

Client Application Development Guide

Version 6.2

It's smarter, it's safer. It's VO.



Copyright

The contents of this documentation are strictly confidential and the receiver is obliged to use them exclusively for his or her own purposes as defined in the contractual relationship. No part of Viaccess-Orca applications or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Viaccess S.A and/or-Orca Interactive.

The information in this document is subject to change without notice. Viaccess S.A nor Orca Interactive warrant that this document is error free. If you find any problems with this documentation or wish to make comments, please report them to Viaccess-Orca in writing at documentation@viaccess-orca.com.

Trademarks

Viaccess-Orca is a trademark of Viaccess S.A ® in France and/or other countries. All other product and company names mentioned herein are the trademarks of their respective owners.

Viaccess S.A and or Orca Interactive may hold patents, patent applications, trademarks, copyrights or other intellectual property rights over the product described in this document. Unless expressly specified otherwise in a written license agreement, the delivery of this document does not imply the concession of any license over these patents, trademarks, copyrights or other intellectual property.

Table of Contents

1	Introduction	1
1.1	In this Guide	1
2	Key Concepts	2
2.1	Concept 1: The System Components	2
2.2	Concept 2: The Users	2
2.2.1	Accounts vs. Profiles	2
2.2.2	Anonymous vs. Authenticated Users	2
2.2.3	Online Accounts vs. Operator Accounts	3
2.3	Concept 3: Devices	3
2.4	Concept 4: VOD Content Offers	3
2.5	Concept 5: TV Shows	3
2.6	Concept 6: Live Content	4
2.7	Concept 7: Personalization	4
2.8	Concept 8: The VOD Catalog	4
2.9	Concept 9: Assets	6
2.9.1	Metadata	6
2.9.2	Video Files	6
2.9.3	Image Files	7
2.9.4	Movie Attachments	7
3	General Guidelines	8
3.1	Designing for Cache Efficiency	8
4	Using the Tutorials	9
4.1	Flowcharts	9
4.2	Parameter/Property Identifiers	9
4.3	API Identifiers	10
4.4	Connectors	10
5	Development Paths	11
6	VOD Development Paths	12
6.1	TVOD (Rent or Buy): Movies	12
6.1.1	Screen Creation	12
6.1.2	Process Implementation	12
6.2	TVOD (Rent or Buy): TV Shows	13
6.2.1	Screen Creation	13
6.2.2	Process Implementation	13
6.3	SVOD (Subscription VOD)	14
6.3.1	Screen Creation	14



6.3.2 Process Implementation	14
6.4 VOD Movie Packages	15
6.4.1 Screen Creation	15
6.4.2 Process Implementation	15
6.5 Create a Movie Screen	16
6.5.1 Screen Creation	16
6.5.2 Process Implementation	17
7 Live Content Development Path	18
7.1 Screen Creation	18
7.2 Process Implementation	19
8 Service Plan Development Path	22
8.1 Screen Creation	22
8.2 Process Implementation	22
10 Account Management Development Path	24
10.1 Managing Accounts	24
10.1.1 Screen Creation	24
10.1.2 Process Implementation	25
10.2 Managing Users	26
10.2.1 Screen Creation	26
10.2.2 Process Implementation	27
10.3 Managing Devices	28
10.3.1 Screen Creation	28
10.3.2 Process Implementation	28
10.4 Managing Payment Methods	28
10.4.1 Screen Creation	28
10.4.2 Process Implementation	29
10.5 Language Management	29
10.5.1 Screen Creation	29
10.5.2 Process Implementation	30
10.6 Parental Control Management	30
10.6.1 Screen Creation	30
10.6.2 Process Implementation	30
11 Common Screen Development	32
11.0.1 The Home Screen	32
11.0.2 The My Library Screen	32
11.0.3 The My Offers/All Offers Screen	34
12 Common Functionality Development	35
12.1 Search	35
12.2 Display Images	35
12.3 Get List of Recommendations	36



12.4 Registering Events	36
12.5 Sharing Over Social Networks	37
13 Create a Home Screen	38
13.1 Screen Overview	38
13.2 Requests and Responses	39
13.3 Recommendation Requests	40
13.4 Related Tutorials	41
14 Create a TVOD Home Screen	42
14.1 Screen Overview	42
14.2 Related Tutorials	43
15 Get List of TVOD Recommendations	44
15.1 Overview	44
15.2 Requests and Responses	44
15.3 Recommendation Requests	44
15.4 Related Tutorials	45
16 Create a TVOD Movies Catalog Screen	47
16.1 Screen Overview	47
16.2 Related Tutorials	48
17 Get TVOD Movie Categories	49
17.1 Overview	49
17.2 Requests and Responses	49
17.3 Related Tutorials	50
17.4 Example	51
18 Get TVOD Movies	52
18.1 Overview	52
18.2 Requests and Responses	52
18.3 Related Tutorials	53
18.4 Example	54
19 Create a TVOD TV Shows Catalog Screen	56
19.1 Screen Overview	56
19.2 Related Tutorials	57
20 Get VOD TV Show Categories	58
20.1 Overview	58
20.2 Requests and Responses	58
20.3 Related Tutorials	59
20.4 Example	59



21	Get VOD TV Show Series and Seasons in a Category	61
21.1	Overview	61
21.2	Requests and Responses	62
21.3	Related Tutorials	63
21.4	Example	64
22	Get Seasons in a TVOD TV Show Series	66
22.1	Overview	66
22.2	Requests and Responses	67
22.3	Related Tutorials	68
22.4	Example	68
23	Get Episodes in a TVOD Season	71
23.1	Overview	71
23.2	Requests and Responses	72
23.3	Related Tutorials	73
23.4	Example	73
24	Create an SVOD Catalog Screen	75
24.1	Screen Overview	75
24.2	Related Tutorials	77
25	Get List of SVOD Recommendations	78
25.1	Overview	78
25.2	Requests and Responses	78
25.3	Recommendation Requests	78
25.4	Related Tutorials	79
26	Get SVOD Services	80
26.1	Overview	80
26.2	Requests and Responses	80
26.3	Related Tutorials	81
26.4	Example	82
27	Get SVOD Service Contents	83
27.1	Overview	83
27.2	Requests and Responses	83
27.2.1	Responses	84
27.3	Related Tutorials	86
27.4	Example	86
27.4.1	Example - Movie	86
27.4.2	Example - Series	89
	Example: Season	91



28 Identify Subscribed Services	94
28.1 Overview	94
28.2 Requests and Responses	94
28.3 Related Tutorials	95
28.4 Example	95
29 Create a VOD Movie Packages Catalog Screen	96
29.1 Screen Overview	96
29.2 Related Tutorials	97
30 Get List of VOD Movie Package Recommendations	98
30.1 Overview	98
30.2 Requests and Responses	98
30.3 Recommendation Requests	98
30.4 Related Tutorials	99
31 Get VOD Movie Packages	100
31.1 Overview	100
31.2 Requests and Responses	100
31.3 Related Tutorials	101
31.4 Example	101
32 Get VOD Package Contents	103
32.1 Overview	103
32.2 Requests and Responses	103
32.3 Related Tutorials	104
32.4 Example	104
33 Create a Movie Screen	106
33.1 Overview	106
33.2 Related Tutorials	107
34 Get Movie Metadata	108
34.1 Overview	108
34.2 Requests and Responses	109
34.3 Related Tutorials	111
34.4 Example	111
35 List Compatible Devices	113
35.1 Overview	113
35.2 Requests and Responses	113
35.3 Related Tutorials	114
35.4 Example	114



36 List Offers	115
36.1 Overview	115
36.2 Related Tutorials	115
37 List TVOD Offers	117
37.1 Overview	117
37.2 Requests and Responses	117
37.3 Related Tutorials	120
37.4 Example	120
38 List SVOD Offers for VOD Movie	122
38.0.1 Overview	122
38.1 Requests and Responses	122
38.2 Related Tutorials	123
38.3 Example	123
39 List Service Plan Offers for a VOD Movie	125
39.1 Overview	125
39.2 Requests and Responses	126
39.3 Related Tutorials	127
39.4 Example	127
40 List VOD Movie Package Offers	129
40.1 Overview	129
40.2 Requests and Responses	129
40.3 Related Tutorials	130
40.4 Example	131
41 Get Pricing Matrix Info	132
41.1 Overview	132
41.2 Requests and Responses	132
41.3 Related Tutorials	133
41.4 Example	133
42 Get User Ratings	135
42.1 Overview	135
42.2 Requests and Responses	135
42.3 Related Tutorials	136
42.4 Example	136
43 Playing a Trailer	137
43.1 Overview	137
43.2 Requests and Responses	137



43.3 Related Tutorials	138
43.4 Example	138
44 Find a Movie Page	139
44.1 Overview	139
44.2 Requests and Responses	139
44.3 Related Tutorials	140
45 Playing a Movie	141
45.1 Overview	141
45.2 Related Tutorials	141
46 Check if Account Has Rights to Play Movie	143
46.1 Related Tutorials	143
47 Check TVOD and VOD Movie Package Video Tickets	144
47.1 Overview	144
47.2 Requests and Responses	144
47.3 Related Tutorials	145
47.4 Example	145
48 Check SVOD Subscriptions	148
48.1 Overview	148
48.2 Requests and Responses	148
48.2.1 Requests and Responses: GetAggregatedVideo	148
48.2.2 Requests and Responses: GetHouseHoldServiceList	149
48.3 Related Tutorials	149
48.4 Example	149
49 Get Movie Playback URL	151
49.1 Overview	151
49.2 Requests and Responses	151
49.3 Related Tutorials	152
49.4 Example	152
50 Using the Media Player	154
50.1 Related Tutorials	154
51 Mark Last Position	155
51.1 Overview	155
51.2 Requests and Responses	155
51.3 Related Tutorials	156
51.4 Example	156
52 Activate VOD Package Movie Ticket	157



52.1 Overview	157
52.2 Requests and Responses	157
52.3 Related Tutorials	158
52.4 Example	158
53 Manage a Wish List	160
53.1 Overview	160
53.2 Related Tutorials	161
54 Check the Wish List	162
54.1 Overview	162
54.2 Requests and Responses	162
54.3 Related Tutorials	163
54.4 Example	163
55 Add to the Wish List	165
55.1 Requests and Responses	165
55.2 Related Tutorials	166
55.3 Example	166
56 Remove from the Wish List	167
56.1 Requests and Responses	167
56.2 Related Tutorials	167
56.3 Example	168
57 Rate a Movie	169
57.1 Overview	169
57.2 Requests and Responses	169
57.3 Related Tutorials	170
57.4 Example	170
58 Get Recommended Movies	171
58.1 Overview	171
58.2 Requests and Responses	172
58.3 Recommendation Requests – Movie Screen	173
58.4 Related Tutorials	173
59 Create an Episode Screen	174
59.1 Overview	174
59.2 Related Tutorials	174
60 Create a TVOD TV Show Series Screen	175
60.1 Overview	175
60.2 Requests and Responses	175

60.3 Related Tutorials	177
60.4 Example	177
61 Create a VOD Season Screen	179
61.1 Overview	179
61.2 Requests and Responses	180
61.3 Related Tutorials	181
61.4 Example	181
62 Get Recommended TVOD Seasons	184
62.1 Overview	184
62.2 Requests and Responses	184
62.3 Example	186
63 Create a TV Guide	188
63.1 Overview	188
63.2 Basic Display Guidelines	188
63.3 Additional Display Guidelines	189
63.4 Functionality Requirements	189
63.5 Related Tutorials	189
64 Get a List of Channels	190
64.1 Overview	190
64.2 Requests and Responses	190
64.3 Related Tutorials	191
64.4 Performance Guidelines	191
64.5 Example	191
65 Get List of Channel Programs	193
65.1 Overview	193
65.2 Requests and Responses	193
65.3 Related Tutorials	194
65.4 Performance Guidelines	195
65.5 Example	195
66 Identify Subscribed Channels	197
66.1 Overview	197
66.2 Related Tutorials	197
67 Get Account Channel Packages	198
67.1 Overview	198
67.2 Requests and Responses	198
67.3 Related Tutorials	199

67.4 Example	199
68 Get Channels in Package	200
68.1 Overview	200
68.2 Requests and Responses	200
68.3 Related Tutorials	201
68.4 Example	201
69 Create a Program Screen	202
69.1 Overview	202
69.2 Basic Display Guidelines	202
69.3 Functionality Requirements	203
69.4 Related Tutorials	204
70 Get Program Metadata	205
70.1 Overview	205
70.2 Requests and Responses	205
70.3 Related Tutorials	206
70.4 Example	207
71 Get Channel Info from Internal ID	208
71.1 Overview	208
71.2 Requests and Responses	208
71.3 Related Tutorials	209
71.4 Example	209
72 List Offers for a Channel	211
72.1 Requests and Responses	212
72.2 Related Tutorials	214
72.3 Example	214
73 Get Compatible Devices	216
73.1 Overview	216
73.2 Requests and Responses	216
73.3 Related Tutorials	217
73.4 Example	217
74 Setting Program Reminders	218
74.1 Overview	218
74.2 Related Tutorials	218
75 Get Recommended Live Content	219
75.1 Overview	219
75.2 Requests and Responses	219

75.3 Recommendation Calls	219
75.4 Related Tutorials	219
76 Playing Live Programs	220
76.1 Overview	220
76.2 Related Tutorials	220
77 Play Unencrypted Channel	221
77.1 Overview	221
77.2 Requests and Responses	221
77.3 Related Tutorials	222
77.4 Example	222
78 Playing Encrypted Channels	223
78.1 Overview	223
78.2 Requests and Responses	223
78.3 Related Tutorials	224
78.4 Example	224
79 Create a Now Playing Screen	225
79.1 Overview	225
79.2 Implementation	225
79.3 Related Tutorials	226
80 Create a Login Screen	227
80.1 Overview	227
80.2 Related Tutorials	227
81 Create an Account	229
81.1 Overview	229
81.2 Related Tutorials	230
82 Resend Verification Email	231
82.1 Overview	231
82.2 Requests and Responses	232
82.3 Related Tutorials	232
83 Get CAPTCHA Key	233
83.1 Overview	233
83.2 Requests and Responses	233
83.3 Related Tutorials	233
83.4 Example	234
84 Get CAPTCHA Image	235
84.1 Overview	235

84.2 Requests and Responses	235
84.3 Related Tutorials	235
84.4 Example	235
85 Submit User Details and Capture Challenge	237
85.1 Overview	237
85.2 Requests and Responses	237
85.3 Related Tutorials	238
85.4 Example	239
86 Logging In and Out	240
86.1 Overview	240
86.2 Related Tutorials	241
87 Log into RiGHTv Account	243
87.1 Overview	243
87.2 Requests and Responses	243
87.3 Related Tutorials	244
87.4 Example	244
88 Log into COMPASS Account	246
88.1 Overview	246
88.2 Requests and Responses	246
88.3 Related Tutorials	246
88.4 Example	247
89 Get User List	248
89.1 Overview	248
89.2 Requests and Responses	248
89.3 Related Tutorials	249
89.4 Example	250
90 Log in the User	251
90.1 Overview	251
90.2 Requests and Responses	251
90.3 Related Tutorials	252
90.4 Example	252
91 Log Out of RiGHTv	253
91.1 Overview	253
91.2 Requests and Responses	253
91.3 Related Tutorials	253
91.4 Example	254

92 Log Out of COMPASS	255
92.1 Overview	255
92.2 Requests and Responses	255
92.3 Related Tutorials	255
92.4 Example	255
93 Creating and Updating Users	257
93.1 Overview	257
93.2 Related Tutorials	258
94 Create a New User	259
94.1 Overview	259
94.2 Requests and Responses	259
94.3 Related Tutorials	260
94.4 Example	260
95 Update a User's Name	262
95.1 Overview	262
95.2 Requests and Responses	262
95.3 Related Tutorials	263
95.4 Example	263
96 Building a My Account Screen	264
96.1 Overview	264
96.2 Requests and Responses	264
97 Get Account Information	267
97.1 Overview	267
97.2 Requests and Responses	267
97.3 Example	267
98 Managing User Images	269
98.1 Overview	269
98.2 Related Tutorials	269
99 Get a User's Image	270
99.1 Overview	270
99.2 Requests and Responses	270
99.3 Related Tutorials	270
99.4 Example	270
100 Create a User Image	272
100.1 Overview	272
100.2 Requests and Responses	272

100.3 Related Tutorials	272
100.4 Example	272
101 Update a User Image	274
101.1 Overview	274
101.2 Requests and Responses	274
101.3 Related Tutorials	274
101.4 Example	274
102 Delete a Profile Image	276
102.1 Overview	276
102.2 Requests and Responses	276
102.3 Related Tutorials	276
102.4 Example	276
103 Managing User Usage Data Opt-In/out Status	278
103.1 Overview	278
103.2 Related Tutorials	278
104 Get the Current Opt-in/out Status	279
104.1 Overview	279
104.2 Requests and Responses	279
104.3 Related Tutorials	279
104.4 Example	280
105 Opt-in and Opt-out	281
105.1 Overview	281
105.2 Requests and Responses	281
105.3 Related Tutorials	281
105.4 Example	281
106 Updating and Resetting Account Passwords	283
106.1 Overview	283
106.2 Related Tutorials	283
107 Reset an Account's Password	284
107.1 Overview	284
107.2 Requests and Responses	284
107.3 Related Tutorials	286
107.4 Example	286
108 Update an Account's Password	288
108.1 Overview	288
108.2 Requests and Responses	288



108.3 Related Tutorials	289
108.4 Example	289
109 Purchase PIN Management	291
109.1 Overview	291
109.2 Related Tutorials	291
110 Get Purchase PIN	292
110.1 Overview	292
110.2 Requests and Responses	292
110.3 Related Tutorials	293
110.4 Example	293
111 Reset a User's PIN	294
111.1 Overview	294
111.2 Requests and Responses	294
111.3 Related Tutorials	295
111.4 Example	295
112 Update a User's PIN	296
112.1 Overview	296
112.2 Requests and Responses	296
112.3 Related Tutorials	297
112.4 Example	297
113 Device Management	298
113.1 Overview	298
113.2 Related Tutorials	298
114 Register a New Device	299
114.1 Requests and Responses	299
114.2 Related Tutorials	300
114.3 Example	300
115 Get Device List	301
115.1 Overview	301
115.2 Requests and Responses	301
115.3 Related Tutorials	302
115.4 Example	302
116 Remove Device	304
116.1 Overview	304
116.2 Requests and Responses	304
116.3 Related Tutorials	305

116.4 Example	305
117 Update Device Name	306
117.1 Overview	306
117.2 Requests and Responses	306
117.3 Related Tutorials	307
117.4 Example	307
118 Identify Account Type	308
118.1 Overview	308
118.2 Requests and Responses	308
118.3 Related Tutorials	308
118.4 Example	309
119 The Purchase Process	312
119.1 Overview	312
119.2 Online Account Purchase Process	312
119.3 Traditional Account Purchase Process	315
120 Managing Payment Methods	318
120.1 Overview	318
120.2 Related Tutorials	318
121 Register a New Payment Method via Adyen	320
121.1 Overview	320
121.2 Related Tutorials	320
122 Get Payment Registration Properties	321
122.1 Overview	321
122.2 Requests and Responses	321
122.3 Related Tutorials	322
122.4 Example	322
123 Register Payment Method	324
123.1 Overview	324
123.2 Example	324
123.3 Related Tutorials	325
124 Complete Registration	327
124.1 Overview	327
124.2 Requests and Responses	327
124.3 Related Tutorials	327
124.4 Example	328

125 Register a New Payment Method via the Agnostic Payment Service	329
125.1 Overview	329
125.2 Requests and Responses	329
125.3 Related Tutorials	330
126 Get an Online Account's Payment Method	331
126.1 Overview	331
126.2 Requests and Responses	331
126.3 Related Tutorials	332
126.4 Example	332
127 Delete Payment Method	334
127.1 Overview	334
127.2 Requests and Responses	334
127.3 Related Tutorials	334
127.4 Example	334
128 Ordering TVOD Content	336
128.1 Overview	336
128.2 Requests and Responses	336
128.3 Related Tutorials	338
128.4 Example	338
129 Ordering a VOD Movie Package	341
129.1 Overview	341
129.2 Requests and Responses	342
129.3 Related Tutorials	343
129.4 Example	344
130 Subscribing to an SVOD Service	345
130.1 Overview	345
130.2 Requests and Responses	345
130.3 Related Tutorials	347
130.4 Example	347
131 Subscribe to a Service Plan	348
131.1 Overview	348
131.2 Requests and Responses	348
131.3 Related Tutorials	349
131.4 Example	350
132 Unsubscribing from a Service Plan	351

132.1 Related Tutorials	351
133 Finish Service Plan Subscription	353
133.1 Overview	353
133.2 Requests and Responses	353
133.3 Related Tutorials	354
133.4 Example	354
134 Remove Service Plan Subscription	355
134.1 Overview	355
134.2 Requests and Responses	355
134.3 Related Tutorials	356
134.4 Example	356
135 Opening a Ticket for an SVOD Movie	357
135.1 Overview	357
135.2 Requests and Responses	357
135.3 Related Tutorials	358
135.4 Example	358
136 Check SVOD Video Tickets	359
136.1 Overview	359
136.2 Requests and Responses	359
136.3 Related Tutorials	360
136.4 Example	360
137 Unsubscribing from an SVOD Service	362
137.1 Related Tutorials	362
138 Finish Service Subscription	364
138.1 Overview	364
138.2 Requests and Responses	364
138.3 Related Tutorials	364
138.4 Example	365
139 Remove Service Subscription	366
139.1 Overview	366
139.2 Requests and Responses	366
139.3 Related Tutorials	366
139.4 Example	367
140 Subscribe to a Channel Package	368
140.1 Overview	368
140.2 Requests and Responses	368



140.3 Related Tutorials	370
140.4 Example	370
141 Unsubscribe from a Channel Package	371
141.1 Overview	371
141.2 Related Tutorials	372
142 Finish Channel Package Subscription	373
142.1 Overview	373
142.2 Requests and Responses	373
142.3 Related Tutorials	374
142.4 Example	374
143 Remove Channel Package Subscription	375
143.1 Overview	375
143.2 Requests and Responses	375
143.3 Related Tutorials	376
143.4 Example	376
144 Performing a Search	377
144.1 Overview	377
144.2 Related Tutorials	377
145 Perform a General Search	378
145.1 Overview	378
145.2 Requests and Responses	378
145.3 Related Tutorials	382
145.4 Example	383
146 Performing a Live Programs Search	384
146.1 Overview	384
146.2 Requests and Responses	384
146.3 Related Tutorials	388
146.4 Example	388
147 Performing a Video Search	390
147.1 Overview	390
147.2 Requests and Responses	391
147.3 Related Tutorials	395
147.4 Example	395
148 Adding Auto-suggestion to Search	397
148.1 Overview	397
148.2 Requests and Responses	397

148.3 Additional Information	399
148.4 Related Tutorials	400
148.5 Example	400
149 Display Images	402
149.1 Overview	402
149.2 Requests and Responses	403
149.3 Retrieving Images	404
149.4 On the Fly Resizing	404
149.5 Example 1: From RiGHTv	404
149.6 Example 2: From COMPASS	405
149.7 Example 3: Live Channel Image	405
150 Get List of Recommendations	406
150.1 Overview	406
150.2 Requests and Responses – VOD Content	406
150.3 Example	408
150.4 Recommendations for Live Content	410
150.5 Example	411
151 Creating a My Library Screen	413
151.1 Overview	413
151.2 Related Tutorials	413
152 List My Movies	415
152.1 Overview	415
152.2 Requests and Responses	415
152.3 Related Tutorials	416
152.4 Example	416
153 List My VOD Package Movies	418
153.1 Overview	418
153.2 Requests and Responses	418
153.3 Related Tutorials	419
153.4 Example	419
154 List My SVOD Subscriptions	420
154.1 Overview	420
154.2 Requests and Responses	420
154.3 Requests and Responses	421
154.4 Related Tutorials	421
154.5 Example	421

155 List My Service Plans	423
155.1 Overview	423
155.2 Requests and Responses	423
155.3 Related Tutorials	425
155.4 Example	425
156 Show My Wishlist	426
156.1 Requests and Responses	426
156.2 Related Tutorials	426
156.3 Example	427
157 Show My Reminders	428
157.1 Overview	428
157.2 Related Tutorials	428
158 My Offers / All Offers Screen	429
158.1 Overview	429
158.2 Related Tutorials	430
159 List All Available SVOD Offers	431
159.1 Overview	431
159.2 Requests and Responses	432
159.2.1 Requests and Responses: GetSubscriptionOffers	432
159.2.2 Requests and Responses: GetHouseholdSubscriptions	433
159.3 Related Tutorials	435
159.4 Example	436
160 List Channel Package Offers	438
160.1 Requests and Responses	438
160.1.1 Requests and Responses: GetSubscriptionOffers	439
160.1.2 Requests and Responses: GetHouseholdSubscriptions	440
160.2 Related Tutorials	442
160.3 Example	442
161 List All Available Service Plan Offers	444
161.1 Overview	444
161.2 Requests and Responses	445
161.2.1 GetSubscriptionOffers	445
161.2.2 GetHouseholdSubscriptions	450
161.3 Related Tutorials	451
161.4 Example	451
162 Implementing Start Over	453
162.1 Overview	453



