    "id": "string",
    "broadcast": false,
    "name": "string",
    "text": "string",
    "initiatorType": "ORGANIZATION",
    "initiatorEntityId": "string",
    "unread": false,
    "dateFrom": "2018-08-10T14:07:37.830Z",
    "dateTo": "2018-08-10T14:07:37.830Z",
    "lastMessageDateFrom": "2018-08-10T14:07:37.830Z",
    "lastMessageDateTo": "2018-08-10T14:07:37.830Z"
  },
  "sort": {
    "date": "asc",
    "lastMessageDate": "asc",
    "unread": "asc"
  },
  "pageNumber": 0,
  "pageSize": 10
}
```

3. System Operator returns List of Conversations to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Retrieve the list of conversations by specified filter”.

Endpoint URL: POST /conversations/view

Parameters:

```
{  
    "filter": {  
        "id": "string",  
        "broadcast": false,  
        "name": "string",  
        "text": "string",  
        "initiatorType": "ORGANIZATION",  
        "initiatorEntityId": "string",  
        "unread": false,  
        "dateFrom": "2018-08-10T14:07:37.830Z",  
        "dateTo": "2018-08-10T14:07:37.830Z",  
        "lastMessageDateFrom": "2018-08-10T14:07:37.830Z",  
        "lastMessageDateTo": "2018-08-10T14:07:37.830Z"  
    },  
    "sort": {  
        "date": "asc",  
        "lastMessageDate": "asc",  
        "unread": "asc"  
    },  
    "pageNumber": 0,  
    "pageSize": 10  
}
```

3. System Operator returns List of Conversations to User (See Result example below).

Post Conditions

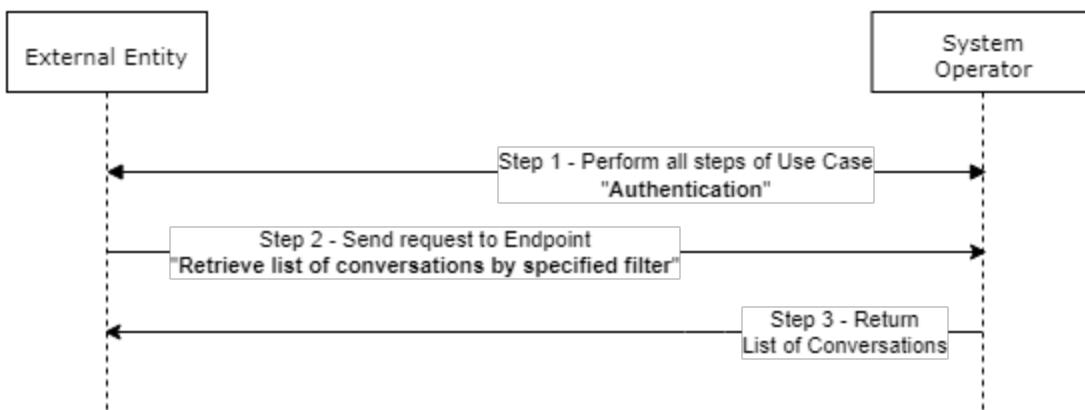
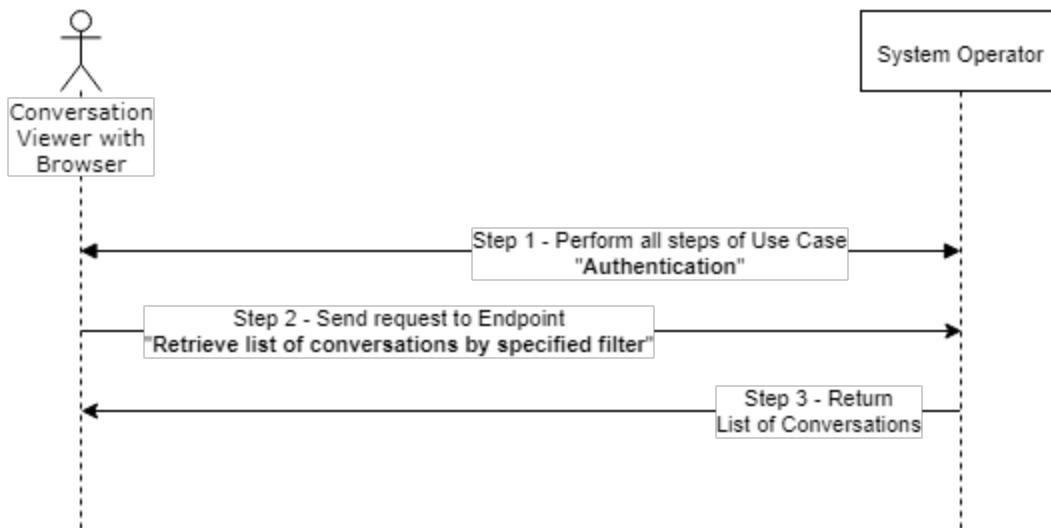
Requested list is available.

Result example