163 Identify Start Over Enabled Programs	454
163.1 Overview	454
163.2 Requests and Responses: GetExtendedProgram	455
163.3 Requests and Responses: GetProgramList	456
163.4 Related Tutorials	457
163.5 Example	457
164 Playing a Live Channel through Start Over	458
164.1 Overview	458
164.2 Requests and Responses	458
164.3 Related Tutorials	460
164.4 Example	460
165 Guidelines for Start Over Player Controls	462
165.1 Overview	462
165.2 Related Tutorials	463
166 Implementing Catch-Up	464
166.1 Overview	464
167 List Catch-Up Offers	465
167.1 Overview	465
167.2 Requests and Responses	465
167.2.1 GetSubscriptionOffers	466
167.2.2 GetHouseholdSubscriptions	467
167.3 Related Tutorials	469
167.4 Example	469
168 Subscribe to a Catch-Up Service	471
168.1 Overview	471
168.2 Requests and Responses	471
168.3 Related Tutorials	472
168.4 Example	472
169 Unsubscribing from a Catch-Up Service	474
169.1 Related Tutorials	474
170 Finish Catch-Up Subscription	475
170.1 Overview	475
170.2 Requests and Responses	475
170.3 Related Tutorials	476
170.4 Example	476
171 Remove Catch-Up Subscription	477

171.1 Overview	477
171.2 Requests and Responses	477
171.3 Related Tutorials	478
171.4 Example	478
172 List My Catch-Up Services	479
172.1 Overview	479
172.2 Requests and Responses	479
172.3 Related Tutorials	482
172.4 Example	482
173 Identify Catch-Up Enabled Programs	483
173.1 Overview	483
173.2 Requests and Responses: GetExtendedProgram	483
173.3 Requests and Responses: GetProgramList	484
173.4 Related Tutorials	485
173.5 Example	486
174 Determining the Catch-Up Availability Period	487
174.1 Overview	487
174.2 Requests and Responses	487
174.3 Related Tutorials	488
174.4 Example	488
175 Checking if Account is Subscribed to Catch-Up	490
175.1 Overview	490
175.2 Requests and Responses	490
175.3 Related Tutorials	490
176 Playing a Program through Catch-Up	492
176.1 Overview	492
176.2 Requests and Responses	493
176.3 Related Tutorials	494
176.4 Example	495
177 Marking Last Watched Position	496
177.1 Overview	496
177.2 Requests and Responses	496
177.3 Related Tutorials	497
177.4 Example	497
178 Guidelines for Catch-Up Player Controls	498
178.1 Overview	498



178.2 Related Tutorials	499
179 Working with Languages	500
179.1 Overview	500
179.2 Related Tutorials	500
180 Get a List of Supported Languages	501
180.1 Overview	501
180.2 Requests and Responses	501
180.3 Related Tutorials	501
180.4 Example	501
181 Set the Language for a New Account	503
181.1 Overview	503
181.2 Related Tutorials	503
181.3 Example	503
182 Set the Language for an Existing Account	504
182.1 Overview	504
182.2 Requests and Responses	504
182.3 Related Tutorials	504
182.4 Example	504
183 Get the Account's Current Language	506
183.1 Overview	506
183.2 Requests and Responses	506
183.3 Related Tutorials	506
183.4 Example	507
184 Specifying the Response Language	508
184.1 Overview	508
184.2 Requests and Responses	508
184.3 Related Tutorials	508
184.4 Example	508
185 Parental Control	509
185.1 Overview	509
185.2 Related Tutorials	509
186 Verify the Parental Pin	511
186.1 Overview	511
186.2 Requests and Responses	511
186.3 Related Tutorials	511
186.4 Example	511



187 Determine Parental Level of an Entity	513
187.1 Overview	513
187.2 Related Tutorials	513
188 Changing Parental Pins	514
188.1 Overview	514
188.2 Related Tutorials	514
189 Updating the Parental Pin	515
189.1 Overview	515
189.2 Requests and Responses	515
189.3 Related Tutorials	515
189.4 Example	515
190 Resetting the Parental Pin	517
190.1 Overview	517
190.2 Requests and Responses	517
190.3 Related Tutorials	517
190.4 Example	517
191 Get a List of Parental Ratings Used	519
191.1 Overview	519
191.2 Requests and Responses	519
191.3 Related Tutorials	519
191.4 Example	519
192 Set the Parental Control Threshold	521
192.1 Overview	521
192.2 Related Tutorials	521
193 Exclude Adult Content from Returned Lists	522
193.1 Overview	522
193.2 Requests and Responses	522
193.3 Related Tutorials	522
193.4 Example	523
194 Sharing Via Facebook	524
194.1 App Name	525
194.2 Content Image	525
194.3 Content Title	525
194.4 Post Message	526
194.5 Service Name	526
195 Registering Events for Content Discovery	527



195.1 Why Content Discovery?	527
195.2 The Role of the Apps	527
195.3 An Example	527
195.4 Registered Events	528
196 Register Event: Order Video	529
196.1 Overview	529
196.2 Requests and Responses	530
196.3 Related Tutorials	531
196.4 Example	531
197 Register Event: Preview Trailer	532
197.1 Overview	532
197.2 Requests and Responses	532
197.3 Related Tutorials	533
197.4 Example	533
198 Register Event: Content End	534
198.1 Overview	534
198.2 Requests and Responses	534
198.3 Related Tutorials	535
198.4 Example	535
199 Register Event: Rating	536
199.1 Overview	536
199.2 Requests and Responses	536
199.3 Related Tutorials	536
199.4 Example	536
200 Register Event: Set Reminder	538
200.1 Overview	538
200.2 Requests and Responses	538
200.3 Related Tutorials	539
200.4 Example	539
201 Register Event: Channel Zap	540
201.1 Overview	540
201.2 Requests and Responses	540
201.3 Related Tutorials	541
201.4 Example	541
202 Register Search Event	542
202.1 Overview	542

202.2 Requests and Responses	542
202.3 Related Tutorials	543
202.4 Example	543
203 Register Click Event	545
203.1 Overview	545
203.2 Requests and Responses	545
203.3 Example	546
203.4 Related Tutorials	546
203 Register a Banner Click Event	547
203.5 Overview	547
203.6 RegisterEvent Parameter Values	547
204 Providing Support for Chrome v59 Onward	549
204.1 Procedure	549
205 Apps and Analytics	550
205.1 Why Analytics?	550
205.2 The Role of the Apps	550
205.3 Registered Events	551
205.3.1 Ordering a Video	551
205.3.2 Ordering a VOD Package	552
205.3.3 Watching Videos	553
205.3.4 Click Events	555
205.3.5 Searching Content	557
205.4 Additional Guidelines	558
205.4.1 TV Show Implementation	558

1 Introduction

This guide aims to streamline the client application development process. The guide explains the use of the APIs and demonstrates best practices in application design.

The guide presents essential Pay TV concepts to provide the developer with the terminology required to advance through the rest of the guide.

A series of tutorials guides the developer through the process of developing all essential aspects of a functional client application. Example API calls are shown throughout.

1.1 In this Guide

This guide includes the following chapters:

Chapter	Description
2 Key Concepts (p.2)	Explains basic concepts behind a Pay TV service.
4 Using the Tutorials (p.9)	Explains how to use the tutorials in this guide.
5 Development Paths (p.11)	Provides a directory to all tutorials in this guide, sorted by functional area.

2 Key Concepts

This section explains basic concepts behind a Pay TV service.

2.1 Concept 1: The System Components

The TVE (TV Everywhere) solution combines two main components:

- RiGHTv

The Service Delivery Platform, responsible for managing content, subscribers, business models, etc. It provides a powerful selection of pre-integrated end-to-end solutions to help service providers deploy fast, protect content and deliver compelling services.

- COMPASS

The Content Discovery Platform, responsible for providing users with personalized viewing recommendations. With COMPASS, subscribers get personalized recommendations enabling them to find desirable content quickly and easily.

A separate API set is provided for each component, and each is documented in its own guide.

Throughout this resource, the combined solution is referred to as the Service Delivery Platform. Only where necessary are the two components referred to separately. For all API calls mentioned, the relevant API set (RiGHTv or COMPASS) is indicated.

2.2 Concept 2: The Users

When using a client application, the functionality and permissions available vary depending on the nature of the user, as discussed below.

2.2.1 Accounts vs. Profiles

The basic customer unit within the Service Delivery Platform is called an Account.

A number of individual Profiles can be defined within each Account.

2.2.2 Anonymous vs. Authenticated Users

Client applications can be set up to allow the user to browse content without first being authenticated by a log in process:

Both anonymous and authenticated users can:

- Browse available content
- View content metadata
- Watch trailers

Authenticated users can additionally:

- Purchase content
- View purchased content

2.2.3 Online Accounts vs. Operator Accounts

Two types of accounts can be supported by the client application:

- Operator accounts

These accounts are debited using the operator's billing systems. They typically pay monthly.

Customer service is provided to these accounts primarily by the operator's Customer Service department.

- Online accounts

Anyone who opens an account through a providers' app or website opens an online account. With an application by credit card, without the need for a Customer Service department.

2.3 Concept 3: Devices

Using VO's API set, client applications can be developed for multiple types of devices, such as STB's (Set-top boxes), PCs, smartphones, and tablets. Devices are also referred to as terminals, for example in the APIs.

2.4 Concept 4: VOD Content Offers

Users can buy VOD content through different types of offers:

- Transaction VOD (TVOD)

The user buys access to a single content item (i.e. movie or TV show episode). A user can buy TVOD offers with the following delivery options:

- Rental

The customer receives time-limited access to the content.

- eST (electronic sell-through)

The customer purchases VOD content for an unlimited period of time.

- Movie Package

The user buys access to a predefined group of movies. As for TVOD, both Rental and eST delivery options are available.

- SVOD

The user pays a monthly fee for a service that provides access to a set of movies that can change over time.

2.5 Concept 5: TV Shows

A TV Show series may consist of a number of seasons, each consisting of a number of individual episodes.

Any of the available VOD content offer types can be applied to TV shows. See [2.4 Concept 4: VOD Content Offers \(p.3\)](#).



2.6 Concept 6: Live Content

Users may watch live programs broadcast on different channels. Users choose channels to watch either from a TV Guide screen, or from a Now Playing screen that plays the selected channel and allows the user to flip from one channel to another.

Operators group channels together into channel packages which they then sell as services to customers. A user can subscribe to one or more channel packages. A single channel can be included in multiple channel packages.

2.7 Concept 7: Personalization

The Service Delivery Platform is able to provide content recommendations to users.

Recommendations are generated based on:

- User specific parameters, such as the viewer's personal viewing preferences and ratings.
- Non-user specific parameters, such as external viewer ratings, popularity, most recent content, promotions and related content.

The relevant APIs allow recommendation lists to be customized and pre-filtered, so that different context-relevant recommendation lists can be displayed on each screen of a client application.

The operator can additionally define Blends, which are preconfigured combinations of recommendations.

2.8 Concept 8: The VOD Catalog

Users browse through the operator's VOD Catalog to find content to watch or buy. A typical VOD Catalog structure is shown below.

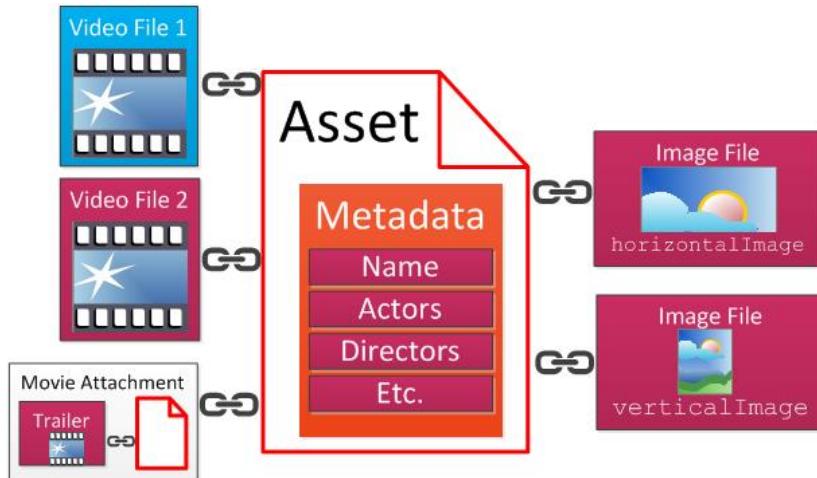
Note: Folders within the catalog are frequently referred to as *Categories* throughout this guide.

<p>Main Folders</p> <p>Each VOD offer type (TVOD, SVOD, and VOD Packages) is assigned its own main folder in the catalog.</p>	<ul style="list-style-type: none"> ▶ TVOD Services ▶ SVOD Services ▶ VOD Packages 									
<p>TVOD Folders</p> <p>The TVOD folder is divided between Movies and TV Shows</p>	<ul style="list-style-type: none"> ▼ TVOD Services <ul style="list-style-type: none"> ▶ Movies ▶ TV Shows 									
<p>TVOD/Movie Folders</p> <p>The TVOD/Movies folder is subdivided into folders by Genre, with each folder including a number of movies that match the genre.</p>	<ul style="list-style-type: none"> ▼ TVOD Services <ul style="list-style-type: none"> ▼ Movies <ul style="list-style-type: none"> ▶ All ■ Action ▶ Science Fiction ▶ Comedy ▶ Drama ▶ Foreign ▶ Horror ▶ Kids ▶ Mystery <div data-bbox="873 774 1286 1044"> <p>Action</p> <p>11 Results 0 selected</p> <table border="1"> <thead> <tr> <th>Title</th> </tr> </thead> <tbody> <tr> <td> Mad Max: Fury Road</td> </tr> <tr> <td> The Last Witch Hunter</td> </tr> <tr> <td> Deadpool</td> </tr> <tr> <td> Avatar</td> </tr> <tr> <td> Avengers: Age of Ultron</td> </tr> <tr> <td> Cinderella</td> </tr> <tr> <td> Django Unchained</td> </tr> <tr> <td> Furious 7</td> </tr> </tbody> </table> </div>	Title	Mad Max: Fury Road	The Last Witch Hunter	Deadpool	Avatar	Avengers: Age of Ultron	Cinderella	Django Unchained	Furious 7
Title										
Mad Max: Fury Road										
The Last Witch Hunter										
Deadpool										
Avatar										
Avengers: Age of Ultron										
Cinderella										
Django Unchained										
Furious 7										
<p>TVOD/TV Shows Folders</p> <p>The TVOD/TV Shows folder is subdivided into folders by Genre, with each folder including a number of series that match the genre.</p>	<ul style="list-style-type: none"> ▶ Movies ▼ TV Shows <ul style="list-style-type: none"> ▶ All ■ Drama ▶ Comedy ▶ Foreign <div data-bbox="873 1190 1222 1482"> <p>Drama</p> <p>6 Results 0 selected</p> <table border="1"> <thead> <tr> <th>Title</th> </tr> </thead> <tbody> <tr> <td> Weeds</td> </tr> <tr> <td>■ Sherlock</td> </tr> <tr> <td> Grey's Anatomy</td> </tr> <tr> <td> The Good Wife</td> </tr> </tbody> </table> </div>	Title	Weeds	■ Sherlock	Grey's Anatomy	The Good Wife				
Title										
Weeds										
■ Sherlock										
Grey's Anatomy										
The Good Wife										
<p>TVOD/TV Shows/Series Folders</p> <p>A series folder includes subfolders for each of its seasons. Each season subfolder includes its episodes.</p>	<ul style="list-style-type: none"> ▼ Sherlock <ul style="list-style-type: none"> ▶ Sherlock Season 3 ▶ Sherlock Season 4 ▶ House of cards ▶ The big bang theory <div data-bbox="873 1594 1445 1864"> <p>Sherlock Season 3</p> <p>2 Results 0 selected</p> <table border="1"> <thead> <tr> <th>Title</th> </tr> </thead> <tbody> <tr> <td>■ Sherlock S3EP01</td> </tr> <tr> <td> Sherlock S3EP02</td> </tr> </tbody> </table> </div>	Title	■ Sherlock S3EP01	Sherlock S3EP02						
Title										
■ Sherlock S3EP01										
Sherlock S3EP02										

<p>SVOD Folders</p> <p>Each SVOD folder contains all movies that are part of that SVOD service. They can also include folders that hold season episodes.</p>	<p>SVOD Services</p> <ul style="list-style-type: none"> ▶ Oscar ▶ Blockbusters ▶ HBO ▶ Fox ▶ Disney 	<p>Disney</p> <p>12 Results 0 selected</p> <table border="1"> <thead> <tr> <th>Title</th> </tr> </thead> <tbody> <tr> <td>Austin & Ally</td> </tr> <tr> <td>Phineas and Ferb</td> </tr> <tr> <td>Jessie</td> </tr> <tr> <td>Mickey Mouse Clubhouse</td> </tr> <tr> <td>Maleficent</td> </tr> <tr> <td>Brave</td> </tr> </tbody> </table>	Title	Austin & Ally	Phineas and Ferb	Jessie	Mickey Mouse Clubhouse	Maleficent	Brave
Title									
Austin & Ally									
Phineas and Ferb									
Jessie									
Mickey Mouse Clubhouse									
Maleficent									
Brave									

2.9 Concept 9: Assets

The graphic below illustrates the concept of an asset.



An asset represents a content item in the system, such as a movie or TV show episode. The four main components of an asset are its metadata, movie files, image (poster) files, and attachment movies.

2.9.1 Metadata

Assets include metadata such as the movie name, a description, actor names, details of awards won, etc. The operator can also define additional fields.

This textual information can easily be retrieved and displayed in the front end application.

2.9.2 Video Files

Each asset may include one or more video files – the physical video files that are eventually played through the front end application. Multiple video files may be provided to support different devices, levels of video quality, etc.

2.9.3 Image Files

A number of attachment image types can be defined for each entity within the platform. These types can be thought of as placeholders that can contain images. For assets, *verticalImage* and *horizontalImage* image attachment types are by default defined for assets.

These images can be retrieved and displayed within the front end application, for example, to be used as thumbnails in catalog screens.

The platform supports on-the-fly resizing, so that one uploaded image can be displayed at any required size.

2.9.4 Movie Attachments

A number of attachment movies can be defined for each entity within the platform. These types can be thought of as placeholders that can contain movies. For assets, trailer movie attachment types are by default defined for assets.

Unlike for images, attachment movies are not uploaded and attached to an asset directly – rather a separate asset is created for each attachment movie, and that asset is then attached to the main asset.

3 General Guidelines

This topic presents important guidelines to be followed throughout the client application development process.

3.1 Designing for Cache Efficiency

Much of the data requested by client applications on a regular basis can be classified as public data - data that is common to all users and is not specific to any one user. Examples include:

- Lists of genres
- Lists of live channels
- Lists of upcoming programs on a live channel, etc.

The RiGHTv solution includes caching of public data. Whenever a client application requests public data, the cache is surveyed to see if the same data has been requested previously - if it has, the data is returned to the client application directly from the cache, without polling the RiGHTv server itself at all. Making use of the cache is essential for ensuring server efficiency and performance.

Requests are identified and compared by URL only. Data will only be returned from the cache if the identical URL was previously received. So, for example, if the following two requests are received from two client applications:

- `https://<operator>.tvaas.com/RTEFacade/GetProgramLists?channel_external_ids=France24Ar&start_date=1517443200&end_date=1517486400&client=json`
- `https://<operator>.tvaas.com/RTEFacade/GetProgramLists?channel_external_ids=France24Ar&start_date=1517443201&end_date=1517443201&client=json`

since the start and end dates differ, even though only slightly, the two calls are judged to be different and the RiGHTv server will be polled when the second request is received.

The following guidelines should thus be followed:

- All the Operator's client applications (e.g. Android, iOS, and web) should use identical API requests for all operations that request public data.
- API requests that include start and end times should standardize those times. For example, when requesting live program data:
 - Don't set the Start Time to <now> and the end time to <now + 4 hours>
 - Rather set the Start Time to <beginning of current day> and the end time to <end of current day>, or to the start and end of the current day section, for example 0:00 to 12:00 and 12:00 to 24:00.



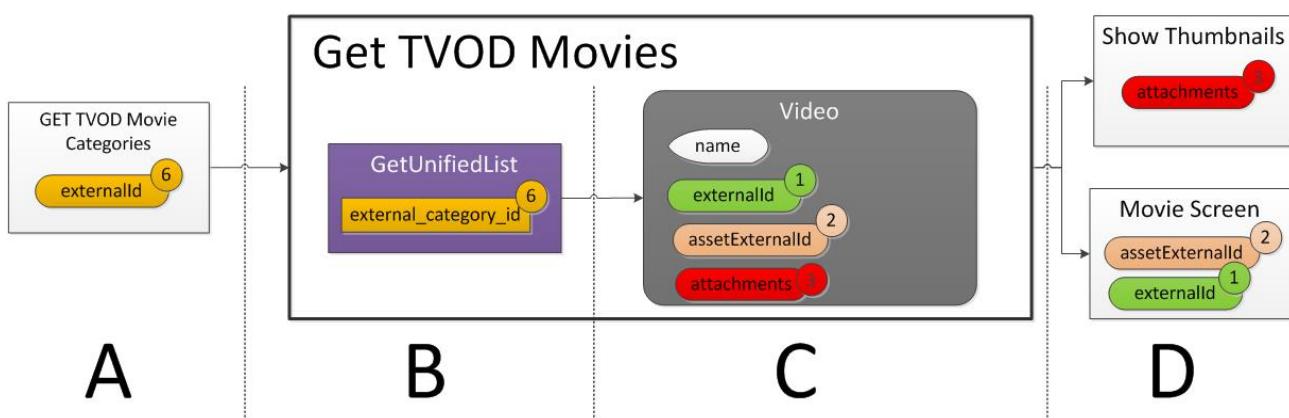
4 Using the Tutorials

This Guide presents a number of tutorials to guide the developer through the process of creating a Pay TV client application using Viaccess-Orca's rich API set.

The full list of tutorials can be found in [5 Development Paths \(p.11\)](#)

4.1 Flowcharts

The inputs to many API requests are obtained from the outputs of others. To clarify the workflow, flowcharts such as the one below are often presented.



- Each flowchart is divided into four sections as indicated above:
- A lists the tutorial from which the input to the current tutorial is obtained. The name of the output property obtained is shown within the block.
- B and C present the Request-Response pair discussed in the tutorial:
 - B shows the API request used in the current tutorial. The main parameters sent within the request are listed within the block.
 - C shows the response to the API request with the main properties returned listed.
- D lists the tutorials in which the output properties from the current tutorial are used. The properties used in each tutorial are listed within the blocks.

4.2 Parameter/Property Identifiers

Many of the API parameters/properties are used frequently, though the names used may vary from one API request to another. To help identify them, color and number/letter codes have been assigned to many parameters/properties, as listed below.



Where a returned property is useful for immediate display on the screen, the shape of the property block is shown as .

Where a parameter used in by an API request refers to information the user has to enter into the app, the shape of the parameter is shown as .

4.3 API Identifiers

API functions from both the RiGHTv and COMPASS APIs are used. Each function block is colored accordingly. Blocks representing internal logic required within the client application are also differentiated by color.



4.4 Connectors

The following connectors are used:

- Typical Flow
- Flow in specific circumstances
- Flow indicating verification success required

5 Development Paths

This section groups the tutorials presented in this guide under the following separate development paths:

- [6 VOD Development Paths \(p.12\)](#)
- [7 Live Content Development Path \(p.18\)](#)
- [8 Service Plan Development Path \(p.22\)](#)
- [10 Account Management Development Path \(p.24\)](#)
- [\(p.1\)](#)
- [11 Common Screen Development \(p.32\)](#)
- [12 Common Functionality Development \(p.35\)](#)

6 VOD Development Paths

The tutorials listed in this section explain how to create movie catalogs for different types of VOD offers, how to construct a Movie Screen, and how to process user purchases of content.

The following paths are described:

- [6.1 TVOD \(Rent or Buy\): Movies \(p.12\)](#)
- [6.2 TVOD \(Rent or Buy\): TV Shows \(p.13\)](#)
- [6.3 SVOD \(Subscription VOD\) \(p.14\)](#)
- [6.4 VOD Movie Packages \(p.15\)](#)
- [6.5 Create a Movie Screen \(p.16\)](#)

6.1 TVOD (Rent or Buy): Movies

6.1.1 Screen Creation

Screen Creation	Sub-components	Sub-components
14 Create a TVOD Home Screen (p.42).	15 Get List of TVOD Recommendations (p.44).	
16 Create a TVOD Movies Catalog Screen (p.47).	17 Get TVOD Movie Categories (p.49).	
	18 Get TVOD Movies (p.52).	
33 Create a Movie Screen (p.106).	152 List My Movies (p.415)	
151 Creating a My Library Screen (p.413).		

6.1.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
128 Ordering TVOD Content (p.336).		

6.2 TVOD (Rent or Buy): TV Shows

6.2.1 Screen Creation

Screen Creation	Sub-components	Sub-components
14 Create a TVOD Home Screen (p.42).	15 Get List of TVOD Recommendations (p.44).	
19 Create a TVOD TV Shows Catalog Screen (p.56).	20 Get VOD TV Show Categories (p.58).	
	21 Get VOD TV Show Series and Seasons in a Category (p.61).	
59 Create an Episode Screen (p.174).		
61 Create a VOD Season Screen (p.179).		
60 Create a TVOD TV Show Series Screen (p.175).	62 Get Recommended TVOD Seasons (p.184).	
151 Creating a My Library Screen (p.413)	152 List My Movies (p.415)	

6.2.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
128 Ordering TVOD Content (p.336).		