```
{  
    "records": [  
        {  
            "id": "string",  
            "name": "string",  
            "broadcast": false,  
            "unread": false,  
            "createdAt": "2018-08-10T14:07:38.788Z",  
            "lastMessageCreatedAt": "2018-08-10T14:07:38.788Z",  
            "initiator": {  
                "id": "string",  
                "type": "ORGANIZATION",  
                "entityId": "string"  
            },  
            "participants": [  
                {  
                    "id": "string",  
                    "type": "ORGANIZATION",  
                    "entityId": "string"  
                }  
            ]  
        }  
    ],  
    "status": "ok",  
    "message": "string",  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalRecords": 0,  
    "totalPages": 0  
}
```

Retrieve list of conversations by specified filter scheme

Use case: Retrieve list of conversations by specified filter

Basic FLow**Optional Web UI Flow**

Retrieve list of messages for specified conversation description

Use Case Name

Retrieve the list of messages for the specified conversation

Brief Description

A User or External Entity on behalf of a User with role permission CONVERSATION_VIEWER will go through all steps of “Retrieve the list of conversations by specified filter” Use Case, and then send a request to Endpoint “Retrieve the list of messages for specified conversation”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_VIEWER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Retrieve the list of conversations by specified filter”.
2. External Entity sends a request to Endpoint “Retrieve the list of messages for a specified conversation”.

Endpoint URL: POST /conversations/{conversationId}/messages/view

Parameters:

```
{
  "filter": {
    "messageId": "string",
    "text": "string",
    "senderType": "ORGANIZATION",
    "senderEntityId": "string",
    "unread": false,
    "dateFrom": "2018-08-10T14:07:37.834Z",
    "dateTo": "2018-08-10T14:07:37.834Z"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns List of messages to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Retrieve the list of conversations by specified filter”.
2. A user sends a request to Endpoint “Retrieve the list of messages for specified conversation”.

Endpoint URL: POST /conversations/{conversationId}/messages/view

Parameters:

```
{
  "filter": {
    "messageId": "string",
    "text": "string",
    "senderType": "ORGANIZATION",
    "senderEntityId": "string",
    "unread": false,
    "dateFrom": "2018-08-10T14:07:37.834Z",
    "dateTo": "2018-08-10T14:07:37.834Z"
  },
  "sort": {
    "date": "asc"
  },
  "pageNumber": 0,
  "pageSize": 0
}
```

3. System Operator returns List of messages to User (See Result example below).

Post Conditions

Message list is available.