6.3 SVOD (Subscription VOD)

6.3.1 Screen Creation

Screen Creation	Sub-components	Sub-components
24 Create an SVOD Catalog Screen (p.75).	25 Get List of SVOD Recommendations (p.78)	
	26 Get SVOD Services (p.80).	
	27 Get SVOD Service Contents (p.83).	
33 Create a Movie Screen (p.106).		
151 Creating a My Library Screen (p.413).	154 List My SVOD Subscriptions (p.420).	
158 My Offers / All Offers Screen (p.429).	159 List All Available SVOD Offers (p.431).	
	28 Identify Subscribed Services (p.94).	

6.3.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
130 Subscribing to an SVOD Service (p.345).		
135 Opening a Ticket for an SVOD Movie (p.357).	136 Check SVOD Video Tickets (p.359).	
137 Unsubscribing from an SVOD Service (p.362).	138 Finish Service Subscription (p.364).	
	139 Remove Service Subscription (p.366).	

6.4 VOD Movie Packages

6.4.1 Screen Creation

Screen Creation	Sub-components	Sub-components
29 Create a VOD Movie Packages Catalog Screen (p.96).	30 Get List of VOD Movie Package Recommendations (p.98).	
	31 Get VOD Movie Packages (p.100).	
	32 Get VOD Package Contents (p.103).	
33 Create a Movie Screen (p.106).		
151 Creating a My Library Screen (p.413).	153 List My VOD Package Movies (p.418).	

6.4.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
129 Ordering a VOD Movie Package (p.341).		
52 Activate VOD Package Movie Ticket (p.157).		

6.5 Create a Movie Screen

6.5.1 Screen Creation

Screen Creation	Sub-components	Sub-components
33 Create a Movie Screen (p.106).	34 Get Movie Metadata (p.108).	
	42 Get User Ratings (p.135).	
	35 List Compatible Devices (p.113).	
	40 List VOD Movie Package Offers (p.129)	37 List TVOD Offers (p.117).
		38 List SVOD Offers for VOD Movie (p.122).
		40 List VOD Movie Package Offers (p.129).
		39 List Service Plan Offers for a VOD Movie (p.125).
	58 Get Recommended Movies (p.171).	

6.5.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
45 Playing a Movie (p.141).	46 Check if Account Has Rights to Play Movie (p.143).	47 Check TVOD and VOD Movie Package Video Tickets (p.144). 48 Check SVOD Subscriptions (p.148).
	49 Get Movie Playback URL (p.151).	
	135 Opening a Ticket for an SVOD Movie (p.357).	
	50 Using the Media Player (p.154).	
	51 Mark Last Position (p.155).	
43 Playing a Trailer (p.137).	44 Find a Movie Page (p.139).	
57 Rate a Movie (p.169).		
53 Manage a Wish List (p.160).	54 Check the Wish List (p.162).	
	55 Add to the Wish List (p.165).	
	56 Remove from the Wish List (p.167).	
	156 Show My Wishlist (p.426).	
119 The Purchase Process (p.312).		

7 Live Content Development Path

The tutorials listed in this section explain how to implement functionality required to support Live content. Tutorials explain how to construct a TV Guide and a Now Playing screen, how to create a Program Screen, and how to handle user purchases of channel packages.

7.1 Screen Creation

Screen Creation	Sub-components	Sub-components
63 Create a TV Guide (p.188).	64 Get a List of Channels (p.190).	
	65 Get List of Channel Programs (p.193)	
69 Create a Program Screen (p.202).	70 Get Program Metadata (p.205).	71 Get Channel Info from Internal ID (p.208).
	73 Get Compatible Devices (p.216).	
	75 Get Recommended Live Content (p.219).	
	157 Show My Reminders (p.428).	
79 Create a Now Playing Screen (p.225).	64 Get a List of Channels (p.190).	
	65 Get List of Channel Programs (p.193).	
158 My Offers / All Offers Screen (p.429).	160 List Channel Package Offers (p.438).	



7.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
66 Identify Subscribed Channels (p.197).	67 Get Account Channel Packages (p.198).	
	68 Get Channels in Package (p.200).	
72 List Offers for a Channel (p.211).		
76 Playing Live Programs (p.220).	77 Play Unencrypted Channel (p.221).	
	78 Playing Encrypted Channels (p.223).	
	50 Using the Media Player (p.154).	
140 Subscribe to a Channel Package (p.368).		
141 Unsubscribe from a Channel Package (p.371).	142 Finish Channel Package Subscription (p.373)	
	143 Remove Channel Package Subscription (p.375)	

Process	Secondary Steps	Tertiary Steps
162 Implementing Start Over (p.453).	163 Identify Start Over Enabled Programs (p.454).	
	164 Playing a Live Channel through Start Over (p.458).	
	165 Guidelines for Start Over Player Controls (p.462).	

Process	Secondary Steps	Tertiary Steps
166 Implementing Catch-Up (p.464).	167 List Catch-Up Offers (p.465).	
	168 Subscribe to a Catch-Up Service (p.471).	
	169 Unsubscribing from a Catch-Up Service (p.474).	
	172 List My Catch-Up Services (p.479).	
	174 Determining the Catch-Up Availability Period (p.487).	
	173 Identify Catch-Up Enabled Programs (p.483). 172 List My Catch-Up Services (p.479)	
	175 Checking if Account is Subscribed to Catch-Up (p.490).	
	176 Playing a Program through Catch-Up (p.492).	
	177 Marking Last Watched Position (p.496).	
	178 Guidelines for Catch-Up Player Controls (p.498).	
74 Setting Program Reminders (p.218).		



8 Service Plan Development Path

8.1 Screen Creation

Screen Creation	Sub-components	Sub-components
158 My Offers / All Offers Screen (p.429).	161 List All Available Service Plan Offers (p.444).	
151 Creating a My Library Screen (p.413).	155 List My Service Plans (p.423).	

8.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
131 Subscribe to a Service Plan (p.348).		
132 Unsubscribing from a Service Plan (p.351).	133 Finish Service Plan Subscription (p.353).	
	134 Remove Service Plan Subscription (p.355).	

10 Account Management Development Path

The tutorials in this section explain how to create and manage accounts and individual users within accounts. Management of devices, languages, payment methods, and parental control are all described.

- [10.1 Managing Accounts \(p.24\)](#)
- [10.2 Managing Users \(p.26\)](#)
- [10.3 Managing Devices \(p.28\)](#)
- [10.4 Managing Payment Methods \(p.28\)](#)
- [10.5 Language Management \(p.29\)](#)
- [10.6 Parental Control Management \(p.30\)](#)

10.1 Managing Accounts

10.1.1 Screen Creation

Screen Creation	Sub-components	Sub-components
80 Create a Login Screen (p.227).		
96 Building a My Account Screen (p.264).	97 Get Account Information (p.267).	
	126 Get an Online Account's Payment Method (p.331).	
	115 Get Device List (p.301). 115 Get Device List (p.301)	
	183 Get the Account's Current Language (p.506).	



10.1.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
81 Create an Account (p.229).	83 Get CAPTCHA Key (p.233). 84 Get CAPTCHA Image (p.235). 85 Submit User Details and Capture Challenge (p.237). 82 Resend Verification Email (p.231)	
Logging in With Facebook		
86 Logging In and Out (p.240).	87 Log into RiGHTv Account (p.243). 88 Log into COMPASS Account (p.246). 89 Get User List (p.248). 90 Log in the User (p.251). 91 Log Out of RiGHTv (p.253). 92 Log Out of COMPASS (p.255).	
118 Identify Account Type (p.308).		



106 Updating and Resetting Account Passwords (p.283).	107 Reset an Account's Password (p.284).	
	108 Update an Account's Password (p.288).	
Parental Control – see 10.6 Parental Control Management (p.30)		

10.2 Managing Users

10.2.1 Screen Creation

Screen Creation	Sub-components	Sub-components
96 Building a My Account Screen (p.264).	99 Get a User's Image (p.270).	
	104 Get the Current Opt-in/out Status (p.279).	

10.2.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
93 Creating and Updating Users (p.257).	94 Create a New User (p.259).	
	95 Update a User's Name (p.262).	
98 Managing User Images (p.269).	99 Get a User's Image (p.270).	
	100 Create a User Image (p.272).	
	101 Update a User Image (p.274).	
	102 Delete a Profile Image (p.276).	
103 Managing User Usage Data Opt-In/out Status (p.278).	104 Get the Current Opt-in/out Status (p.279).	
	105 Opt-in and Opt-out (p.281).	
109 Purchase PIN Management (p.291).	110 Get Purchase PIN (p.292).	
	111 Reset a User's PIN (p.294).	
	112 Update a User's PIN (p.296).	

10.3 Managing Devices

10.3.1 Screen Creation

Screen Creation	Sub-components	Sub-components
96 Building a My Account Screen (p.264).	115 Get Device List (p.301).	

10.3.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
113 Device Management (p.298).	115 Get Device List (p.301).	
	114 Register a New Device (p.299).	
	116 Remove Device (p.304).	
	117 Update Device Name (p.306).	

10.4 Managing Payment Methods

10.4.1 Screen Creation

Screen Creation	Sub-components	Sub-components
96 Building a My Account Screen (p.264).	126 Get an Online Account's Payment Method (p.331).	

10.4.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
Overview	120 Managing Payment Methods (p.318).	
126 Get an Online Account's Payment Method (p.331).		
121 Register a New Payment Method via Adyen (p.320).	122 Get Payment Registration Properties (p.321).	
	123 Register Payment Method (p.324).	
	124 Complete Registration (p.327).	
125 Register a New Payment Method via the Agnostic Payment Service (p.329).		
127 Delete Payment Method (p.334).		

10.5 Language Management

10.5.1 Screen Creation

Screen Creation	Sub-components	Sub-components
96 Building a My Account Screen (p.264).	183 Get the Account's Current Language (p.506).	

10.5.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
Overview	179 Working with Languages (p.500).	
181 Set the Language for a New Account (p.503).	183 Get the Account's Current Language (p.506).	
	180 Get a List of Supported Languages (p.501).	
184 Specifying the Response Language (p.508).		

10.6 Parental Control Management

10.6.1 Screen Creation

Screen Creation	Sub-components	Sub-components
96 Building a My Account Screen (p.264).	97 Get Account Information (p.267).	

10.6.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
Overview	185 Parental Control (p.509).	
188 Changing Parental Pins (p.514).	189 Updating the Parental Pin (p.515).	
	190 Resetting the Parental Pin (p.517).	
	191 Get a List of Parental Ratings Used (p.519).	



Process	Secondary Steps	Tertiary Steps
187 Determine Parental Level of an Entity (p.513).		
186 Verify the Parental Pin (p.511).		
192 Set the Parental Control Threshold (p.521).		
193 Exclude Adult Content from Returned Lists (p.522).		

11 Common Screen Development

The tutorials listed in this section explain how to construct general screens that should be present in the application, regardless of which business models are implemented.

- [11.0.1 The Home Screen \(p.32\)](#)
- [11.0.2 The My Library Screen \(p.32\)](#)
- [11.0.3 The My Offers/All Offers Screen \(p.34\)](#)

11.0.1 The Home Screen

11.0.1.1 Screen Creation

Screen Creation	Sub-components	Sub-components
13 Create a Home Screen (p.38).	150 Get List of Recommendations (p.406).	

11.0.2 The My Library Screen

11.0.2.1 Screen Creation

Screen Creation	Sub-components	Sub-components
151 Creating a My Library Screen (p.413).	List My Movies	
	153 List My VOD Package Movies (p.418).	
	154 List My SVOD Subscriptions (p.420).	
	155 List My Service Plans (p.423).	
	172 List My Catch-Up Services (p.479).	



	156 Show My Wishlist (p.426).	
	157 Show My Reminders (p.428).	

11.0.2.2 Process Implementation

Process	Secondary Steps	Tertiary Steps
Subscribe to a Service Plan	131 Subscribe to a Service Plan (p.348).	
Subscribe to a Channel Package	140 Subscribe to a Channel Package (p.368).	
Subscribe to an SVOD Service	130 Subscribing to an SVOD Service (p.345).	
Unsubscribe from a Service Plan	132 Unsubscribing from a Service Plan (p.351).	133 Finish Service Plan Subscription (p.353).
		134 Remove Service Plan Subscription (p.355).
Unsubscribe from a Channel Package	141 Unsubscribe from a Channel Package (p.371).	142 Finish Channel Package Subscription (p.373).
		143 Remove Channel Package Subscription (p.375).
Unsubscribe from an SVOD Service	137 Unsubscribing from an SVOD Service (p.362).	138 Finish Service Subscription (p.364).
		139 Remove Service Subscription (p.366).



11.0.3 The My Offers/All Offers Screen

11.0.3.1 Screen Creation

Screen Creation	Sub-components	Sub-components
158 My Offers / All Offers Screen (p.429).	159 List All Available SVOD Offers (p.431).	
	160 List Channel Package Offers (p.438).	
	161 List All Available Service Plan Offers (p.444).	
	167 List Catch-Up Offers (p.465).	

12 Common Functionality Development

The tutorials listed in this section explain how to implement general functionality within a client application, regardless of which business models are implemented.

- [12.1 Search \(p.35\)](#)
- [12.2 Display Images \(p.35\)](#)
- [12.3 Get List of Recommendations \(p.36\)](#)
- [12.4 Registering Events \(p.36\)](#)
- [12.5 Sharing Over Social Networks \(p.37\)](#)

12.1 Search

Process	Secondary Steps	Tertiary Steps
Overview	144 Performing a Search (p.377).	
145 Perform a General Search (p.378).		
146 Performing a Live Programs Search (p.384).		
147 Performing a Video Search (p.390).		
148 Adding Auto-suggestion to Search (p.397).		

12.2 Display Images

Process	Secondary Steps	Tertiary Steps
149 Display Images (p.402).		

12.3 Get List of Recommendations

Process	Secondary Steps	Tertiary Steps
150 Get List of Recommendations (p.406).		

12.4 Registering Events

Process	Secondary Steps	Tertiary Steps
195 Registering Events for Content Discovery (p.527)	196 Register Event: Order Video (p.529).	
	197 Register Event: Preview Trailer (p.532).	
	198 Register Event: Content End (p.534).	
	199 Register Event: Rating (p.536).	
	200 Register Event: Set Reminder (p.538).	
	201 Register Event: Channel Zap (p.540).	
	202 Register Search Event (p.542).	
205 Apps and Analytics (p.550)		

12.5 Sharing Over Social Networks

Process	Secondary Steps	Tertiary Steps
194 Sharing Via Facebook (p.524)		

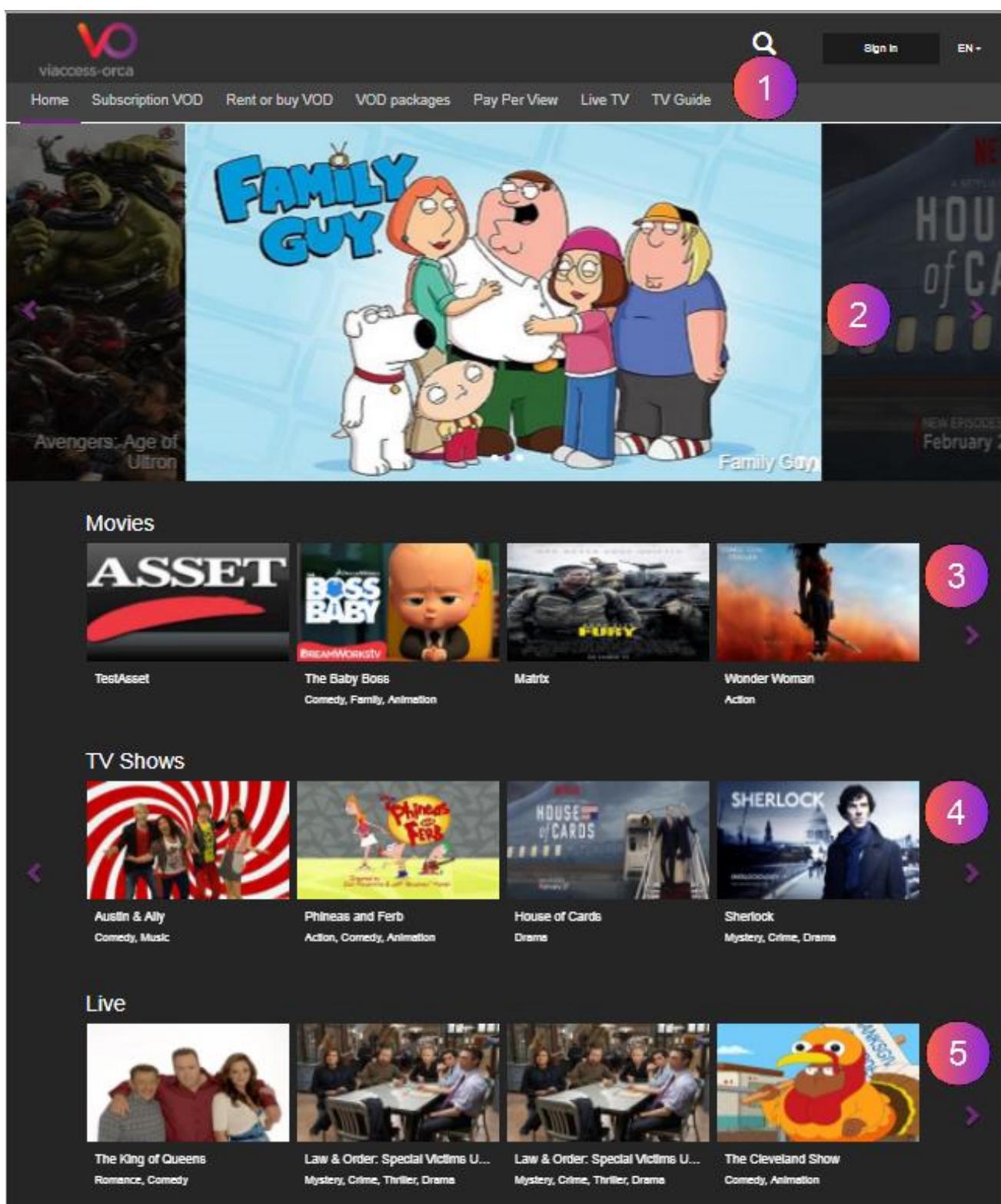
13 Create a Home Screen

13.1 Screen Overview

The Home screen is the screen to which the apps should open on start-up. The Home screen should showcase the Operator's content offering of all content types and across all supported business models. The Home screen should include.

1. A menu providing access to the home pages for the various business models offered.
2. A list of promoted content (of all types, from all business models)
3. A list of recommended movies (from all business models)
4. A list of recommended TV Shows (from all business models)
5. A list of recommended live TV shows

For each movie/TV show listed, its thumbnail poster should be shown.



13.2 Requests and Responses

The `GetVideoRecommendationList` and `GetLiveRecommendationList` calls are used to create lists of promoted and recommended VOD and Live content respectively. See [150 Get List of Recommendations \(p.406\)](#) for more information.

13.3 Recommendation Requests

The table below provides best-practice examples of recommendation calls for the Home screen.

Recommendation List	Blend	Categories	Type
Promotions	Promotions	Movies, TV Shows, SVOD Services, VOD Packages	(all)
	<pre>https://<operator>-atp.tvaas.com/compass/GetVideoRecommendationList?blend=promotions&random=false&subscription=false&categories=Movies%2CTV+Shows%2CSVOD+Services%2CVOD+Packages&max_results=5&max_pr_level=9998&client=json&language_code=en</pre> <div style="border: 1px solid black; padding: 5px;"> Note If a season promotion is clicked (i.e. in the response <i>contentType = Season</i> and <i>type = promotions</i>), open the relevant Season Page (using the <i>externalSeasonId</i> identifier). </div>		
Recommended Movies	Movies	Movies, TV Shows, SVOD Services, VOD Packages	(all)
	<pre>https://<operator>-atp.tvaas.com/compass/GetVideoRecommendationList?blend=movies&random=false&subscription=false&categories=Movies%2CTV+Shows%2CSVOD+Services%2CVOD+Packages&max_results=10&max_pr_level=9998&client=json&language_code=en</pre>		



Recommendation List	Blend	Categories	Type
Recommended VOD TV Series Season	Series	Movies, TV Shows, SVOD Services, VOD Packages	Season
<pre>https://<operator>- atp.tvaas.com/compass/GetVideoRecommendationList? contentType=Season&blend=series&random=false& subscription=false& categories=Movies%2CTV+Shows%2CSVOD+Services%2 CVOD+Packages&max_results=10&max_pr_ level=9998&client=json&language_code=en</pre>			
<p>Note</p> <p>If a season recommendation is clicked (i.e. <i>contentType = Season</i>), open the relevant TV Show Series Page (using the <i>externalSeriesId</i> identifier).</p>			
Recommended Live Programs	Live	-	(all)
<pre>https://<operator>- atp.tvaas.com/compass/GetLiveRecommendationList? type=all&blend=live&max_results=10&max_pr_ level=9998&client=json&language_code=en</pre>			

13.4 Related Tutorials

To	Refer to Tutorial
Learn more about recommendations	150 Get List of Recommendations (p.406)
Display attachment images	149 Display Images (p.402)



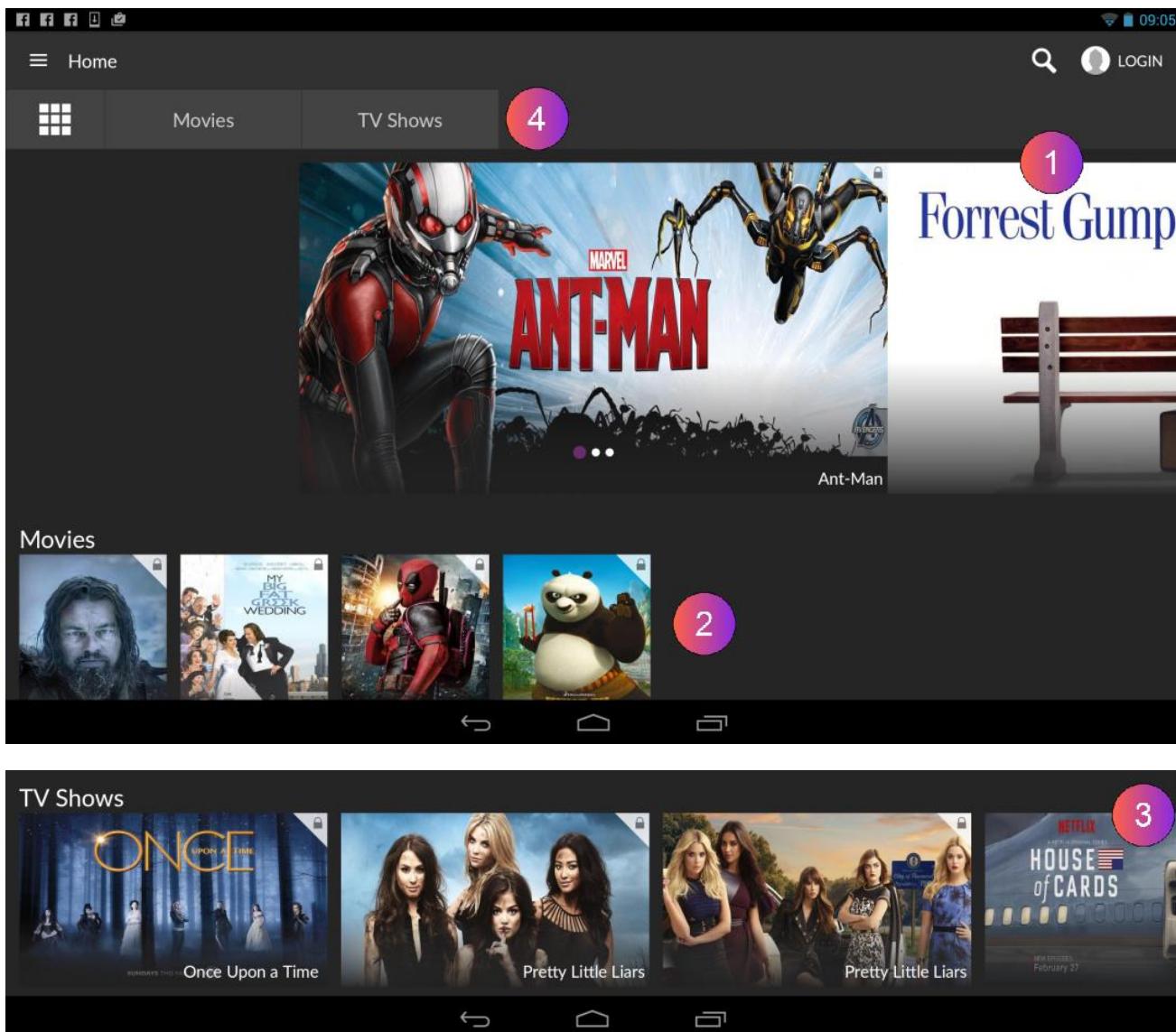
14 Create a TVOD Home Screen

14.1 Screen Overview

A TVOD home screen should showcase the TVOD offers. Such a screen should include:

1. A list of content available through TVOD offers that the operator wants to promote
2. A list of recommended movies available through TVOD offers
3. A list of recommended TV shows available through TVOD offers
4. Tabs to access the TVOD Movies and TVOD TV Shows catalog screens.

Each content item should be represented by a thumbnail poster.



14.2 Related Tutorials

To	Refer to Tutorial
Build lists of TVOD recommendations	15 Get List of TVOD Recommendations (p.44)

15 Get List of TVOD Recommendations

15.1 Overview

The TVOD home screen in the client application should display:

- A list of movies and TV shows available in TVOD offers that the operator wants to promote
- A list of recommended movies available in TVOD offers
- A list of recommended TV shows available in TVOD offers.