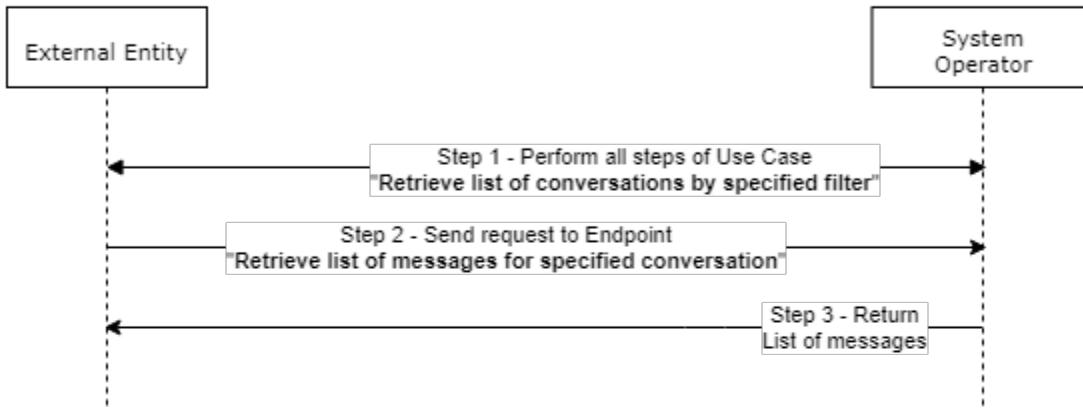
Result example

```
{
  "messages": [
    {
      "id": "string",
      "sender": {
        "id": "string",
        "type": "ORGANIZATION",
        "entityId": "string"
      },
      "text": "string",
      "createdAt": "2018-08-10T14:07:38.809Z",
      "updatedAt": "2018-08-10T14:07:38.809Z",
      "read": false
    }
  ],
  "status": "ok",
  "message": "string",
  "pageNumber": 0,
  "pageSize": 0,
  "totalRecords": 0,
  "totalPages": 0
}
```

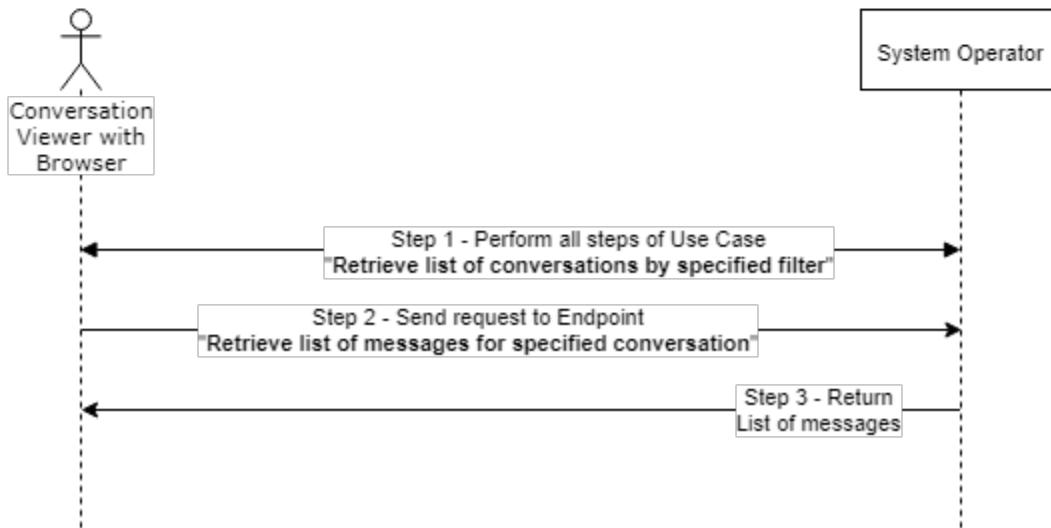
Retrieve list of messages for specified conversation scheme

Use case: Retrieve list of messages for specified conversation

Basic FLow



Optional Web UI Flow



Write a message as participant of the specified conversation description

Use Case Name

Write a message as a participant of the specified conversation

Brief Description

A User or External Entity on behalf of a User with role permission CONVERSATION_MESSAGE_CREATION_EXECUTOR will go through all steps of “Retrieve the list of messages for specified conversation” Use Case and then send a request to Endpoint “Write a message as the participant of the specified conversation”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: CONVERSATION_MESSAGE_CREATION_EXECUTOR.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile with sufficient access privileges.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Retrieve the list of messages for specified conversation”.
2. External Entity sends a request to Endpoint “Write a message as a participant of the specified conversation”.

Endpoint URL: POST /conversations/{conversationId}/messages

Parameters:

```
{
  "text": "string"
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Retrieve the list of messages for specified conversation”.
2. A user sends a request to Endpoint “Write a message as a participant of the specified conversation”.

Endpoint URL: POST /conversations/{conversationId}/messages

Parameters:

```
{
  "text": "string"
}
```

3. System Operator returns result information to User (See Result example below).

Post Conditions

The message is sent.

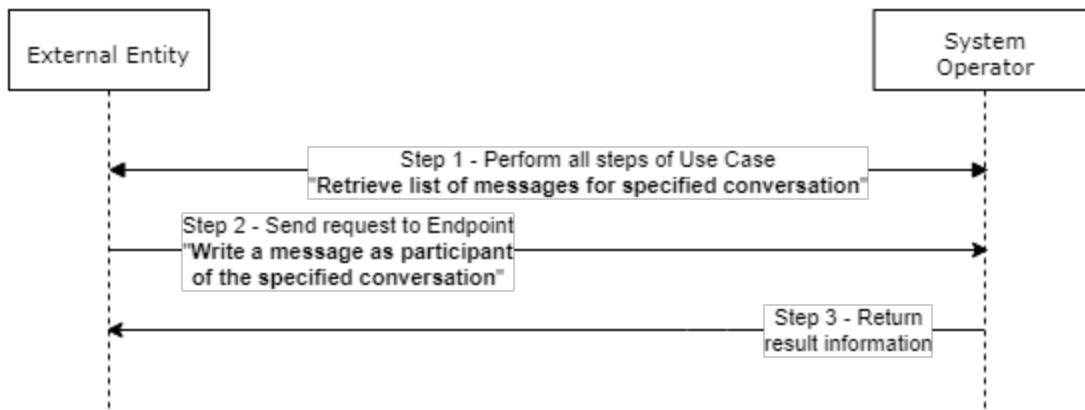
Result example