For each movie/TV show listed, its thumbnail poster should be shown.

15.2 Requests and Responses

The `GetVideoRecommendationList` call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.

15.3 Recommendation Requests

The table below provides best-practice examples of recommendation calls for the TVOD home screen.



Recommen dation List	Blend	Categories	Con tent Typ e
1. Promotions	promotions	Movies, TV Shows	-
	<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=promotions&random=false&subscription=false&categories=Movies%2CTV+Shows&max_results=5&max_pr_level=9998&client=json&language_code=en</code>		
2. Recommen ded Movies	movies	Movies	-
	<code>https://<operator>/compass/GetVideoRecommendationList?blend=movies&random=false&subscription=false&categories=Movies&max_results=10&max_pr_level=9998&client=json&language_code=en</code>		
3. Recommen ded TV Shows	series	TV Shows	Sea son
	<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?content_type=Season&blend=series&random=false&subscription=false&categories=TV+Shows&max_results=10&max_pr_level=9998&client=json&language_code=en</code>		

15.4 Related Tutorials

To	Refer to Tutorial
Create a TVOD Home Screen	14 Create a TVOD Home Screen (p.42)
Display attachment images	149 Display Images (p.402)
Learn more about recommendations	150 Get List of Recommendations (p.406)



16 Create a TVOD Movies Catalog Screen

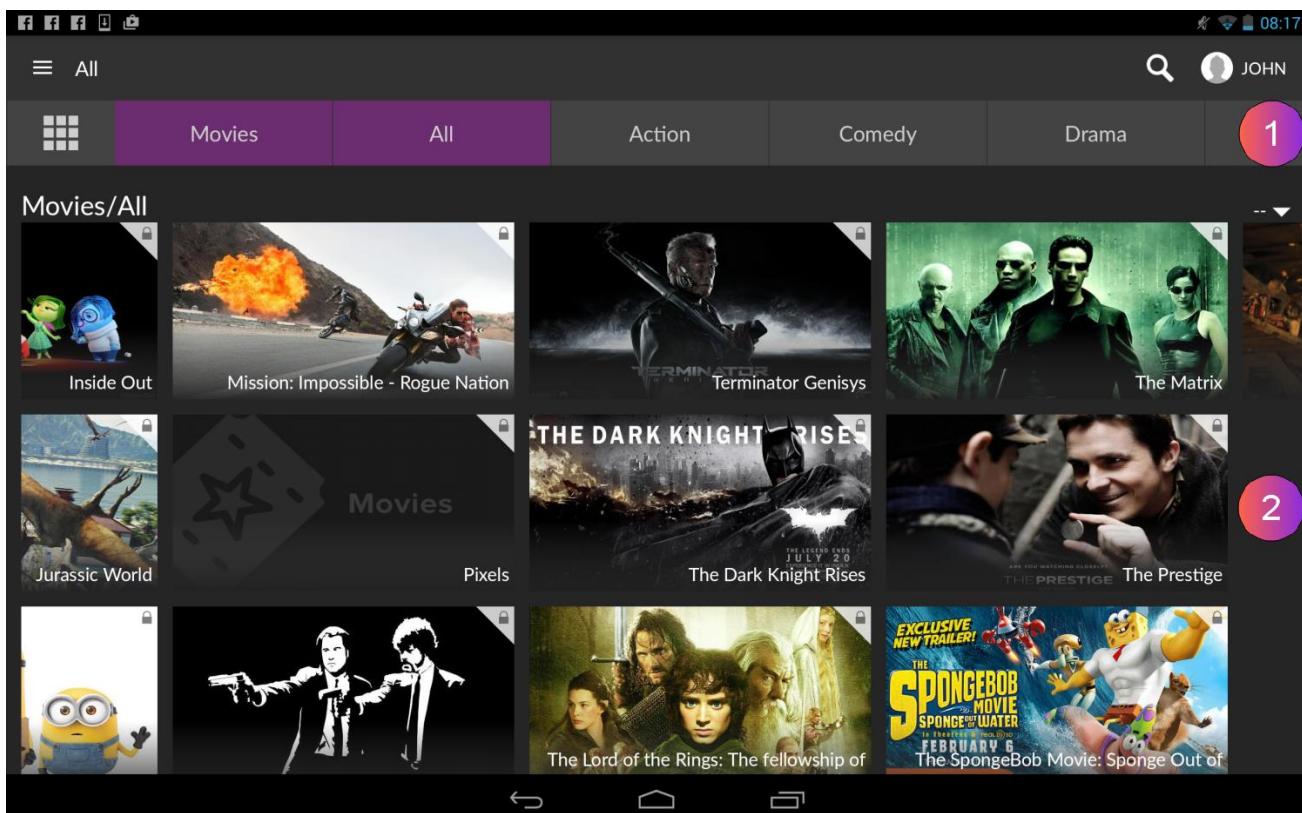
16.1 Screen Overview

The TVOD Movies Catalog screen is accessed from a tab on the TVOD Home Screen (see [14 Create a TVOD Home Screen \(p.42\)](#)). This screen allows the user to browse the operator's movies catalog.

The user must be able to click any of the movies listed to access the relevant Movie screen ([33 Create a Movie Screen \(p.106\)](#)).

The screen should show:

1. A list of movie categories
2. A list of all movies within the selected category.



16.2 Related Tutorials

To	Refer to Tutorial
Create a TVOD Home Screen	14 Create a TVOD Home Screen (p.42)
Get a list of TVOD movie categories	17 Get TVOD Movie Categories (p.49)
Display attachment images	149 Display Images (p.402)

17 Get TVOD Movie Categories

17.1 Overview

Typically, all TVOD movie categories are located under the Movies category in the TVOD catalog. The VOD Movies catalog screen should show these categories.

Excerpt of TVOD Movies Category in VOD Catalog <pre> > TVOD Services > Movies > All > Action > Science Fiction > Comedy > Drama > Foreign > Horror > Kids > Mystery > Romance > TestCategory > EmptyCategory > TV Shows </pre>	List of Movies Categories in Application
--	---

17.2 Requests and Responses



Request: [GetUnifiedList](#) (from RiGHTv Front-end API)

The list of categories returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The ID of the folder whose contents you want listed. Typically set to movies.
<i>order</i>	The order in which the results should be listed, for example by name for alphabetical or <code>tree</code> for the same order as listed by the operator.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Page \(from RiGHTv Front-end API\)](#)

The returned entity includes various properties. For obtaining a list of movie categories, the most relevant are:

Property	Description
<i>name</i>	The name of the category.
<i>externalId</i>	The external ID of the category.

17.3 Related Tutorials

To	Refer to Tutorial
Create a TVOD Movies catalog screen	16 Create a TVOD Movies Catalog Screen (p.47)
Get a list of TVOD movies in a category	18 Get TVOD Movies (p.52)



17.4 Example

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=Movies&client=json

This request calls for a list of categories below the `Movies` category.

The result set for one of the categories, Drama, is shown below. The external ID and Name are indicated.

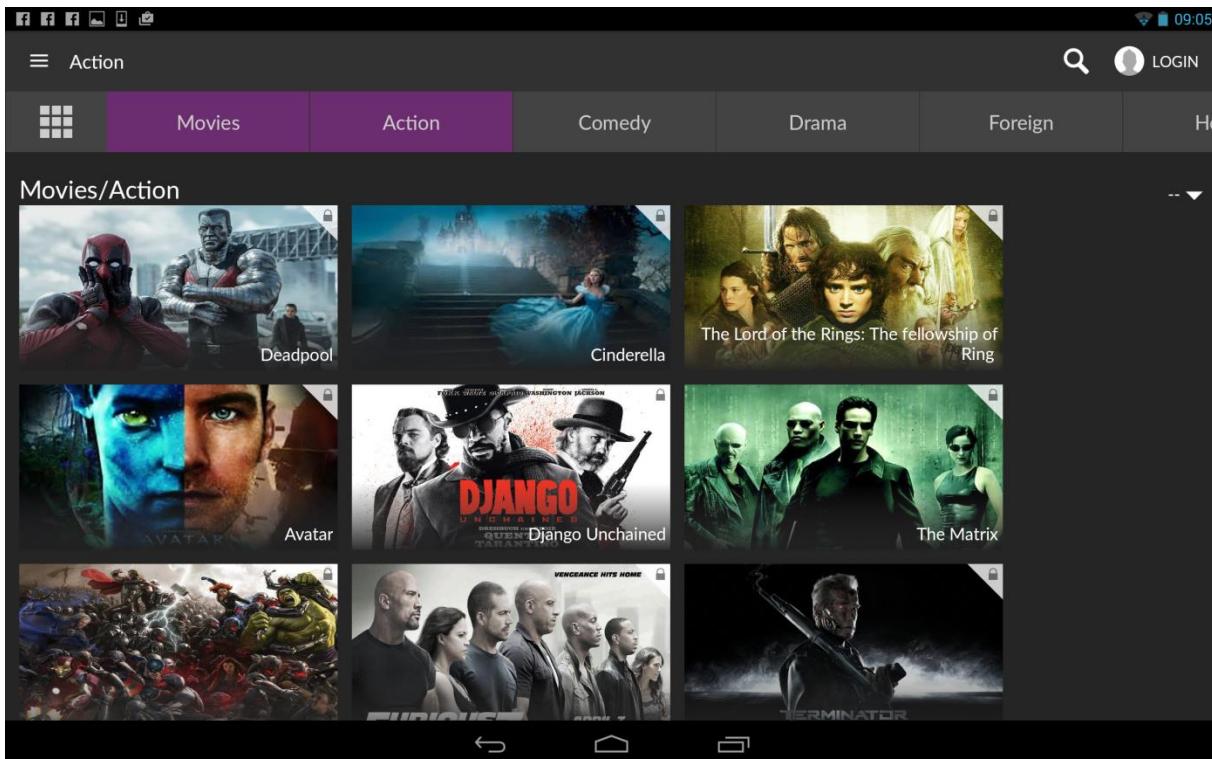
```
{  
    "template": "Folder",  
    "isSecured": false,  
    "attachments": [],  
    "description": "Drama",  
    "externalId": "Drama",  
    "isBlockedBrowsing": false,  
    "parentId": 201,  
    "responseElementType": "Page",  
    "name": "Drama",  
    "extrafields": [  
        {  
            "responseElementType": "Extrafield",  
            "name": "exclude",  
            "value": ""  
        }  
    ],  
    "securityGroups": [],  
    "id": 206,  
    "status": 2  
},
```



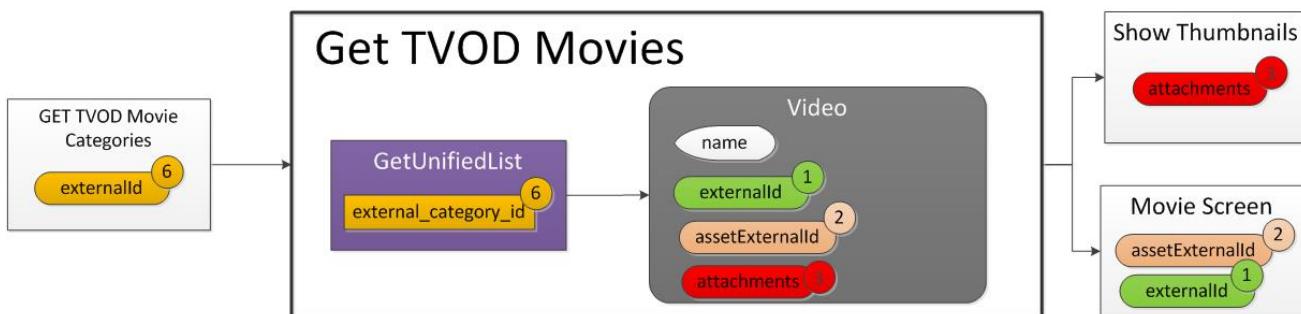
18 Get TVOD Movies

18.1 Overview

Typically, all TVOD movie categories are located under the Movies category in the VOD catalog. Each category includes multiple movies. The TVOD Movies catalog screen in the application should show the movies within the category the user selects. Shown below is the contents of the Action category.



18.2 Requests and Responses



Request: GetUnifiedList (from RiGHTv Front-end API)

The list of movies returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The ID of the VOD catalog category whose contents you want listed.
<i>order</i>	The order in which the results should be listed, for example by <code>name</code> for alphabetical or <code>tree</code> for the same order as listed by the operator.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: Video (from RiGHTv Front-end API)

The returned entity includes various properties. For displaying the movies listed in a category, the most relevant are:

Property	Description
<i>name</i>	The name of the movie.
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>assetExternalId</i>	The ID of the video asset in the Content Library
<i>externalId</i>	The ID of the movie page in the VOD Catalog.

18.3 Related Tutorials

To	Refer to Tutorial
Create a TVOD Movies catalog screen	16 Create a TVOD Movies Catalog Screen (p.47)
Get a list of TVOD movies in a category	18 Get TVOD Movies (p.52)



Display attachment images	149 Display Images (p.402)
---------------------------	--

18.4 Example

In the example in [17 Get TVOD Movie Categories \(p.49\)](#), the sample response for the Drama category in the Movies category was shown. The *externalId* for that category is Drama.

```
{
  "template": "Folder",
  "isSecured": false,
  "attachments": [],
  "description": "Drama",
  "externalId": "Drama", 
  "isBlockedBrowsing": false,
  "parentId": 201,
  "responseElementType": "Page",
  "name": "Drama", 
  "extrafields": [
    {
      "responseElementType": "Extrafield",
      "name": "exclude",
      "value": ""
    }
  ],
  "securityGroups": [],
  "id": 206,
  "status": 2
},
```

This request calls for a list of movies in the Drama category.

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=Drama&client=json

The relevant sections of the result set for one of the movies, Argo, is shown below. The name and attachment details are indicated.



```
"rentalPeriod": "",  
"name": "Argo",  
"broadcastTime": 0,  
"shortName": "",  
"discountId": "",  
"status": 2,  
"advisories": "",  
"template": "vod_service",  
"attachments": [  
{  
    "responseElementType": "Attachment",  
    "assetId": "",  
    "name": "VerticalImage",  
    "assetName": "",  
    "value": "/attachments/argo_vertical_after_edit.jpg"  
},  
,  
{  
    "responseElementType": "Attachment",  
    "assetId": 80006,  
    "name": "Trailer",  
    "assetName": "Test Trailer",  
    "value": "true"  
},  
,  
{  
    "responseElementType": "Attachment",  
    "assetId": "",  
    "name": "HorizontalImage",  
    "assetName": "",  
    "value": "/attachments/argo-horizontal-After Edit.jpg"  
}]
```

19 Create a TVOD TV Shows Catalog Screen

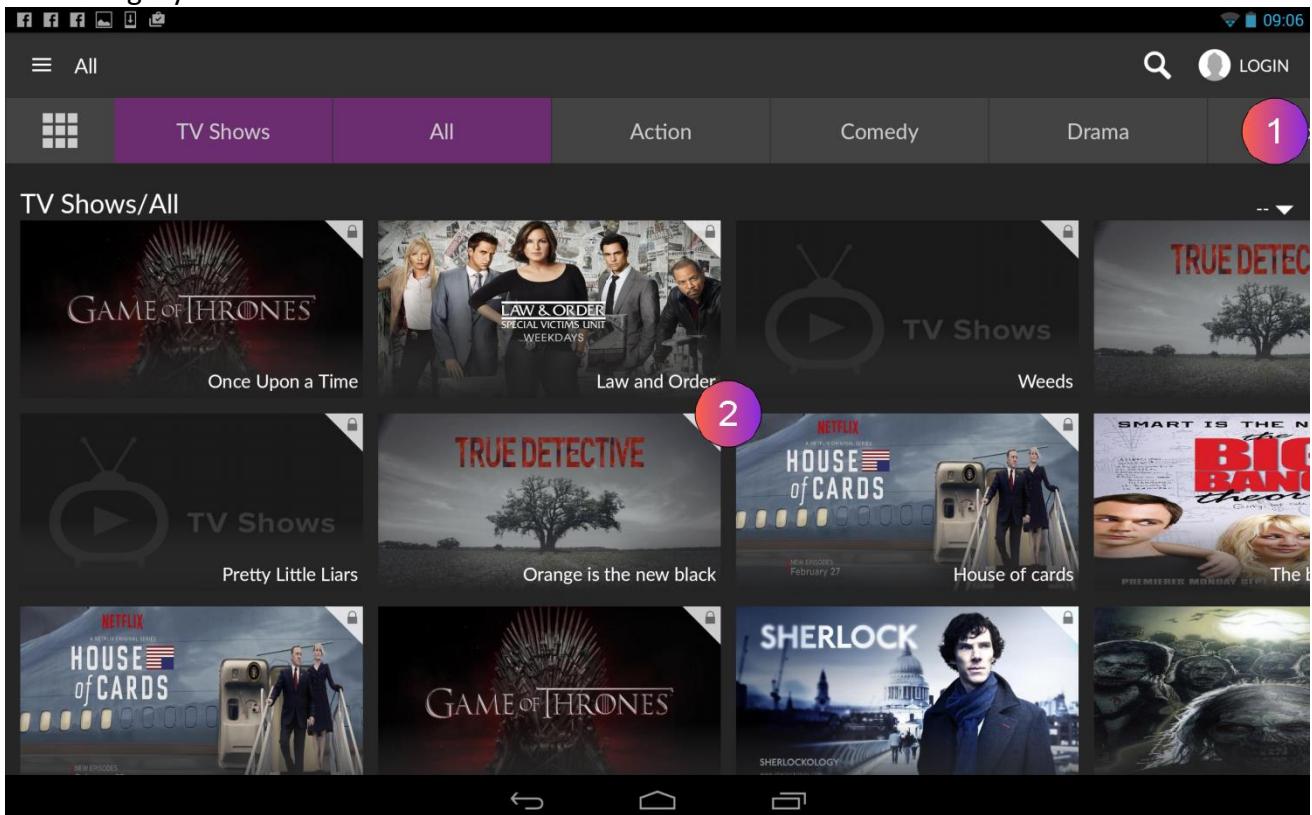
19.1 Screen Overview

The TVOD TV Shows Catalog screen is accessed from a tab on the TVOD Home Screen (see [14 Create a TVOD Home Screen \(p.42\)](#)). This screen allows the user to browse the operator's TV Show catalog.

The user must be able to click any of the TV Shows listed to access the relevant TV Show screen (see [60 Create a TVOD TV Show Series Screen \(p.175\)](#)). The TV Shows Catalog screen may also list single seasons. Clicking one of those should open the relevant Season screen (see [61 Create a VOD Season Screen \(p.179\)](#)).

The screen should show:

1. A list of TV Show categories
2. A list of all TV shows (series) within the selected category, as well as any single seasons in the category.



19.2 Related Tutorials

To	Refer to Tutorial
Create a TVOD Home Screen	14 Create a TVOD Home Screen (p.42)
Get a list of TVOD TV show categories	20 Get VOD TV Show Categories (p.58)
Get a list of TVOD TV shows and series in a category	21 Get VOD TV Show Series and Seasons in a Category (p.61)

20 Get VOD TV Show Categories

20.1 Overview

Typically, all TVOD TV Show categories are located under the TV Shows category in the VOD catalog. The TVOD TV Shows catalog screen should show these categories.

Excerpt of VOD TV Shows Category in VOD Catalog:



List of TV Show Categories in Application:



20.2 Requests and Responses



Request: GetUnifiedList (from RiGHTv Front-end API)

The list of categories returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The ID of the category whose contents you want listed. Typically set to TV Shows.
<i>order</i>	The order in which the results should be listed, for example by <code>name</code> for alphabetical or <code>tree</code> for the same order as listed by the operator.
<i>filter_empty_categories</i>	Set to <code>true</code> to include categories even if empty, or <code>false</code> to exclude empty categories.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: Page (from RiGHTv Front-end API)

The returned entity includes various properties. For obtaining a list of TV show categories, the most relevant are:

Property	Description
<i>name</i>	The name of the category.
<i>externalId</i>	The external ID of the category.

20.3 Related Tutorials

To	Refer to Tutorial
Create a TVOD TV Shows catalog screen	19 Create a TVOD TV Shows Catalog Screen (p.56)
Get a list of TVOD TV shows and series in a category	21 Get VOD TV Show Series and Seasons in a Category (p.61)

20.4 Example

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_



```
id=TV%20Shows&client=json
```

This request calls for a list of categories below the TV Shows category.

The result set for one of the categories, Action_TV, is shown below. The external ID and Name are indicated.

```
{
    "template": "Folder",
    "isSecured": false,
    "attachments": [],
    "description": "Action TV",
    "externalId": "Action_TV",
    "isBlockedBrowsing": false,
    "parentId": 212,
    "responseElementType": "Page",
    "name": "Action",
    "extrafields": [
        {
            "responseElementType": "Extrafield",
            "name": "exclude",
            "value": ""
        }
    ],
    "securityGroups": [],
    "id": 214,
    "status": 2
},
```



21 Get VOD TV Show Series and Seasons in a Category

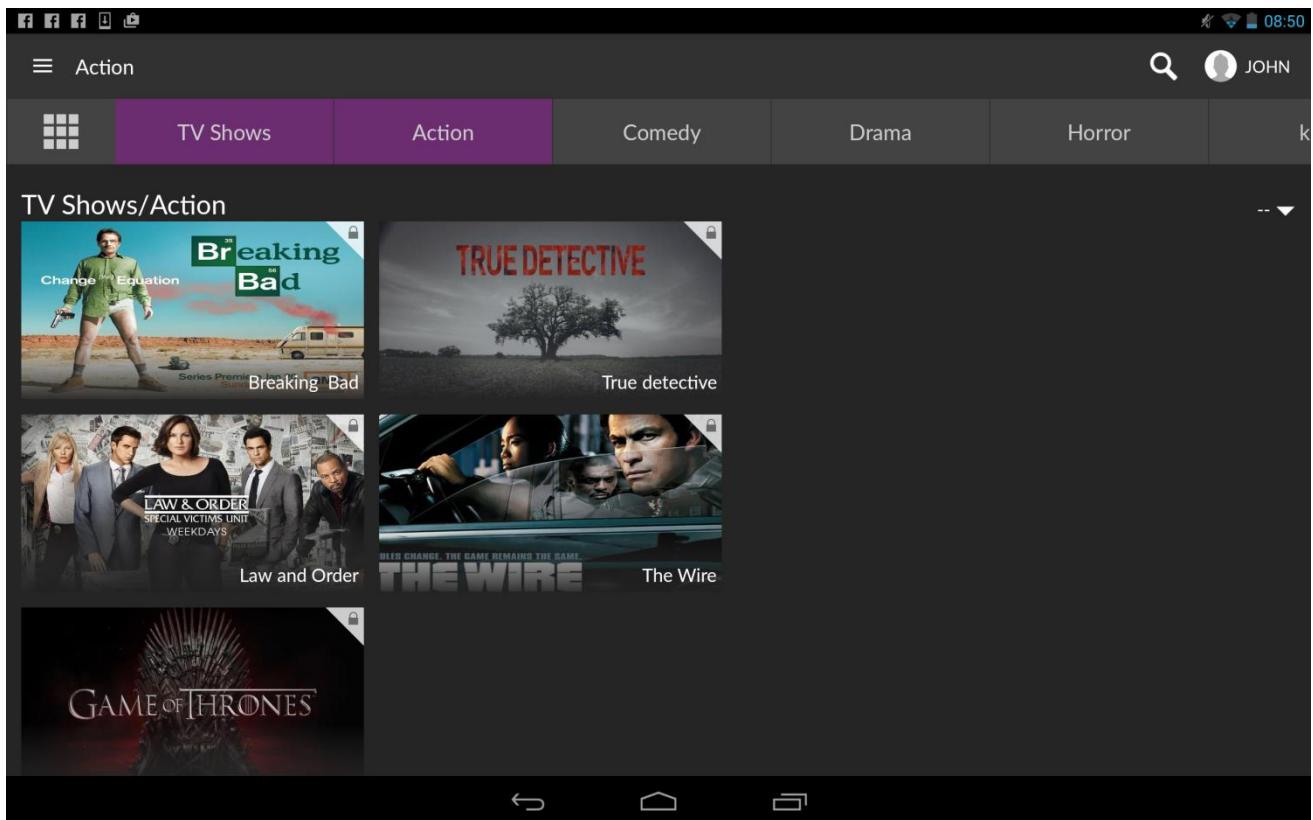
21.1 Overview

Typically, all TVOD TV Show categories are located under the TV Shows category in the VOD catalog. Each category includes multiple TV Shows (series). The TVOD TV Shows catalog screen in the application should show all the series within the category the user selects.

The TVOD TV Shows catalog screen may also include single seasons.