```
{  
    "value": {  
        "id": "string",  
        "sender": {  
            "id": "string",  
            "type": "ORGANIZATION",  
            "entityId": "string"  
        },  
        "text": "string",  
        "createdAt": "2018-08-10T14:07:38.800Z",  
        "updatedAt": "2018-08-10T14:07:38.800Z",  
        "read": false  
    },  
    "status": "ok",  
    "message": "string"  
}
```

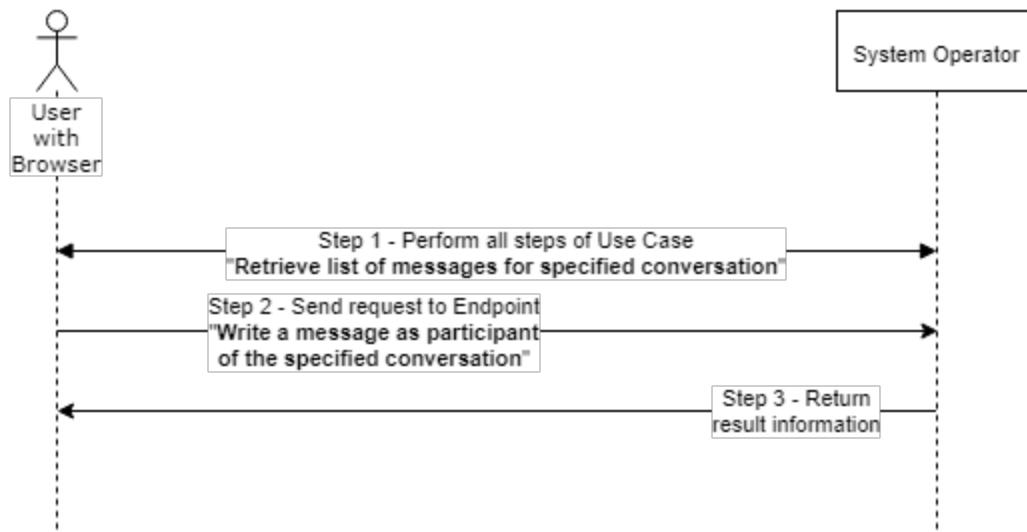
Write a message as participant of the specified conversation scheme

Use case: Write a message as participant of the specified conversation

Basic FFlow



Optional Web UI Flow



Environment variables management

Get list of environment variables description

Use Case Name

Get a list of environment variables

Brief Description

A User or External Entity on behalf of a User with role permission ENVIRONMENT_VARIABLE_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get the list of environment variables”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ENVIRONMENT_VARIABLE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get the list of environment variables”.

Endpoint URL: GET /environment-variables

Parameter: Security TOKEN.

3. System Operator returns List of variables to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get the list of environment variables”.

Endpoint URL: GET /environment-variables

Parameter: Security TOKEN.

3. System Operator returns List of variables to User (See Result example below).

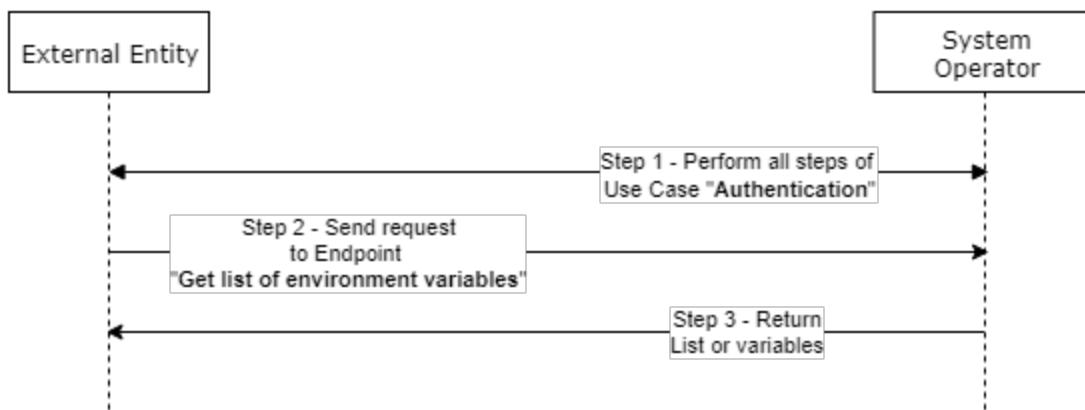
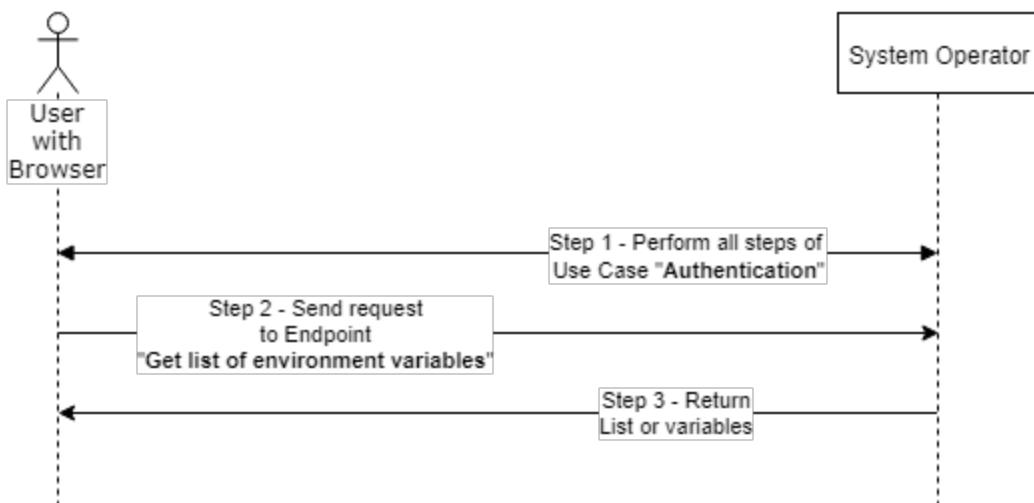
Post Conditions

All variables are available

Result example

```
{
  "variables": [
    {
      "key": "string",
      "value": {
        ...
      }
    }
  ],
  "status": "ok",
  "message": "string"
}
```

Get list of environment variables scheme

Use case: Get list of environment variables**Basic FFlow****Optional Web UI Flow**

Update multiple variables description

Use Case Name

Update multiple variables

Brief Description

A User or External Entity on behalf of a User with role permission ENVIRONMENT_VARIABLE_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Update multiple variables”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ENVIRONMENT_VARIABLE_MANAGER
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Update multiple variables”.

Endpoint URL: POST /environment-variables/batch-update

Parameter:

```
{
  "variables": [
    {
      "key": "string",
      "value": {
        ...
      }
    }
  ]
}
```

3. System Operator returns result information to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Update multiple variables”.

Endpoint URL: POST /environment-variables/batch-update

Parameter:

```
{  
    "variables": [  
        {  
            "key": "string",  
            "value": {  
                }  
        }  
    ]  
}
```

3. System Operator returns result information to User (See Result example below).

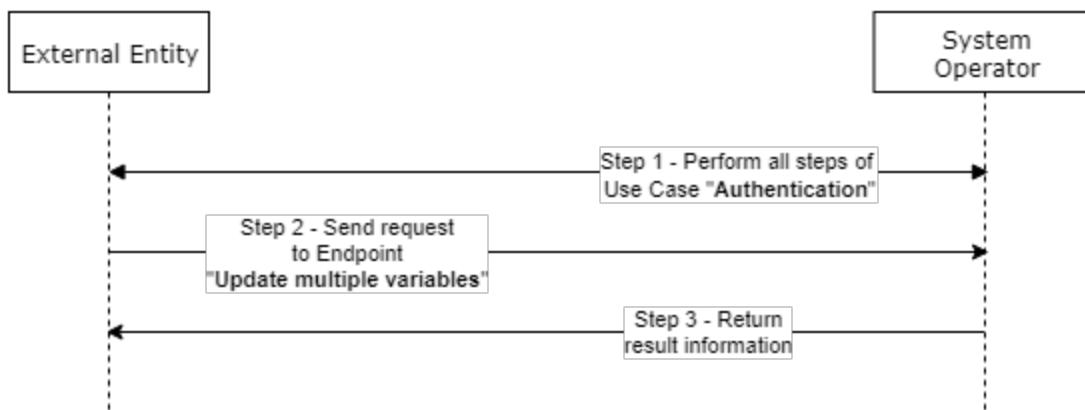
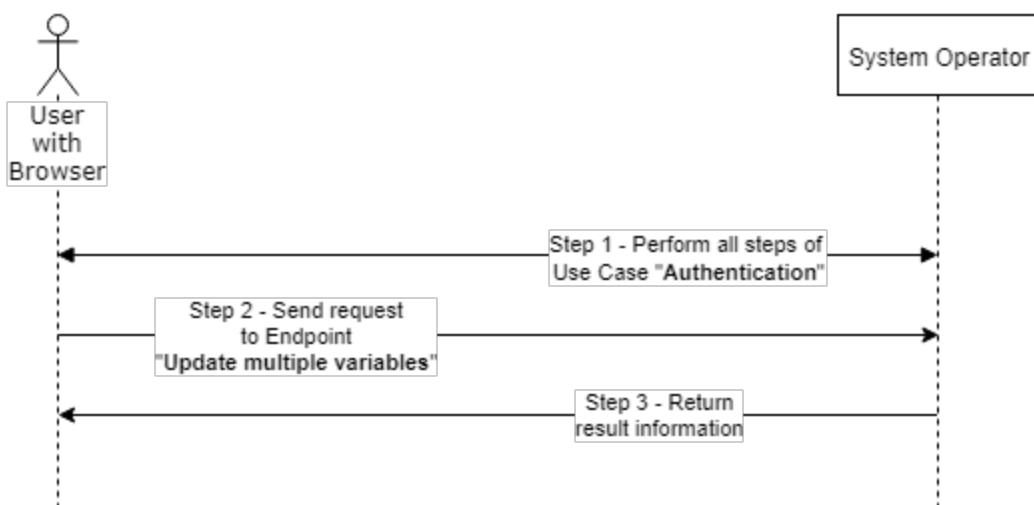
Post Conditions

New variables' data is available for usage.

Result example

```
{  
    "status": "ok",  
    "message": "string"  
}
```

Update multiple variables scheme

Use case: Update multiple variables**Basic FFlow****Optional Web UI Flow**

Error codes

Get list of error codes description

Use Case Name

Get the list of error codes

Brief Description

A User or External Entity on behalf of a User with role permission ENVIRONMENT_VARIABLE_MANAGER will go through all steps of “Authentication” Use Case, and then send a request to Endpoint “Get list of error codes”.

Actors

1. External Entity that can interact with System Operator API acting as a registered System Operator User with permissions: ENVIRONMENT_VARIABLE_MANAGER.
2. System Operator running “SDK.Finance” software and exposing portfolio of financial APIs.

Preconditions

1. The user must have a System Operator profile.

Basic Flow

By default, this flow assumes that External Entity sends Requests to System Operator Endpoints and System Operator sends back Responses to External Entity.

1. Perform all steps of Use Case “Authentication”.
2. External Entity sends a request to Endpoint “Get the list of error codes”.

Endpoint URL: GET /error-codes

Parameter: Security TOKEN.

3. System Operator returns List of Errors to External Entity. (See Result example below)

Optional Flow with Web Browser UI

1. Perform all steps of Use Case “Authentication”.
2. A user sends a request to Endpoint “Get the list of error codes”.

Endpoint URL: GET /error-codes

Parameter: Security TOKEN.

3. System Operator returns List of Errors to User (See Result example below).

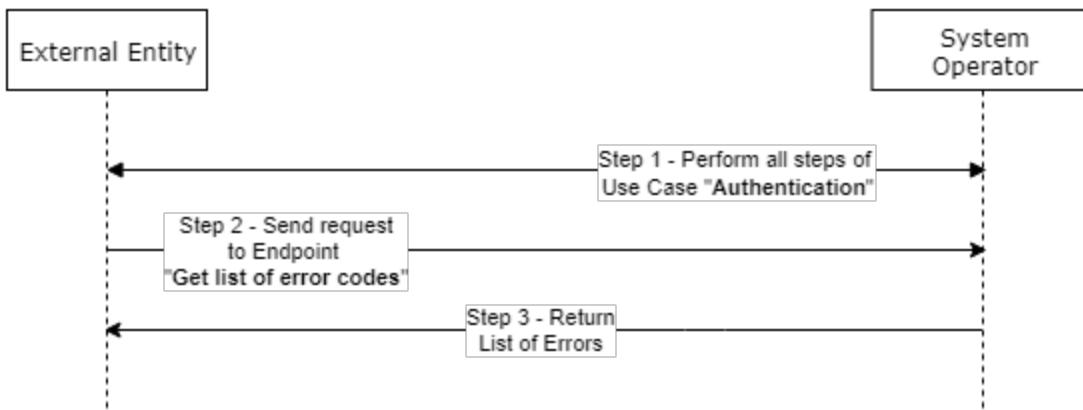
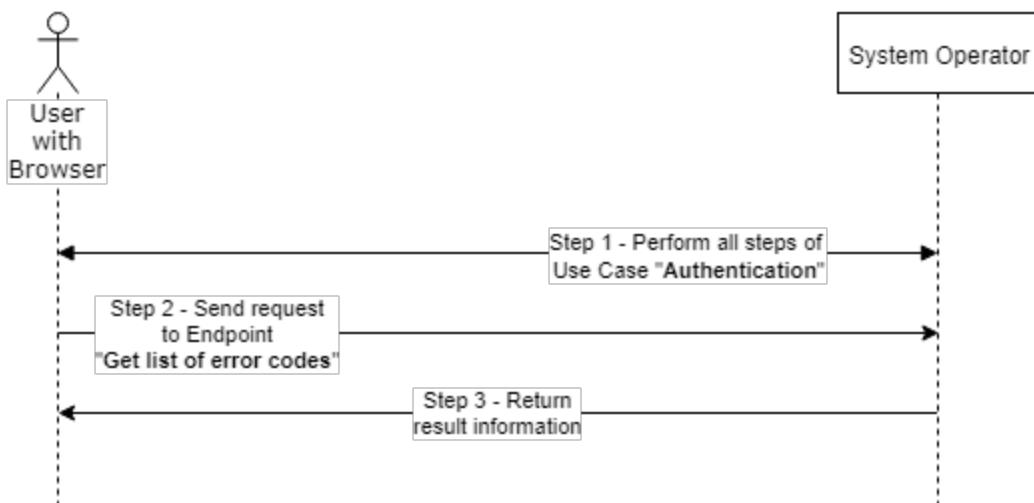
Post Conditions

Message is available

Result example

```
{  
    "records": [  
        {  
            "code": "string",  
            "message": "string"  
        }  
    ],  
    "status": "ok",  
    "message": "string"  
}
```

Get list of error codes scheme

Use case: Get list of error codes**Basic FFlow****Optional Web UI Flow**

Postman Collections

As a developer you can use POSTMAN collections for fast API testing with POSTMAN service here - <https://www.getpostman.com/>