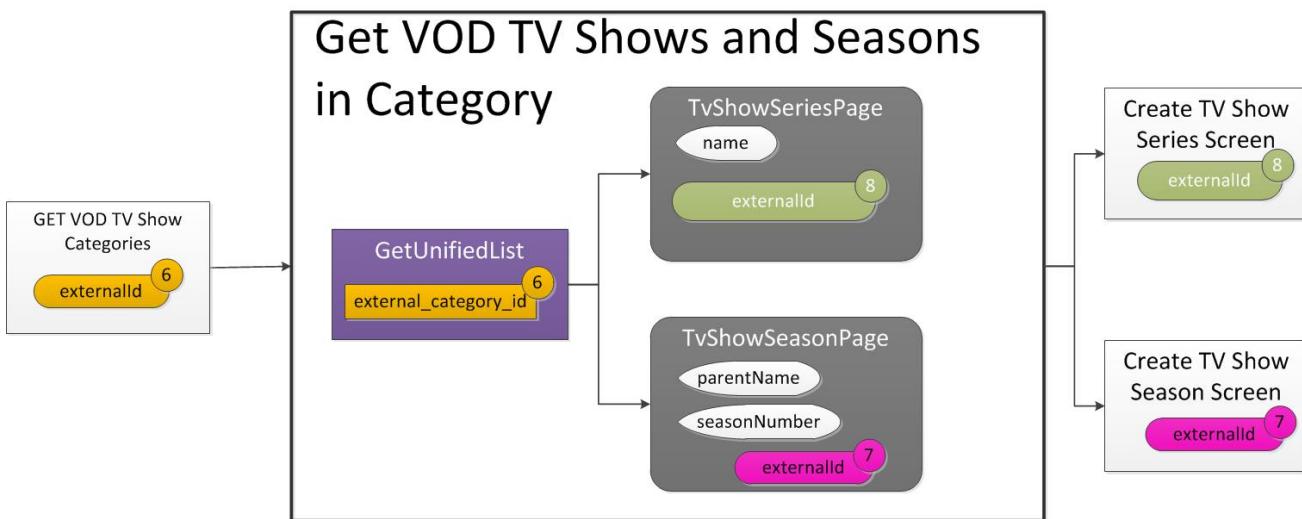
If a TV Show (series) is clicked, the relevant TV Show page should be opened. If a season is clicked, the relevant Season page should be opened.

Shown below is the contents of the Action category.



Each TV show series is represented in the VOD Catalog as its own category. The example below shows the list of TV show series (on the right) within the Action category.

21.2 Requests and Responses



Request: [GetUnifiedList](#) (from RiGHTv Front-end API)

The list of TV shows returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<code>external_category_id</code>	The ID of the VOD catalog category whose contents you want listed.
<code>order</code>	The order in which the results should be listed, for example by <code>name</code> for alphabetical or <code>tree</code> for the same order as listed by the operator.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [TvShowSeriesPage](#) (from RiGHTv Front-end API)

The `TvShowSeriesPage` response includes the `TvShowSeries` entity which describes the TV Show (series). The returned entity includes various properties. For displaying the TV Shows (series) listed in a category, the most relevant are:

Property	Description
<i>name</i>	The name of the TV show.
<i>externalId</i>	The ID of the TV show in the VOD Catalog.
<i>attachments</i>	The poster images for the TV Show. See 149 Display Images (p.402) to learn how to retrieve the image files.

Response: [TvShowSeasonPage](#) (from RiGHTv Front-end API)

The *TvShowSeasonPage* response includes the *TvShowSeason* entity which describes a TV Show season. The returned entity includes various properties. For displaying the seasons listed in a category, the most relevant are:

Property	Description
<i>externalId</i>	The ID of the season in the VOD Catalog.
<i>attachments</i>	The poster images for the TV Show. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>parentName</i>	The name of the TV Show (series) to which the season belongs.
<i>seasonNumber</i>	The number of the season in the series. The name displayed should be constructed as follows: <i>parentName Season: seasonNumber</i> For example: Weeds: Season 7

21.3 Related Tutorials

To	Refer to Tutorial
Create a TVOD TV Shows catalog screen	19 Create a TVOD TV Shows Catalog Screen (p.56)
Get a list of TVOD TV show categories	20 Get VOD TV Show Categories (p.58)



Display attachment images	149 Display Images (p.402)
Create a TV Show Screen	60 Create a TVOD TV Show Series Screen (p.175)
Create a Season Screen	61 Create a VOD Season Screen (p.179)

21.4 Example

In the example in [20 Get VOD TV Show Categories \(p.58\)](#), the sample response for the Action category in the TV Shows category was shown. The *externalId* for that category is Action_TV.

```
{
  "template": "Folder",
  "isSecured": false,
  "attachments": [],
  "description": "Action_TV",
  "externalId": "Action_TV",
  "isBlockedBrowsing": false,
  "parentId": 212,
  "responseElementType": "Page",
  "name": "Action",
  "extrafields": [
    {
      "responseElementType": "Extrafield",
      "name": "exclude",
      "value": ""
    }
  ],
  "securityGroups": [],
  "id": 214,
  "status": 2
},
```

This request calls for a list of TV shows in the Action_TV category.

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=Action_TV&client=json

The partial response below is for a TV Show Series found within the category. Note that *responseElementType* is *TvShowSeriesPage*.



```
"externalId": "Truedetective",
"description": "",
"isBlockedBrowsing": false,
"parentId": 227,
"responseElementType": "TvShowSeriesPage",
"name": "True detective",
"extrafields": [],
"securityGroups": [],
"id": 5924,
"status": 2
```

The partial response below is for a Season found within the category. Note that the *responseElementType* is *TvShowSeasonPage*.

```
"responseElementType": "TvShowSeasonPage"
"name": "Weeds 7",
"extrafields": [],
"securityGroups": [],
"id": 6100,
"status": 2
```

22 Get Seasons in a TVOD TV Show Series

22.1 Overview

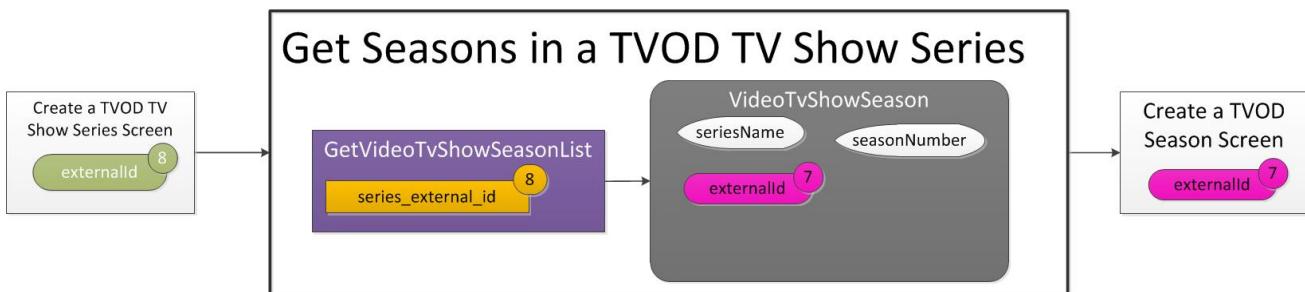
Each TV Show Series consists of one or more seasons. For example, the screen below shows that seasons 4 and 5 of the series Breaking Bad are available.



Each TV show series season is represented in the VOD Catalog as its own category. The example below shows the list of seasons (categories) within the category for Breaking Bad.

Catalog		Breaking Bad		
		2 Results 0 selected		
		Title	Status	Last Change
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Breaking Bad Season 4	Published	07/09/2015
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Breaking Bad Season 5	Published	07/09/2015

22.2 Requests and Responses



Request: [GetVideoTvShowSeasonList](#) (from RiGHTv Front-end API)

The list of TV seasons returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<code>series_external_id</code>	The ID of the TV Show Series.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [VideoTvShowSeason](#) (from RiGHTv Front-end API)

The returned entity includes various properties. For displaying the seasons in a TV show series, the most relevant are:

Property	Description
<code>seriesName</code>	The name of the TV show series.
<code>seasonNumber</code>	The number of the season in the series. The name of the season to display should be constructed as: <code>seriesName: Season seasonNumber</code> .
<code>externalId</code>	The ID of the season in the VOD Catalog.
<code>attachments</code>	The <code>name</code> and <code>value</code> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.

22.3 Related Tutorials

To	Refer to Tutorial
Create a TVOD TV Show Series screen	60 Create a TVOD TV Show Series Screen (p.175)
Create a TVOD Season screen	61 Create a VOD Season Screen (p.179)
Display attachment images	149 Display Images (p.402)

22.4 Example

This request calls for a list of seasons in the TV Show Series Weeds.

`https://<operator>.tvaas.com/RTEFacade/GetVideoTvShowSeasonList?series_external_id=Weeds&language_code=en&client=json`

The response for Weeds season 7 is shown below.



```
"response": [
    {
        "studio": "",
        "prLevel": 3,
        "attachments": [
            {
                "responseElementType": "Attachment",
                "assetId": "",
                "name": "VerticalImage",
                "assetName": "",
                "value": "/attachments/weeds-season-7-three-years-later
                    .jpg"
            },
            {
                "responseElementType": "Attachment",
                "assetId": "",
                "name": "HorizontalImage",
                "assetName": "",
                "value": "/attachments/weeds horizontal.jpg"
            }
        ],
        "seasonCategoryInfoList": [
            {
                "template": "TV_Shows_Season",
                "responseElementType": "TvShowCategoryInfo",
                "attachments": [],
                "extrafields": [],
                "externalId": "52574354",
                "description": "",
                "status": 2
            },
            ...
        ]
    }
]
```

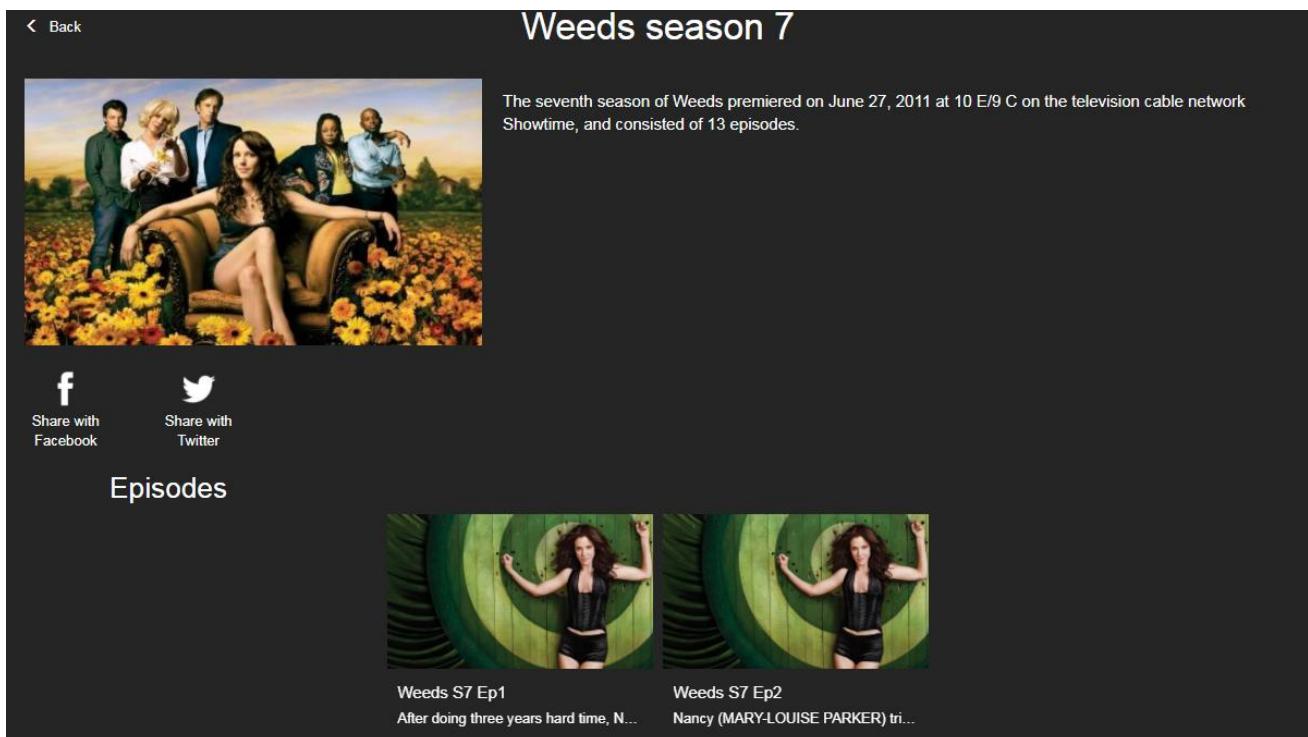
```
[  
  {  
    "template": "TV_Shows_Season",  
    "responseElementType": "TvShowCategoryInfo",  
    "attachments": [],  
    "extrafields": [],  
    "externalId": "Weeds7",  
    "description": "",  
    "status": 2  
  },  
,  
  "year": 2011,  
  "seriesName": "Weeds",  
  "prName": "TV-14",  
  "externalId": "Weeds_season_7",  
  "description": "The seventh season of Weeds premiered on June  
  27, 2011 at 10 E/9 C on the television cable network  
  Showtime, and consisted of 13 episodes.",  
  "shortDescription": "",  
  "seasonNumber": 7,  
  "countries": [],  
  "seriesExternalId": "Weeds",  
  "contentProviderName": "",  
  "numberOfEpisodes": "",  
  "responseElementType": "VideoTvShowSeason",  
  "genres": [],  
  "name": "Weeds season 7",  
  "extrafields": [],  
  "parentalAdvisories": "",  
  "contributors": [],  
  "shortName": "",  
  "contentProviderExternalId": ""  
},  
]
```



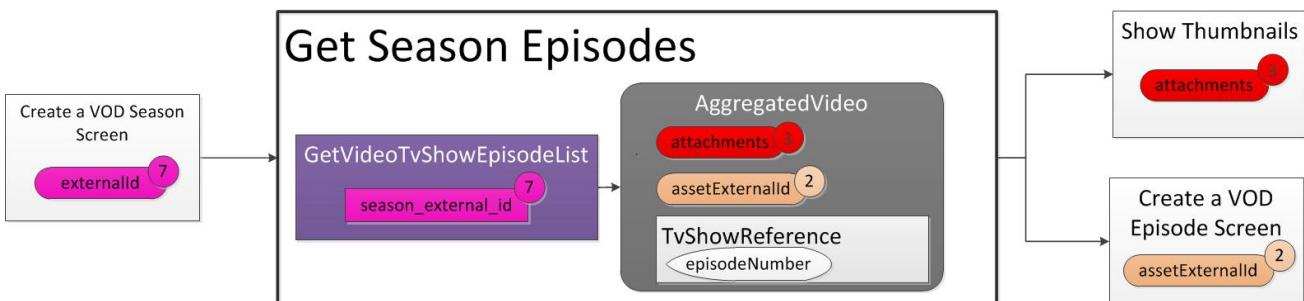
23 Get Episodes in a TVOD Season

23.1 Overview

Each TV Show season includes one or more episodes. For example, the screen below shows that episodes 1 and 2 of season 7 of the series Weeds are available.



23.2 Requests and Responses



Request: [GetVideoTvShowEpisodeList](#) (from RiGHTv Front-end API)

The list of episodes returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<code>season_external_id</code>	The ID of the season for which to find episodes.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [AggregatedVideo](#) (from RiGHTv Front-end API)

The returned entity includes various properties. For displaying the seasons in a TV show series, the most relevant are:

Property	Description				
<code>assetExternalId</code>	The external ID of the episode asset.				
<code>attachments</code>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.				
<code>tvShowReference</code>	Includes the following information on the episode: <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><code>episodeNumber</code></td><td>The number of the episode in the season. This can be used as the name of the episode in the episode list.</td></tr> </tbody> </table>	Property	Description	<code>episodeNumber</code>	The number of the episode in the season. This can be used as the name of the episode in the episode list.
Property	Description				
<code>episodeNumber</code>	The number of the episode in the season. This can be used as the name of the episode in the episode list.				

23.3 Related Tutorials

To	Refer to Tutorial
Create a TV Show Season screen	61 Create a VOD Season Screen (p.179)

23.4 Example

https://<operator>.tvaas.com/RTEFacade/GetVideoTvShowEpisodeList?season_external_id=Weeds_season_7&language_code=en&client=json

This request calls for a list of episodes in season 7 of the series Weeds.

The response for one episode is shown below.

```

"prLevel": 3,
"attachments": [
    {
        "responseElementType": "Attachment",
        "assetId": "",
        "name": "VerticalImage",
        "assetName": "",
        "value": "/attachments/weeds_ver(8).jpg"
    },
    {
        "responseElementType": "Attachment",
        "assetId": 100370,
        "name": "Trailer",
        "assetName": "Trailer",
        "value": "true"
    },
    {
        "responseElementType": "Attachment",
        "assetId": "",
        "name": "HorizontalImage",
        "assetName": "",
        "value": "/attachments/weeds-horizontal(3).jpg"
    }
],
"year": 2011,
"chapters": [],
"externalChannelId": "",
"description": "After doing three years hard time, Nancy (MARY-LOUISE PARKER) is abruptly released from Federal Prison and transferred to a halfway house in New York City. Meanwhile, Andy (JUSTIN KIRK), Silas (HUNTER PARRISH), Shane (ALEXANDER GOULD) and Doug (KEVIN NEALON), who have all spent the last three years in Copenhagen, are informed by Jill Price-Gray (JENNIFER JASON LEIGH) of Nancy's release."

```



```
"uniqueVideos": [
    {
        "isSecured": false,
        "attachments": [],
        "encodings": [
            {
                "responseElementType": "Encoding",
                "name": "HLS"
            }
        ],
        "externalId": "WeedsS7Ep1",
        "responseElementType": "UniqueVideoInfo",
        "simultaneousViewsLimit": "",
        "removalDate": "",
        "extrafields": [
            {
                "responseElementType": "Extrafield",
                "name": "search",
                "value": "1"
            }
        ],
        "pricingMatrixId": 4024,
        "definition": "",
        "securityGroups": [],
        "discountId": "",
        "status": 2
    }
],
"tvShowReference": {
    "seriesExternalId": "Weeds",
    "seasonExternalId": "Weeds_season_7",
    "responseElementType": "TvShowReference",
    "seriesName": "Weeds",
    "seasonName": "Weeds season 7",
    "seasonNumber": 7,
    "episodeNumber": 1
},
```

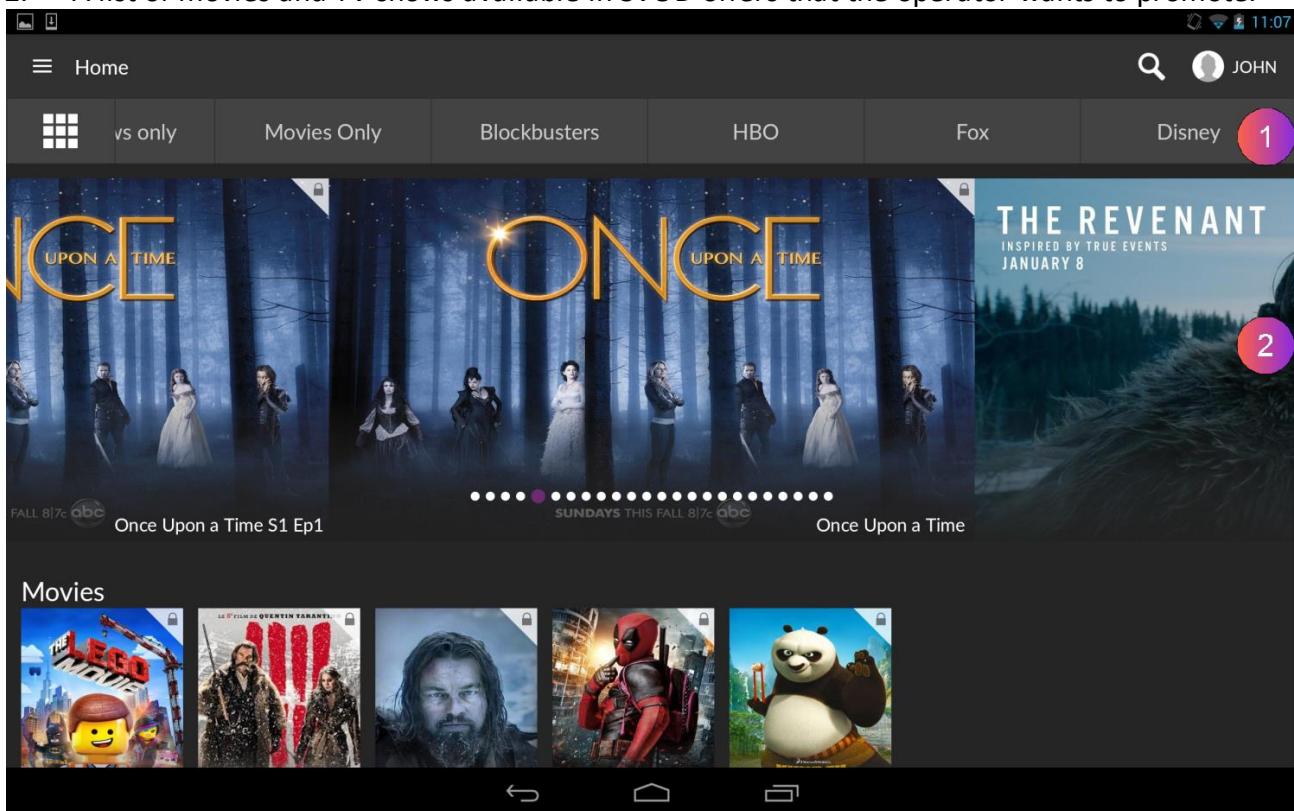


24 Create an SVOD Catalog Screen

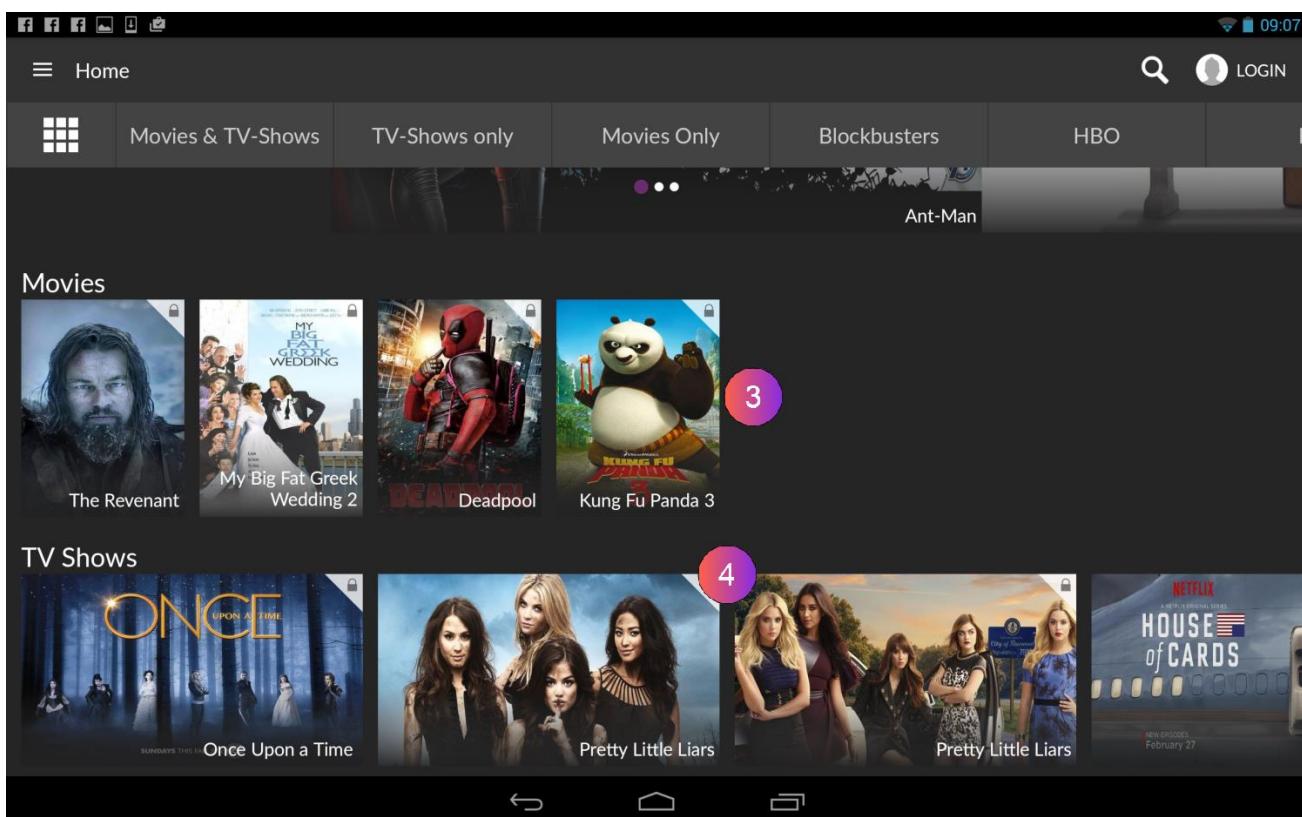
24.1 Screen Overview

The SVOD Catalog screen should showcase the SVOD offers. Such a screen should include:

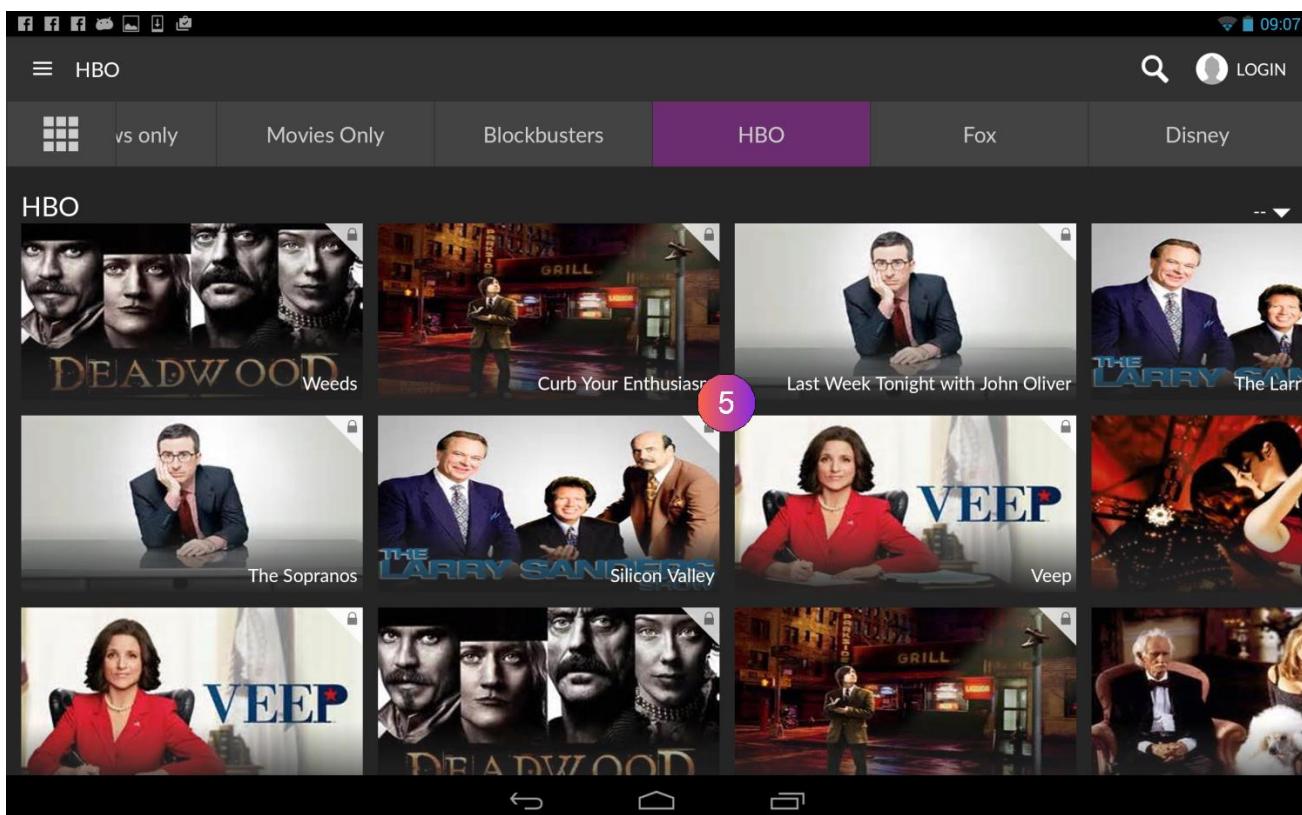
1. A list of available SVOD services.
2. A list of movies and TV shows available in SVOD offers that the operator wants to promote.



3. A list of recommended movies available in SVOD offers.
4. A list of recommended TV shows available in SVOD offers.



5. A list of all contents in a selected SVOD offer.



24.2 Related Tutorials

To	Refer to Tutorial
Get a list of SVOD movie recommendations	25 Get List of SVOD Recommendations (p.78)
Get a list of SVOD services	26 Get SVOD Services (p.80)
Get a list of SVOD movies per SVOD service	27 Get SVOD Service Contents (p.83)
Display attachment images	149 Display Images (p.402)

25 Get List of SVOD Recommendations

25.1 Overview

The client application should display:

- A list of movies and TV shows available in SVOD offers that the operator wants to promote
- A list of recommended movies available in SVOD offers

A list of recommended TV shows available in SVOD offers

25.2 Requests and Responses

The *GetVideoRecommendationList* call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.

25.3 Recommendation Requests

The table below provides best-practice examples of recommendation calls for this screen.

Recommendation List	Blend	Categories
Promotions	promotions	SVOD Services
<pre>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=promotions&random=false&subscription=false&categories=SVOD+Services&max_results=5&max_pr_level=9998&client=json&language_code=en</pre>		
Recommended Movies	movies	SVOD Services
<pre>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=movies&random=false&subscription=false&categories=SVOD+Services&max_results=10&max_pr_level=9998&client=json&language_code=en</pre>		
Recommended TV Shows	series	SVOD Services
<pre>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?content_type=Season&blend=series&random=false&subscription=false&categories=SVOD+Services&max_results=10&max_pr_level=9998&client=json&language_code=en</pre>		



25.4 Related Tutorials

To	Refer to Tutorial
Create an SVOD catalog screen	24 Create an SVOD Catalog Screen (p.75)
Display attachment images	149 Display Images (p.402)

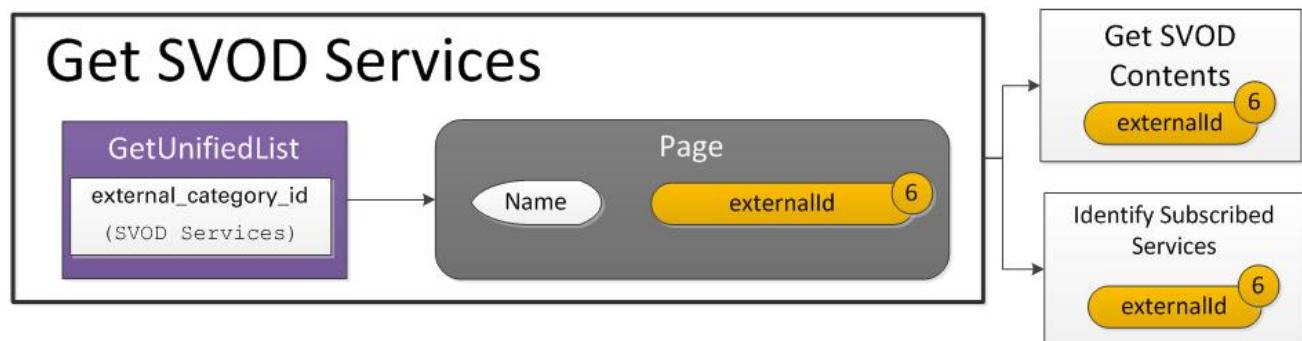
26 Get SVOD Services

26.1 Overview

Users should be shown a list of all available SVOD services. Separate tabs can be shown for each SVOD service.



26.2 Requests and Responses



Request: [GetUnifiedList](#) (from RiGHTv Front-end API)

The list of categories returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The <i>external_category_id</i> of the folder containing the SVOD Services. Typically will be SVOD Services.
<i>filter_empty_categories</i>	Set to True if only categories that contain content should be listed.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Page \(from RiGHTv Front-end API\)](#)

The returned entity includes various properties. For obtaining a list of movie categories, the most relevant is:

Property	Description
<i>name</i>	The name of the category (SVOD Service)
<i>externalId</i>	The external ID of the category (SVOD Service)

26.3 Related Tutorials

To	Refer to Tutorial
Create an SVOD catalog screen	24 Create an SVOD Catalog Screen (p.75)
Get a list of SVOD movies per SVOD service	27 Get SVOD Service Contents (p.83)

26.4 Example

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=SVOD%20Services&client=json

This call requests a list of all categories within the SVOD Services category.

The relevant sections of the result set for one of the services, Blockbusters, is shown below. Its name and externalId are indicated.

```
{  
    "template": "Folder",  
    "isSecured": true,  
    "attachments": [],  
    "description": "",  
    "externalId": "Blockbusters",  
    "isBlockedBrowsing": false,  
    "parentId": 222,  
    "responseElementType": "Page",  
    "name": "Blockbusters",  
    "extrafields": [  
        {  
            "responseElementType": "Extrafield",  
            "name": "exclude",  
            "value": ""  
        }  
    ],  
    "securityGroups": [  
        {  
            "responseElementType": "SecurityGroup",  
            "externalId": "777",  
            "type": "Service"  
        }  
    ],  
    "id": 900,  
    "status": 2  
},
```



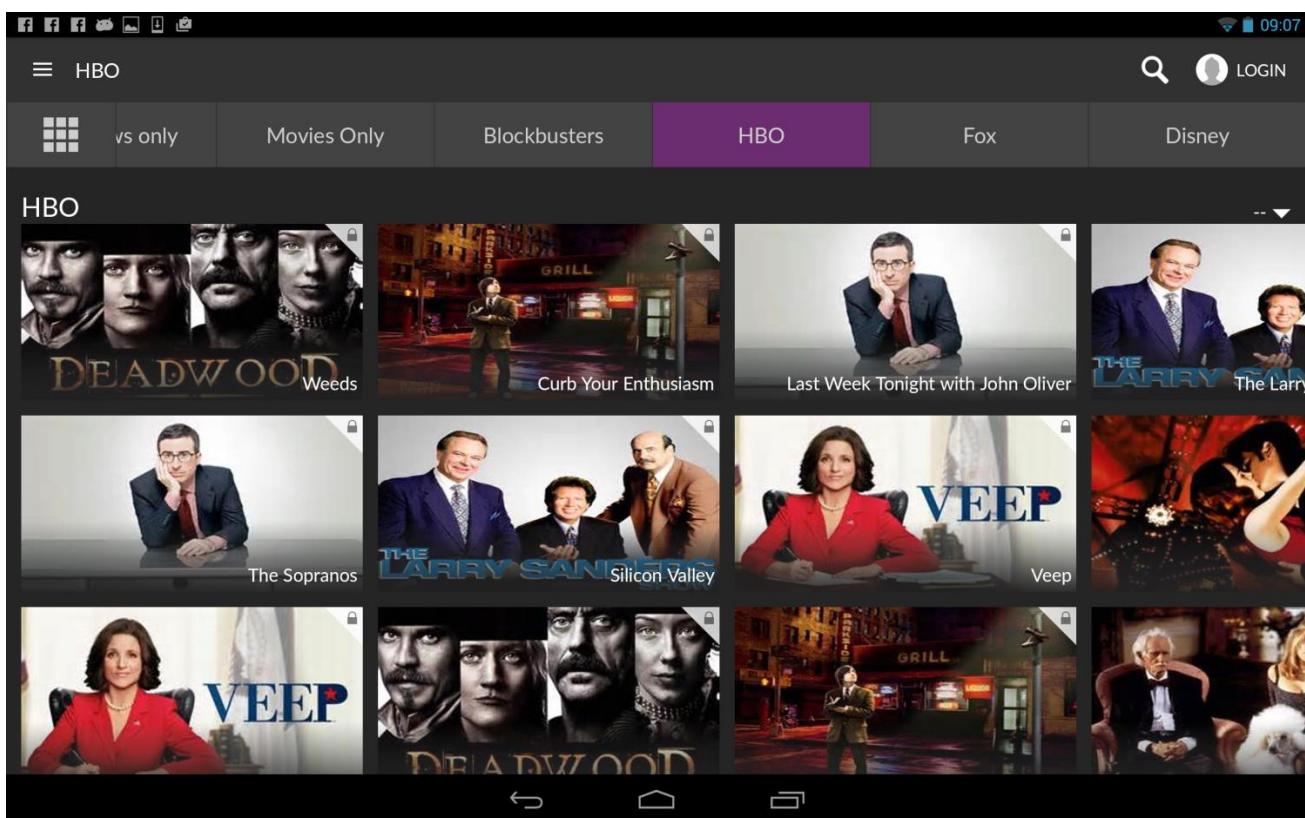
27 Get SVOD Service Contents

27.1 Overview

The SVOD Movies Catalog should list all contents available within a selected SVOD service.

An SVOD catalog can include the following types of contents, and all should be listed:

- Movie: An individual movie
- Series: A series of TV shows
- Season: A season of a TV show series

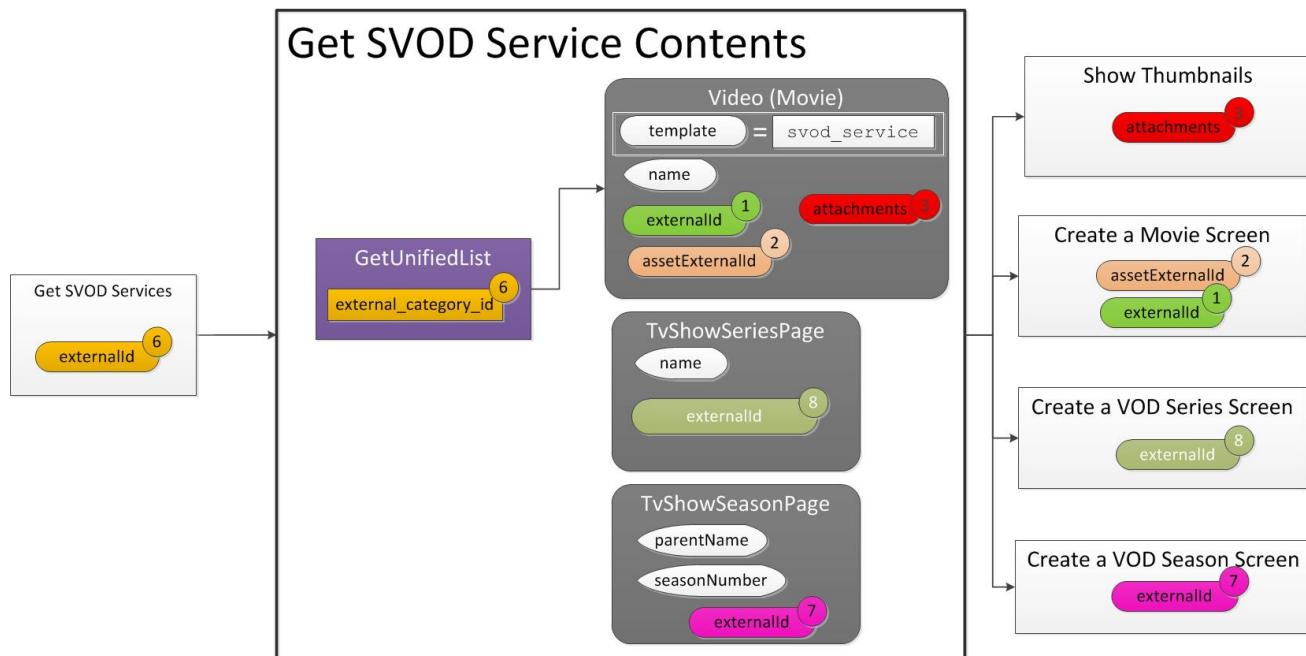


27.2 Requests and Responses

As for the VOD Movies Catalog screen, the *GetUnifiedList* request is used to obtain lists of contents in an SVOD service. It is important to note that VOD movies and SVOD contents are arranged differently in the system:

- VOD movies are arranged in categories by genre, so there may be a category with external ID Drama, and another with an external ID of Action.
- SVOD contents are arranged in categories by SVOD service, so there may be a category with external ID HBO, and another with an external ID of Disney.

Thus, to get a list of contents within an SVOD service, the *external_category_id* must be set to the ID of the SVOD service. To create a list of contents in an SVOD service of a particular genre, the response will have to be filtered by genre by the client application. In other cases, genre subcategories may be present within each SVOD service.



Request: [GetUnifiedList](#) (from RiGHTv Front-end API)

The list of movies returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The ID of the folder whose contents you want listed. See the examples.
<i>order</i>	Specify the <i>order</i> in which the results should be listed, for example by <code>name</code> for alphabetical or <code>tree</code> for the same order as listed by the operator.

For more parameters, see the RiGHTv Front-end API Developers Guide.

27.2.1 Responses

The response can include entities for videos (movies), series and seasons. Identify each response by the template value.

Property	Description
<i>template</i>	Identifies each returned content as a series (<code>TV_Shows_Series</code>), a season (<code>TV_Shows_Season</code>), or as a movie (<code>svod_service</code>).

Response: Video (from RiGHTv Front-end API)

For movies, the following additional properties are returned:

Property	Description
<i>assetExternalId</i>	The ID of the video asset in the Content Library
<i>externalId</i>	The ID of the movie page in the VOD Catalog.
<i>name</i>	The name of the movie
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>genreEntityList</i>	An array that includes all genres assigned to a movie, named in the <i>externalId</i> property.

Response: TvShowSeriesPage (from RiGHTv Front-end API)

The *TvShowSeriesPage* response includes the *TvShowSeries* entity which describes the TV Show (series). The returned entity includes various properties. For displaying the TV Shows (series) listed in a category, the most relevant are:

Property	Description
<i>name</i>	The name of the TV show.
<i>externalId</i>	The ID of the TV show in the VOD Catalog.
<i>attachments</i>	The poster images for the TV Show. See 149 Display Images (p.402) to learn how to retrieve the image files.

Response: TvShowSeasonPage (from RiGHTv Front-end API)

The *TvShowSeasonPage* response includes the *TvShowSeason* entity which describes a TV Show season. The returned entity includes various properties. For displaying the seasons listed in a category, the most relevant are:



Property	Description
<i>externalId</i>	The ID of the season in the VOD Catalog.
<i>attachments</i>	The poster images for the TV Show. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>parentName</i>	The name of the TV Show (series) to which the season belongs.
<i>seasonNumber</i>	<p>The number of the season in the series.</p> <p>The name displayed should be constructed as follows:</p> <p><i>parentName Season: seasonNumber</i></p> <p>For example: Weeds: Season 7</p>

27.3 Related Tutorials

To	Refer to Tutorial
Create an SVOD catalog screen	24 Create an SVOD Catalog Screen (p.75)
Display attachment images	149 Display Images (p.402)
Create a TV Show Screen	60 Create a TVOD TV Show Series Screen (p.175)
Create a Season Screen	61 Create a VOD Season Screen (p.179)

27.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=HBO&client=json`

This request calls for a list of content items in the HBO service.

Sample responses for a movie, series and season are shown below.

27.4.1 Example - Movie

A sample response for a movie is shown below. Note that the *template* is *svod_service*.



```
"metadata": [],
"prLevel": 3,
"keywords": "",
"year": 1999,
"seriesNumberOfEpisodes": "",
"tvShowReference": {},
"episodeId": "",
"type": "Movie",
"assetExternalId": "Girl Interrupted",
"contentProvider": "Default",
"reviews": [
{
    "reviewerOrganization": "IMDB",
    "responseElementType": "Review",
    "reviewerLastName": "",
    "reviewerRating": 7.3,
    "reviewerFirstName": "IMDB",
    "reviewText": ""
}
],
"id": 1704,
"rentalPeriodUnit": "",
"isSecured": true,
>windowStart": 1448981367852,
"adsInfo": "",
"seriesName": "",
"prName": "TV-14",
"rentalPeriod": "",
"name": "Girl, Interrupted",
"broadcastTime": 0,
"shortName": "",
"discountId": "",
"status": 2,
"advisories": "",
"template": "svod_service", 
```

```
"attachments": [
  {
    "responseElementType": "Attachment",
    "assetId": "",
    "name": "VerticalImage",
    "assetName": "",
    "value": "/attachments/Girl_InterruptedVerticalImage
      .jpg"
  }
],
"chapters": [],
"externalChannelId": "",
"reviewerRating": 7.3,
"flags": 0,
"description": "Based on writer Susanna Kaysen's account of
  her 18-month stay at a mental hospital in the 1960s.",
"seriesSeason": "",
"allowedTerminalCategories": [
  {
    "responseElementType": "TerminalCategory",
    "maxTerminalsOfNonOperator": -1,
    "maxTerminals": -1,
    "name": "PC",
    "externalId": "PC"
  },
],
"duration": 3600000,
"genreEntityList": [
  {
    "responseElementType": "Genre",
    "parentName": "",
    "name": "Drama",
    "externalId": "Drama",
    "id": 5057
  },
],
"responseElementType": "Video",
"plannedPublishDate": "",
```

```
"simultaneousViewsLimit": "",  
"assetId": 35012,  
"genres": [  
    5057,  
    12269  
,  
"pricingMatrixId": 4024,  
"definition": "",  
"windowEnd": 0,  
"encodings": [  
    {  
        "responseElementType": "Encoding",  
        "name": "HLS"  
    }  
,  
"externalId": "Girl Interrupted",  
"removalDate": "",  
"awards": [],  
"extrafields": [],  
"securityGroups": [  
    {  
        "responseElementType": "SecurityGroup",  
        "externalId": "HBO",  
        "type": "Service"  
    },  
    {  
        "responseElementType": "SecurityGroup",  
        "externalId": "sport7day",  
        "type": "Service"  
    }  
,  
"contentProviderExternalId": "Default",  
"categoryId": 229  
,  
r
```

27.4.2 Example - Series

A sample response for a season is shown below. Note that the template is *TV_Shows_Series*.



```
"template": "TV_Shows_Series", ←  
"isSecured": true,  
"attachments": [],  
"tvShowSeries": {  
    "isMiniSeries": false,  
    "studio": "",  
    "prLevel": 0,  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "VerticalImage",  
            "assetName": "",  
            "value": "/attachments  
/MV5BMTgxNDMwMjQ5N15BM15BanBnXkFtZTgwMjM4MTAzMzI@  
.V1_UY1200_CR90,0,630,1200_AL_.jpg"  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "HorizontalImage",  
            "assetName": "",  
            "value": "/attachments/Curb-Your-Enthusiasm-casting  
-call-open-audition.jpg"  
        }  
    ],  
    "year": 2000,  
    "prName": "NR",  
    "externalId": "Curb_Your_Enthusiasm",  
    "description": "\"Seinfeld\" co-creator Larry David plays  
a version of himself on the improvised series. He faces  
a constant barrage of life's little annoyances, which in  
David's sometimes well-meaning but terminally fumbling  
hands don't tend to stay small for very long.",  
    "shortDescription": "",  
    "countries": [],  
    "contentProviderName": "",  
    "numberOfSeasons": 2,
```

```
"responseElementType": "TvShowSeries",
"genres": [],
"name": "Curb Your Enthusiasm",
"extrafields": [],
"parentalAdvisories": "",
"contributors": [],
"shortName": "",
"contentProviderExternalId": ""

},
"externalId": "CurbYourEnthusiasm",
"description": "",
"isBlockedBrowsing": false,
"parentId": 230,
"responseElementType": "TvShowSeriesPage",
"name": "Curb Your Enthusiasm",
"extrafields": [],
"securityGroups": [
{
  "responseElementType": "SecurityGroup",
  "externalId": "HBO",
  "type": "Service"
},
"id": 5940,
"status": 2
```

Example: Season

A sample response for a season is shown below. Note that the *template* is *TV_Shows_Season*.



```
{  
    "template": "TV_Shows_Season", ←  
    "isSecured": true,  
    "attachments": [],  
    "externalId": "368354354",  
    "description": "",  
    "isBlockedBrowsing": false,  
    "parentId": 230,  
    "tvShowSeason": {  
        "studio": "",  
        "prLevel": 0,  
        "attachments": [  
            {  
                "responseElementType": "Attachment",  
                "assetId": "",  
                "name": "VerticalImage",  
                "assetName": "",  
                "value": "/attachments/download (1).jpg"  
            },  
            {  
                "responseElementType": "Attachment",  
                "assetId": "",  
                "name": "HorizontalImage",  
                "assetName": "",  
                "value": "/attachments/download.jpg"  
            }  
        ],  
        "year": 2016,  
        "seriesName": "The big bang theory",  
        "prName": "NR",  
        "externalId": "The_big_bang_theory_S8",  
        "description": "The eighth season of the American sitcom  
        The Big Bang Theory first aired on CBS with a one-hour  
        premiere on Monday, September 22, 2014. It returned to  
        its previous Thursday time slot on October 30 for the  
        season's seventh episode.",  
        "shortDescription": "",  
        "seasonNumber": 8,  
    }  
}
```

```
"countries": [],
"seriesExternalId": "The_big_bang_theory",
"contentProviderName": "Default",
"numberOfEpisodes": 8,
"responseElementType": "TvShowSeason",
"genres": [],
"name": "The big bang theory season 8",
"extrafields": [],
"parentalAdvisories": "",
"contributors": [],
"shortName": "",
"contentProviderExternalId": "Default"
},
"responseElementType": "TvShowSeasonPage",
"name": "The big bang theory 8",
"extrafields": [],
"securityGroups": [
{
  "responseElementType": "SecurityGroup",
  "externalId": "HBO",
  "type": "Service"
},
{
  "responseElementType": "SecurityGroup",
  "externalId": "sport7day",
  "type": "Service"
}
],
"id": 6102,
"status": 2
},
```



28 Identify Subscribed Services

28.1 Overview

Although the SVOD Catalog should show the user all available services, there should be an indication of which services are currently subscribed to, or which are not. This section explains how to obtain a list of subscribed services.

28.2 Requests and Responses



Request: [GetHouseholdServiceList](#) (from RiGHTv Front-end API)

Gets a list of the account's services. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [ServiceSubscription](#) (from RiGHTv Front-end API)

The *ServiceSubscription* entity lists all services to which the account is subscribed. The following parameters are relevant:

Property	Description
externalId	The ID of the service.
<i>model</i>	The model of service. Only services of model SVOD are relevant to the SVOD Catalog screen.

28.3 Related Tutorials

To	Refer to Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Create a My Library Screen	151 Creating a My Library Screen (p.413)

28.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHouseholdServiceList?client=json>

This requests a list of all the account's services.

In the sample response below, the account can be seen to be subscribed the HBO service.

```
{  
    "responseElementType": "ServiceSubscription",  
    "subscriptionExpiration": "",  
    "externalId": "HBO",  
    "model": "SVOD",  
    "id": 1,  
    "serviceName": "HBO"  
}
```

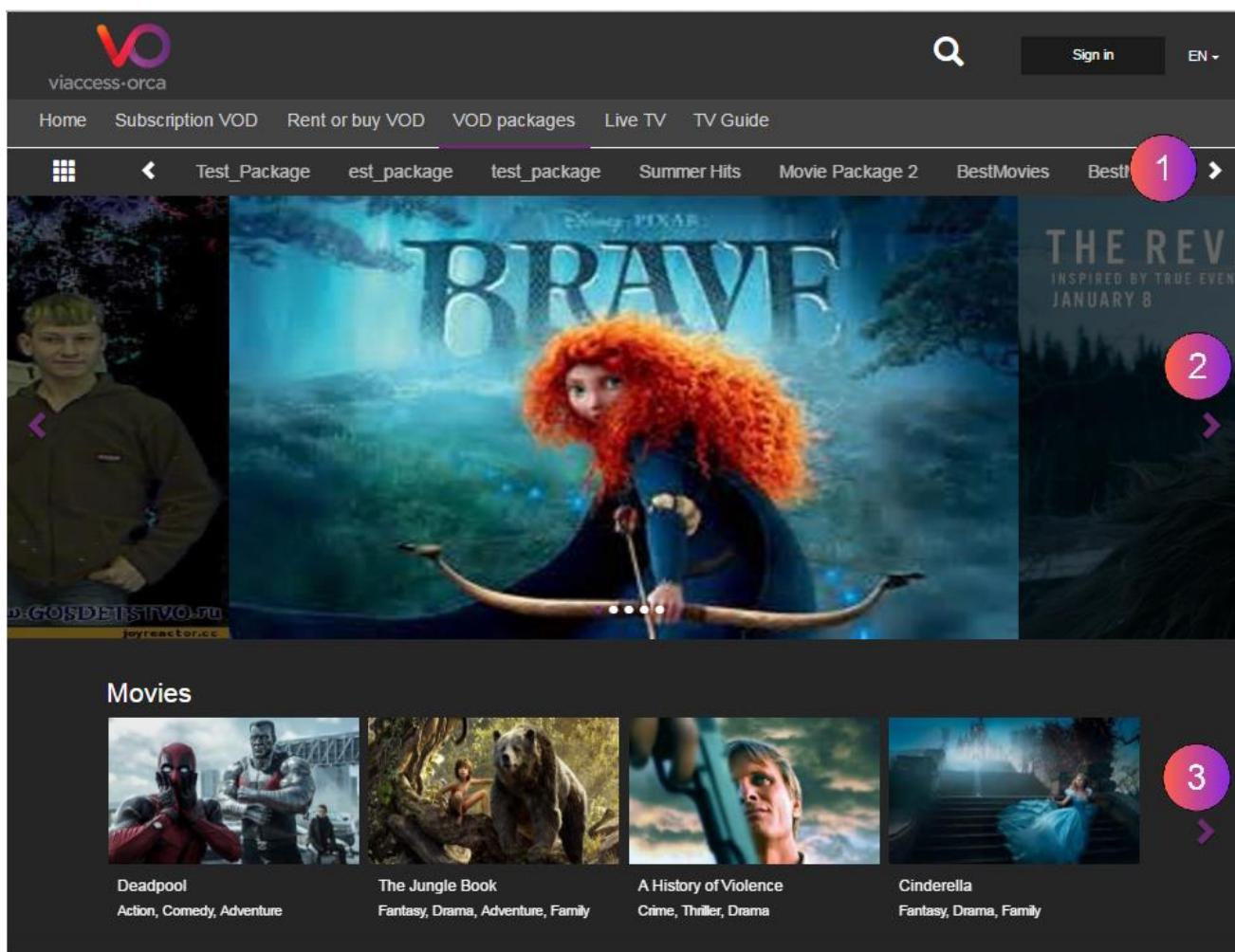
29 Create a VOD Movie Packages Catalog Screen

29.1 Screen Overview

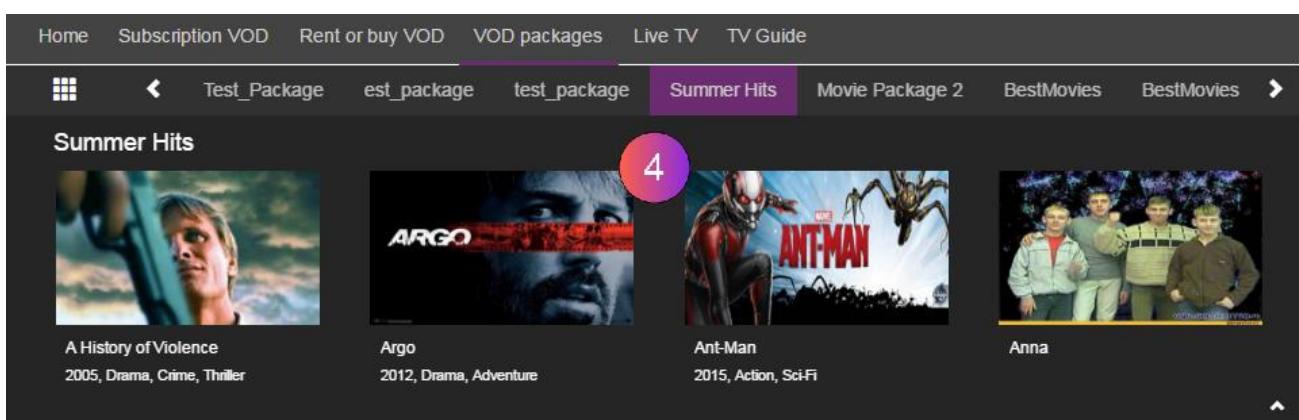
The VOD Movie Packages Catalog screen should showcase the content available through VOD Movie Package offers.

Such a screen should include:

1. A list of available VOD movie packages.
2. A list of movies available in movie package offers that the operator wants to promote.
3. A list of recommended movies available in movie package offers.



4. A list of all contents in a selected movie package.



29.2 Related Tutorials

To	Refer to Tutorial
Get a list of recommended VOD movie package movies	30 Get List of VOD Movie Package Recommendations (p.98)
Get a list of VOD Movie Packages	31 Get VOD Movie Packages (p.100)
Get a list of movies in a VOD Movie Package	32 Get VOD Package Contents (p.103)

30 Get List of VOD Movie Package Recommendations

30.1 Overview

The client application should display lists of recommendations, including:.

- A list of available in movie package offers that the operator wants to promote.
- A list of recommended movies available in movie package offers.

30.2 Requests and Responses

The *GetVideoRecommendationList* call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.

30.3 Recommendation Requests

The table below provides best-practice examples of recommendation calls for this screen.

Recommendation List	Blend	Categories
1. Promotions	promotions	VOD+Packages
	<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=promotions&categories=VOD+Packages&max_results=5&max_pr_level=5&client=json</code>	
2. Recommended Movies	movies	VOD+Packages
	<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=movies&categories=VOD+Packages&max_results=10&max_pr_level=5&client=json</code>	

Thumbnails of all recommended movies should be displayed. See [149 Display Images \(p.402\)](#).



30.4 Related Tutorials

To	Refer to Tutorial
Create a VOD Movie Package catalog screen	29 Create a VOD Movie Packages Catalog Screen (p.96)

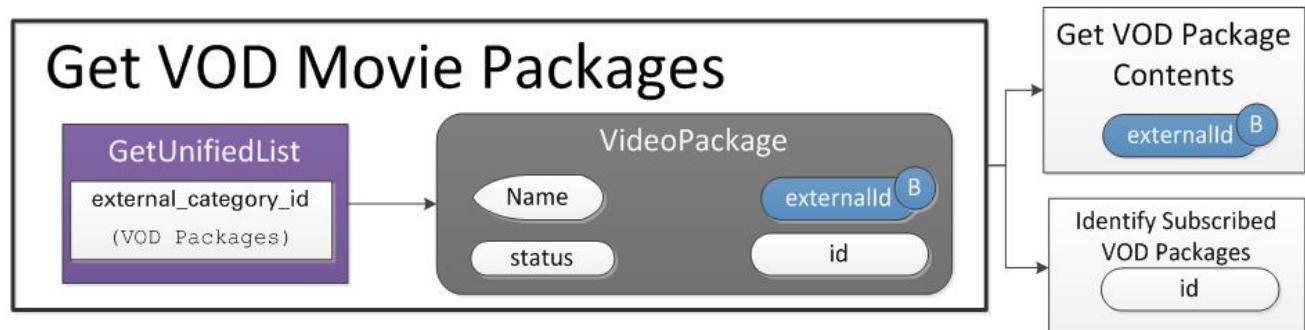
31 Get VOD Movie Packages

31.1 Overview

Typically, all VOD movie package categories are located under the VOD Packages category in the VOD catalog. The VOD Packages Catalog screen should show these categories.

Excerpt of VOD Packages Category in VOD Catalog	List of VOD Packages in Application
<p>Title</p> <p>Movies Package1</p> <p>Monday Madness</p>	<p>Movie Package 2 BestMovies BestMovies MyMovies My Favorites Jack's Movies Movies Package1</p>

31.2 Requests and Responses



The `GetUnifiedList` request gets a list of all the items in a specified category. Typically all VOD packages will be stored in a single category called *VOD Packages*.

In the response, the property `responseElementType` is shown as `VideoPackage` for all VOD packages.

Request: GetUnifiedList (from RiGHTv Front-end API)

Gets the list of all items in the specified category. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	The name of the category in which VOD packages are stored. Typically will be VOD%20Packages.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: VideoPackage (from RiGHTv Front-end API)

The returned entity includes the following properties:

Property	Description
<i>externalId</i>	<i>The external ID of the package.</i>
<i>name</i>	<i>The name of the package.</i>
<i>status</i>	Indicates whether published (2), announced (3), or removed from service (4)
<i>id</i>	The internal ID of the package (a number).

For more properties, see the RiGHTv Front-end API Developers Guide.

31.3 Related Tutorials

To	Refer to Tutorial
Create a VOD Movie Package catalog screen	29 Create a VOD Movie Packages Catalog Screen (p.96)
Get a list of movies in a VOD Movie Package	32 Get VOD Package Contents (p.103)

31.4 Example

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=VOD%20Packages&client=json

This request gets a list of the contents of the VOD Packages category, which should only include VOD movie packages.



The sample response below shows the return for the VOD package est_package. The externalId, responseTypeElement (indicating VideoPackage) and name are all indicated.

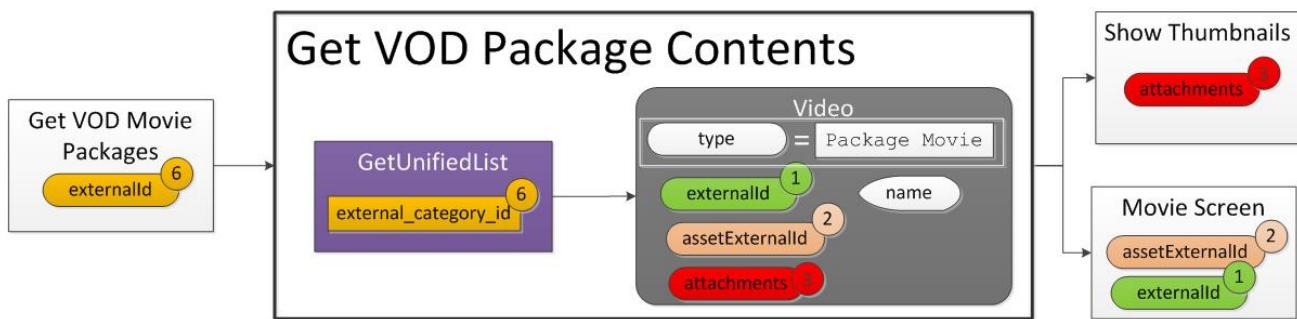
```
{  
    "attachments": [],  
    "rentalPeriod": "",  
    "externalId": "est_package", ←  
    "description": "",  
    "packagePeriod": 4,  
    "responseElementType": "VideoPackage", ←  
    "estPrice": 12,  
    "plannedPublishDate": "",  
    "removalDate": "",  
    "deliveryMode": "Converged",  
    "price": 9,  
    "name": "est_package", ←  
    "extrafields": [],  
    "definition": "",  
    "id": 12103,  
    "packagePeriodUnit": 4,  
    "rentalPeriodUnit": "",  
    "status": 2  
},
```

32 Get VOD Package Contents

32.1 Overview

The VOD Package catalog in the application should show the movies within a package.

32.2 Requests and Responses



Request: [GetUnifiedList](#) (from RiGHTv Front-end API)

The list of movies returned can be customized by parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>external_category_id</i>	Specify the ID of the VOD packages folder whose contents you want listed. Corresponds to <i>externalId</i> returned within <i>VideoPackage</i> . See the example.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Video](#) (from RiGHTv Front-end API)

The returned entity includes various properties. For displaying the movies listed in a category, the most relevant are:

Property	Description
<i>name</i>	The name of the movie.
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>assetExternalId</i>	The ID of the video asset in the Content Library

<i>externalId</i>	The ID of the movie page in the VOD Catalog.
<i>type</i>	Type of Package Movie identifies the item as a movie package movie.

32.3 Related Tutorials

To	Refer to Tutorial
Create a VOD Movie Package catalog screen	29 Create a VOD Movie Packages Catalog Screen (p.96)
Get a list of VOD Movie Packages	31 Get VOD Movie Packages (p.100)
Display attachment images	149 Display Images (p.402)

32.4 Example

In the example in [31 Get VOD Movie Packages \(p.100\)](#), the sample response for the `est_package` package was shown. The *externalId* for that package is `est_package`.

```
{
  "attachments": [],
  "rentalPeriod": "",
  "externalId": "est_package", ←
  "description": "",
  "packagePeriod": 4,
  "responseElementType": "VideoPackage", ←
  "estPrice": 12,
  "plannedPublishDate": "",
  "removalDate": "",
  "deliveryMode": "Converged",
  "price": 9,
  "name": "est_package", ←
  "extrafields": [],
  "definition": "",
  "id": 12103,
  "packagePeriodUnit": 4,
  "rentalPeriodUnit": "",
  "status": 2
},
```

This request calls for a list of movies in the movie package.

https://<operator>.tvaas.com/RTEFacade/GetUnifiedList?external_category_id=est_package&client=json



The sample response below shows the return for one movie within the est_package. The *name*, *externalId*, *type* and *attachments* array are indicated.

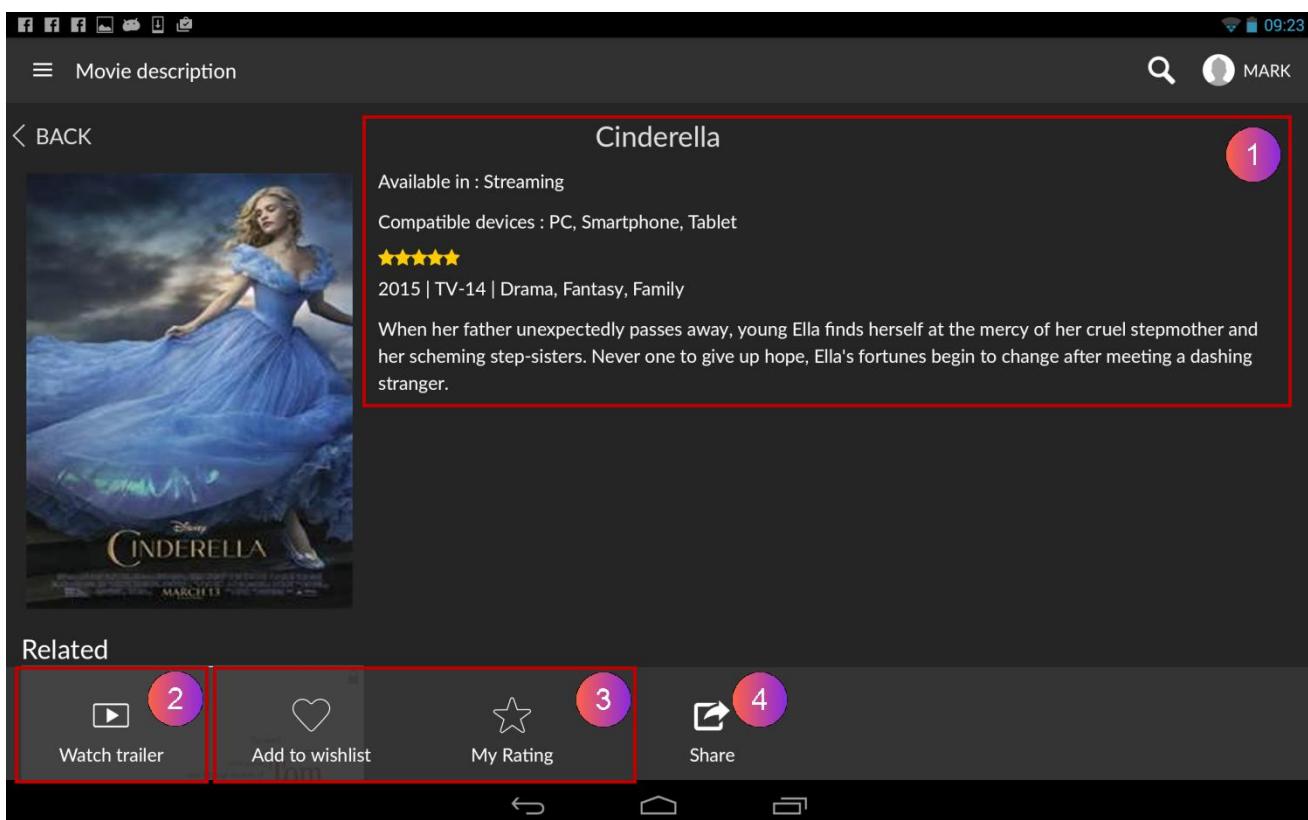
```
{  
    "metadata": [],  
    "prLevel": 4,  
    "keywords": "",  
    "year": 1999,  
    "seriesNumberOfEpisodes": "",  
    "episodeId": "",  
    "type": "Package Movie", ←  
    "assetExternalId": "The Matrix", ←  
    "contentProvider": "Default",  
    "reviews": [ ],  
    "id": 12105, ←  
    "rentalPeriodUnit": "",  
    "isSecured": false,  
    "windowStart": 1441267794913,  
    "adsInfo": "",  
    "seriesName": "",  
    "prName": "R",  
    "rentalPeriod": "",  
    "name": "The Matrix", ←  
    "broadcastTime": 0,  
    "shortName": "",  
    "discountId": "",  
    "status": 2,  
    "advisories": "",  
    "template": "vod package movie",  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "VerticalImage",  
            "assetName": "",  
            "value": "/attachments/The_Matrix_vertical(1).jpg"  
        },  
    ]  
}
```

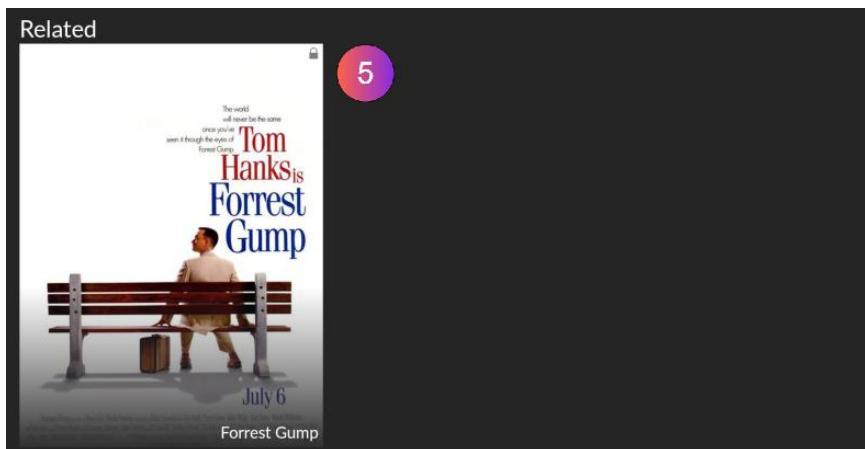
33 Create a Movie Screen

33.1 Overview

Users can be shown lists of available movies in a number of different screens, such as a Catalog screen or a My Library screen. The user should be able to click any of the movies listed and be shown a screen that displays relevant information about the movie, and provides a list of possible actions to take. For example:

1. Movie information including:
 - Metadata about the movie itself, such as its name, poster, the year and country of production, its genre(s), cast names, and description.
 - The list of offers in which the movie can be purchased.
 - User ratings of the movie.
 - Whether the movie is available for download
2. Movie and Trailer Play buttons
3. A list of additional actions that users can take, such as adding the movie to a wish list and rating the movie.
4. A Share button for sharing over social media (or individual buttons for each social media network).
5. A list of recommended movies.





33.2 Related Tutorials

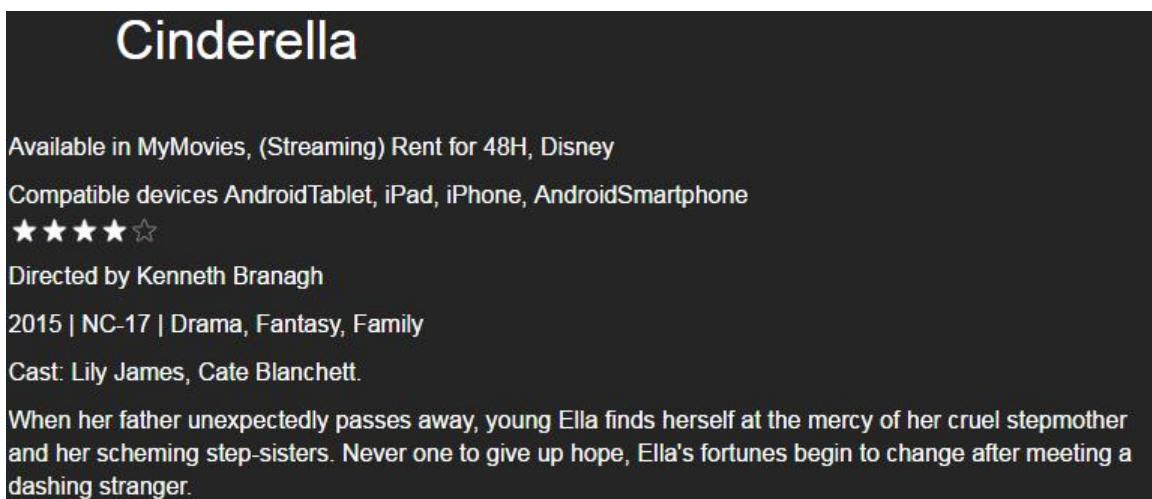
To	Refer to Tutorial
Get most of the movie metadata needed	34 Get Movie Metadata (p.108)
Get user rating information	42 Get User Ratings (p.135)
Get a list of compatible devices for the movie	35 List Compatible Devices (p.113)
List offers through which the movie can be purchased	35 List Compatible Devices (p.113)
Get a list of recommended movies (related to this one)	58 Get Recommended Movies (p.171)
Play the trailer	43 Playing a Trailer (p.137)
Play the movie	45 Playing a Movie (p.141)
Rate the movie	57 Rate a Movie (p.169)
Add the movie to the wishlist (or remove it)	53 Manage a Wish List (p.160)
Share the movie via Facebook	194 Sharing Via Facebook (p.524)
Create an episode screen	59 Create an Episode Screen (p.174)
Determine whether a movie is available for download	Identify if Movie Available for Download

34 Get Movie Metadata

34.1 Overview

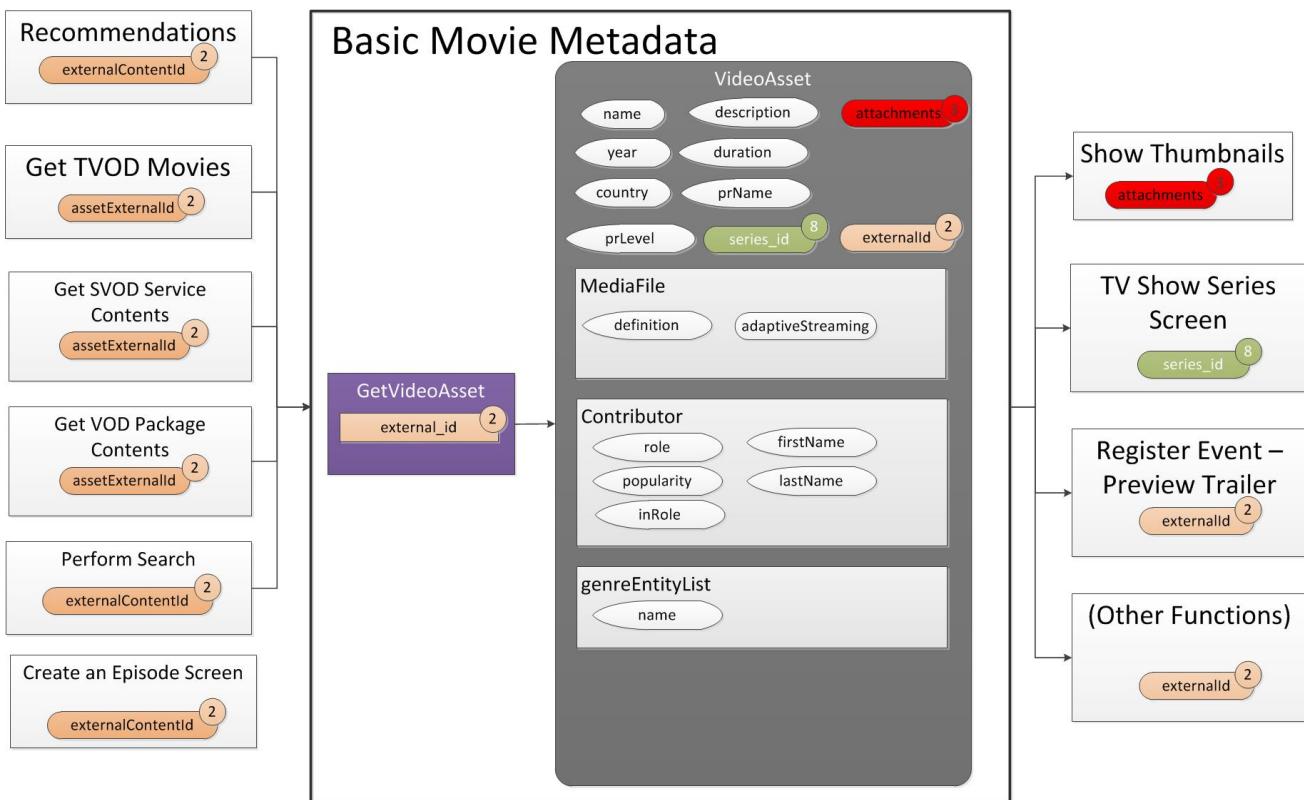
The Movie screen should include the following basic metadata:

1. Information about the movie itself, such as its poster, name, the year and country of production, and description.
2. Cast information
3. Compatible devices (see [35 List Compatible Devices \(p.113\)](#))



A dark-themed movie metadata card for the film "Cinderella". The title "Cinderella" is at the top in white. Below it is a summary: "Available in MyMovies, (Streaming) Rent for 48H, Disney". It lists compatible devices: "Compatible devices AndroidTablet, iPad, iPhone, AndroidSmartphone". A rating of four stars is shown. The director "Directed by Kenneth Branagh" and the release year "2015 | NC-17 | Drama, Fantasy, Family" are listed. The cast includes "Cast: Lily James, Cate Blanchett". A brief plot summary follows: "When her father unexpectedly passes away, young Ella finds herself at the mercy of her cruel stepmother and her scheming step-sisters. Never one to give up hope, Ella's fortunes begin to change after meeting a dashing stranger."

34.2 Requests and Responses



Request: [GetVideoAsset](#) (from RiGHTv Front-end API)

This request gets all metadata on the relevant video, specified by the `external_id` parameter.

Parameter	Description
<code>external_id</code>	The external ID of the video in the Content Library.

For more parameters, see the RiGHTv Front-end Integration Guide.

Response: [VideoAsset](#) (from RiGHTv Front-end API)

The returned entity includes a long list of properties, including:

Parameter	Description
<code>attachments</code>	The <code>name</code> and <code>value</code> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<code>name</code>	The name of the movie.
<code>description</code>	A description of the movie.

Parameter	Description												
<i>duration</i>	Duration of the movie.												
<i>year</i>	The year in which the movie was produced.												
<i>genreEntityList</i>	Lists the names of all genres assigned to the movie.												
<i>prlevel</i>	The parental rating level of the movie.												
<i>externalId</i>	The external ID of the video in the Content Library.												
<i>Contributors</i>	A list of contributors to the movie. <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>role</i></td><td>Actor, Director, etc.</td></tr> <tr> <td><i>popularity</i></td><td>The numeric value indicating the popularity of the contributor in the context of the specific content item.</td></tr> <tr> <td><i>firstName</i></td><td>Contributor first name</td></tr> <tr> <td><i>lastName</i></td><td>Contributor last name</td></tr> <tr> <td><i>inRole</i></td><td>Information about the contributor's role in the movie</td></tr> </tbody> </table>	Property	Description	<i>role</i>	Actor, Director, etc.	<i>popularity</i>	The numeric value indicating the popularity of the contributor in the context of the specific content item.	<i>firstName</i>	Contributor first name	<i>lastName</i>	Contributor last name	<i>inRole</i>	Information about the contributor's role in the movie
Property	Description												
<i>role</i>	Actor, Director, etc.												
<i>popularity</i>	The numeric value indicating the popularity of the contributor in the context of the specific content item.												
<i>firstName</i>	Contributor first name												
<i>lastName</i>	Contributor last name												
<i>inRole</i>	Information about the contributor's role in the movie												
<i>MediaFile</i>	<table border="1"> <tr> <td><i>definition</i></td><td>Media file definition (HD, SD, 3D)</td></tr> </table>	<i>definition</i>	Media file definition (HD, SD, 3D)										
<i>definition</i>	Media file definition (HD, SD, 3D)												
<i>tvShowReference</i>	A list of TV Show properties that are relevant to TV Show episodes. <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>episodeNumber</i></td><td>The number of the episode in the season.</td></tr> </tbody> </table>	Property	Description	<i>episodeNumber</i>	The number of the episode in the season.								
Property	Description												
<i>episodeNumber</i>	The number of the episode in the season.												



34.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Display attachment images	149 Display Images (p.402)
Create an Episode screen	59 Create an Episode Screen (p.174)

34.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetVideoAsset?external_id=Argo&language_code=es&client=json`

This request calls for metadata on the movie Argo.

The relevant sections of the result set are shown below.

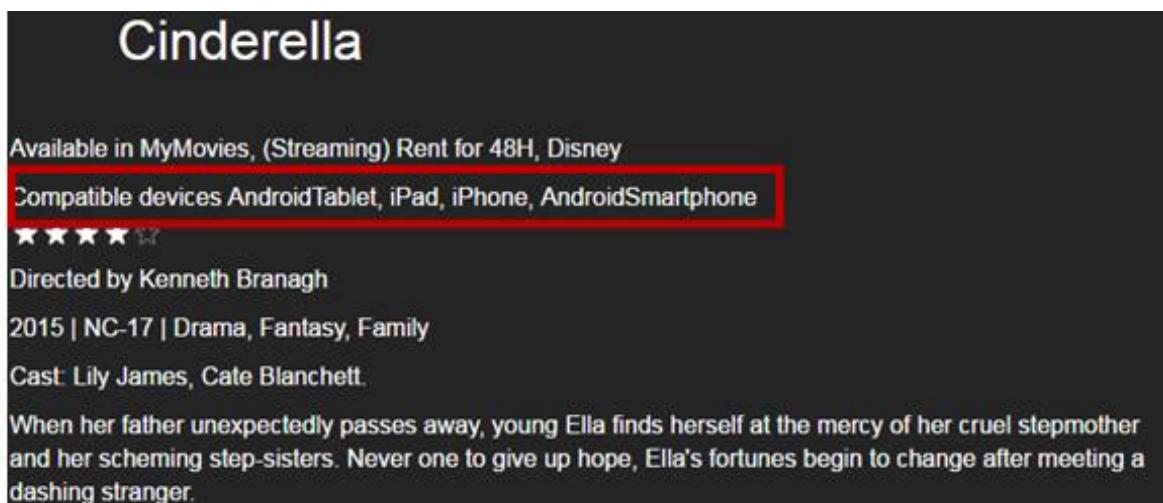
<pre> "attachments": [{ "responseElementType": "Attachment", "assetId": "", "name": "VerticalImage", "assetName": "", "value": "/attachments/argo_vertical_after_edit.jpg" }] </pre>	<pre> "mediaFiles": [{ "verticalSize": 0, "subtitles": [], "externalId": "Argo_EtsMssClrS3", "adaptiveStreaming": "Smooth Streaming", "bitrate": 0, "responseElementType": "MediaFile", "encryption": "", "deliveryMode": "Streaming", "horizontalSize": 0, "dubbing": [], "extrafields": [], "definition": "", "ratio": "" }] </pre>
<pre> "description": "Acting under the cover of a Hollywood producer scouting a location for a science fiction film, a CIA agent launches a dangerous operation to rescue six Americans in Tehran during the U.S. hostage crisis in Iran in 1980. (es)", </pre>	
<pre> "contributors": [{ "responseElementType": "Contributor", "firstName": "Chris", "lastName": "Terrio", "role": "Writer", "popularity": 0, "inRole": "" }, { "responseElementType": "Contributor", "firstName": "Tony", "lastName": "Mendez", "role": "Writer", "popularity": 0, "inRole": "" }] </pre>	<pre> "genreEntityList": [{ "responseElementType": "Genre", "parentName": "", "name": "Drama", "externalId": "Drama", "id": 5057 }, { "responseElementType": "Genre", "parentName": "", "name": "Adventure", "externalId": "Adventure", "id": 14493 }] </pre>
<pre> "year": 2012, </pre>	<pre> "duration": 7200000, </pre>
<pre> "externalId": "Argo", </pre>	<pre> "prLevel": 9999, </pre>
<pre> "name": "Argo", </pre>	



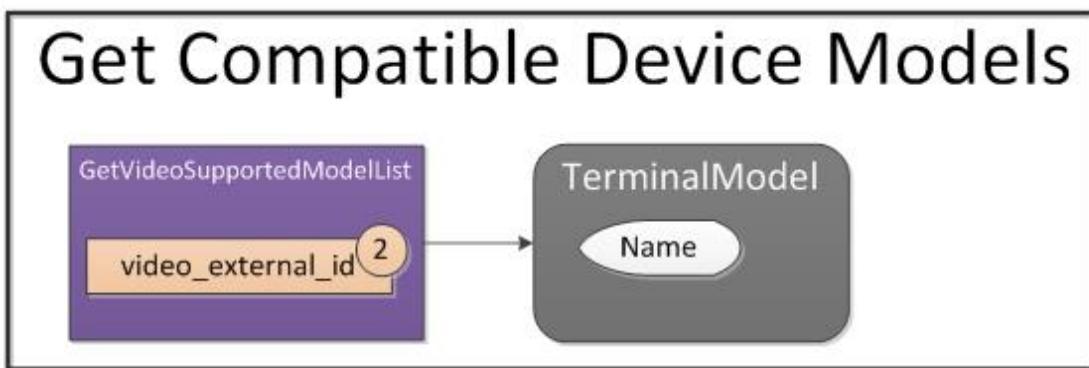
35 List Compatible Devices

35.1 Overview

The Movie Screen must show a list of device models on which the movie can be played.



35.2 Requests and Responses



Request: [GetVideoSupportedModelList](#) (from RiGHTv Front-end API)

This request gets a list of all device models on which the selected movie may be played.

Parameters	Description
video_external_id	The external ID of the video in the Content Library.

Response: TerminalModel (from RiGHTv Front-end API)

The returned entity includes the following properties for each supported model:

Property	Description
name	The name of the device model.

35.3 Related Tutorials

To	Refer to Tutorial
Create a Movie Screen	33 Create a Movie Screen (p.106)

35.4 Example

https://<operator>.tvaas.com/RTEFacade/GetVideoSupportedModelList?video_external_id=Cinderella&client=json

This request calls for a list of all the device models on which the movie Cinderella may be viewed.

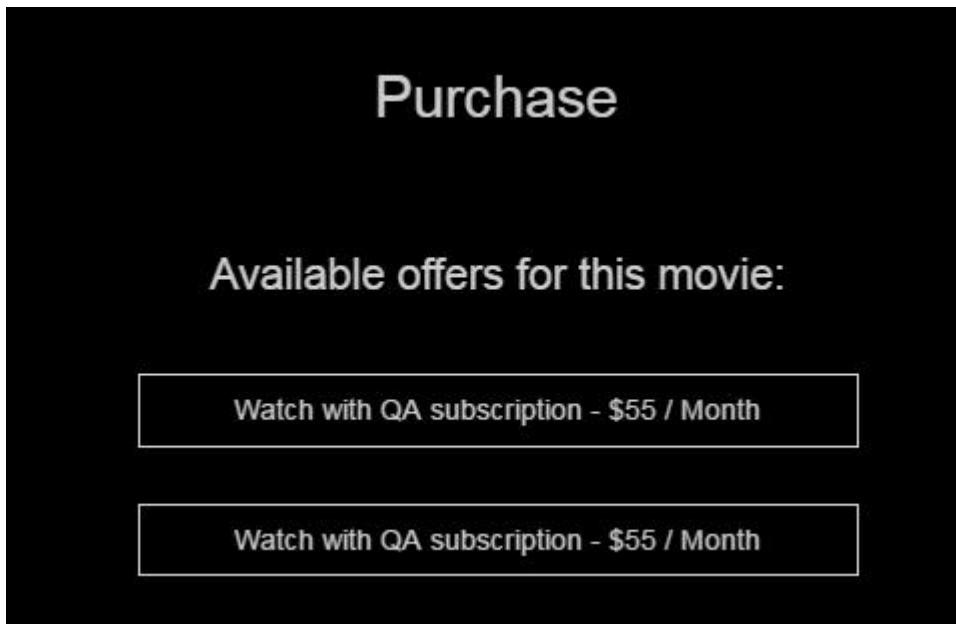
A sample *TerminalModel* element is shown below. The Movie Screen should indicate that this movie can be viewed on an AndroidSmartphone.

```
{
  "responseElementType": "TerminalModel",
  "vodCasNames": [
    "Viaccess-Purple-DRM"
  ],
  "liveVsName": "RiGHTvLive",
  "vodVsName": "amazon-s3",
  "categoryExternalId": "Smartphone",
  "highestDefinitionForVideo": "HD",
  "name": "AndroidSmartphone", 
  "externalId": "AndroidSmartphone",
  "vodCasName": "Viaccess-Purple-DRM",
  "liveCasName": "Viaccess-Purple-DRM",
  "vodVsNames": [
    "amazon-s3"
  ]
}
```

36 List Offers

36.1 Overview

Users need to know through which offers a selected movie is available. For example a movie may be available for TVOD order, through an SVOD subscription, as part of a Movie Package, or through a Service Plan. A list of all offers may be shown on the Movie screen and/or in a separate Offers view that displays when the user presses a Purchase button.



Tutorials that provide instructions for listing offers for different business models are listed below.

36.2 Related Tutorials

To	Refer to Tutorial
Get a list of TVOD offers for the movie	37 List TVOD Offers (p.117)
Get a list of SVOD offers that include the movie	38 List SVOD Offers for VOD Movie (p.122)
Get a list of Service Plan offers that include the movie	39 List Service Plan Offers for a VOD Movie (p.125)
Get a list of Movie Packages that include the movie	40 List VOD Movie Package Offers (p.129)

Create a Movie screen	33 Create a Movie Screen (p.106)
-----------------------	--

37 List TVOD Offers

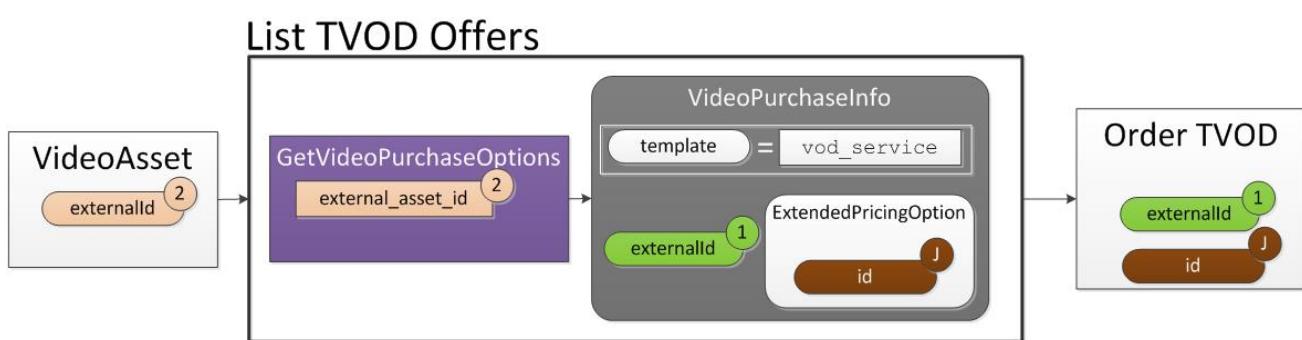
37.1 Overview

Users must be able to see details of all TVOD offers through which a specific movie can be purchased.

In the example below, two TVOD offers are present. The pricing matrix for the first offer (1) provides for View Online and Buy and Keep delivery modes. The pricing matrix for the second offer (2) includes only the View Online delivery mode. Note that the View Online delivery modes are priced differently between the two offers.



37.2 Requests and Responses



Request: [GetVideoPurchaseOptions](#) (from RiGHTv Front-end API)

This request gets data on all TVOD offers, SVOD services, Movie Packages and Service Plans through which the specified movie can be purchased.

Parameter	Description
<i>external_asset_id</i>	The external ID of the video asset in the Content Library.
<i>store_id</i>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [VideoPurchaseInfo](#) (from RiGHTv Front-end API)

The response lists all published pages on which the asset is available, each detailed in a *VideoPurchaseInfo* element. The *VideoPurchaseInfo* element for each page lists all the offers through which the page can be purchased.

For TVOD offers, the significant properties include:

Property	Description
<i>externalId</i>	The external ID of the page in the VOD Catalog.
<i>template</i>	Indicates the purchase option type. For TVOD will be <i>vod_service</i> .
<i>pricingOptions</i>	Lists the TVOD offers through which the page can be purchased, each in an <i>ExtendedPricingOption</i> element. Significant parameters are shown in the table that follows.

pricingOptions Properties:

Property	Description
<i>id</i>	The ID of the pricing option
<i>name</i>	The name of the pricing option.
<i>rentalPeriod</i>	The rental period offered.



<i>rentalPeriodUnit</i>	The unit in which <i>rentalPeriod</i> is specified.								
	<table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>2</td><td>Minutes</td></tr> <tr> <td>3</td><td>Hours</td></tr> <tr> <td>4</td><td>Days</td></tr> </tbody> </table>	Value	Description	2	Minutes	3	Hours	4	Days
Value	Description								
2	Minutes								
3	Hours								
4	Days								
<i>downloadPeriod</i>	The download period offered.								
<i>downloadPeriodUnit</i>	The unit in which <i>downloadPeriod</i> is specified.								
	<table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>2</td><td>Minutes</td></tr> <tr> <td>3</td><td>Hours</td></tr> <tr> <td>4</td><td>Days</td></tr> </tbody> </table>	Value	Description	2	Minutes	3	Hours	4	Days
Value	Description								
2	Minutes								
3	Hours								
4	Days								
<i>price</i>	The price of the offer.								
<i>type</i>	The pricing option type. Possible values: streaming, download, converged, est, Est-Pre-Order								
<i>estOnlineAvailabilityPeriod</i>	The period of time during which content purchased through an eST order can be downloaded and streamed.								



<i>estOnlineAvailability</i>	The unit in which <i>estOnlineAvailability</i> <i>Period</i> is specified.								
<i>PeriodUnit</i>	<table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>4</td><td>Days</td></tr> <tr> <td>6</td><td>Months</td></tr> <tr> <td>7</td><td>Years</td></tr> </tbody> </table>	Value	Description	4	Days	6	Months	7	Years
Value	Description								
4	Days								
6	Months								
7	Years								
<i>discountPurchaseOption</i>	Describes the discount option available.								
<i>storeProductReferenceId</i>	The ID of the app store product reference.								

37.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Purchase the movie	119 The Purchase Process (p.312)

37.4 Example

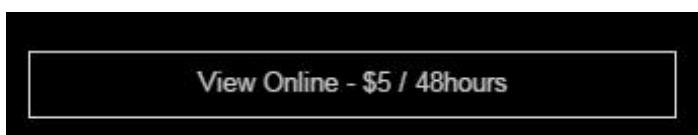
https://<operator>-atp.tvaas.com/RTEFacade/GetVideoPurchaseOptions?external_asset_id=Bridge+of+Spies&language_code=en&client=json

This request calls for purchase option data on the movie *Bridge of Spies*.

A sample VideoInfo element is shown below. The template type *vod_service* indicates this is a TVOD offer. The pricing option displayed indicates that the movie is made available for *rent*, for \$5.

```
1
  └─ template : "vod_service" ←
  └─ externalId : "Bridge of Spies_TVOD"
    ┌─ servicePlans
    ┌─ services
    ┌─ responseElementType : "VideoPurchaseInfo"
    ┌─ videoPackage
    ┌─ pricingOptions
    ┌─ 0
      ┌─ allowedViews : ""
      ┌─ rentalPeriod : 48
      ┌─ type : "streaming"
      ┌─ discountPurchaseOption
        ┌─ estOnlineAvailabilityPeriodUnit : ""
        ┌─ responseElementType : "ExtendedPricingOption"
        ┌─ downloadPeriod : ""
        ┌─ price : 5 ←
        ┌─ estOnlineAvailabilityPeriod : ""
        ┌─ downloadPeriodUnit : ""
        ┌─ name : "Rent" ←
        ┌─ id : 272099
        ┌─ rentalPeriodUnit : 3
      ┌─ 1
      ┌─ 2
    ┌─ seasonOfferOptions
      ┌─ name : "Bridge of Spies"
    ┌─ vodCards
      ┌─ definition : ""
    ┌─ supportedTerminalModels
      ┌─ id : 5308
      ┌─ status : 2
```

The application would display:

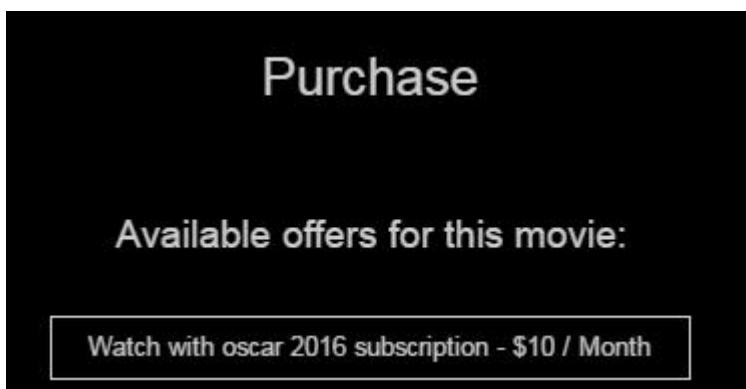


38 List SVOD Offers for VOD Movie

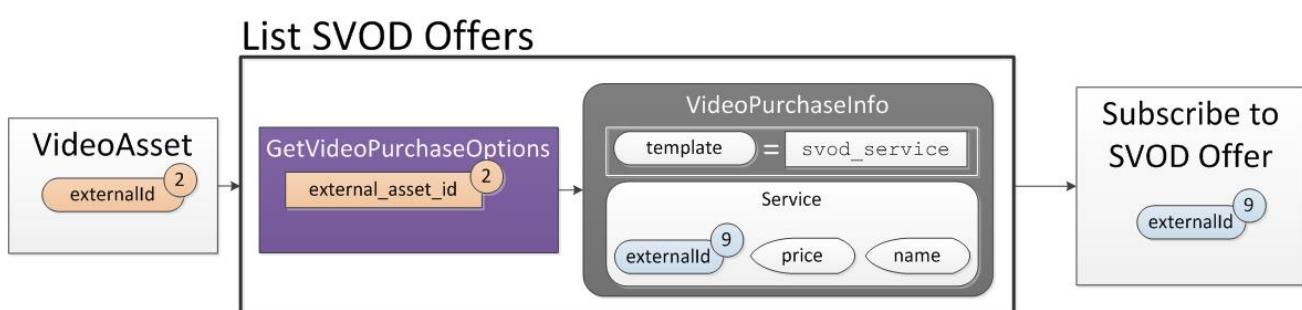
38.0.1 Overview

Users must be able to see details of all SVOD offers through which a specific movie can be accessed.

In the example below, the user can access the movie by subscribing to the Oscar 2016 SVOD offer, for \$10 per month.



38.1 Requests and Responses



Request: [GetVideoPurchaseOptions](#) (from RiGHTv Front-end API)

This request gets data on all TVOD offers, SVOD services, Movie Packages and Service Plans through which the specified movie can be purchased.

Parameter	Description
<code>external_asset_id</code>	The external ID of the video asset in the Content Library.
<code>store_id</code>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [VideoPurchaseInfo](#) (from RiGHTv Front-end API)

The response lists all published pages on which the asset is available, each detailed in a *VideoPurchaseInfo* element. The *VideoPurchaseInfo* element for each page lists all the offers through which the page can be purchased.

For SVOD offers, the significant properties include:

Property	Description										
<i>template</i>	Indicates the page type. For SVOD services will be <i>svod_service</i> .										
<i>service</i>	<p>Lists all SVOD offers that include the specified page, each in a <i>Service</i> element. Significant parameters include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The external ID of the SVOD service.</td></tr> <tr> <td><i>name</i></td><td>The name of the SVOD service.</td></tr> <tr> <td><i>price</i></td><td>The price of the SVOD service.</td></tr> <tr> <td><i>storeProductReferenceId</i></td><td>The ID of the app store product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The external ID of the SVOD service.	<i>name</i>	The name of the SVOD service.	<i>price</i>	The price of the SVOD service.	<i>storeProductReferenceId</i>	The ID of the app store product reference.
Property	Description										
<i>externalId</i>	The external ID of the SVOD service.										
<i>name</i>	The name of the SVOD service.										
<i>price</i>	The price of the SVOD service.										
<i>storeProductReferenceId</i>	The ID of the app store product reference.										

38.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Subscribe to the SVOD service	119 The Purchase Process (p.312)

38.3 Example

https://<operator>-atp.tvaas.com/RTEFacade/GetVideoPurchaseOptions?external_asset_id=Bridge+of+Spies&language_code=en&client=json

This request calls for purchase option data on the movie *Bridge of Spies*.



A sample VideoInfo element is shown below. The template type `svod_service` indicates this is an SVOD offer. The SVOD offer with external ID `oscar 2016` costs \$10 per month.

```

JSON
+ metadata
+ response
  + 0
  + 1
  + 2
    - template : "svod_service" ←
    - externalId : "Bridge of Spies_SVOD"
+ servicePlans
+ services
  + 0
    - quotaSizeGB : ""
    - attachments
      + 0
        - description : ""
        - externalId : "oscar 2016" ←
    - serviceCatalogFolders
      + 0
        - responseElementType : "FolderReference"
        - externalId : "Oscar_SVOD"
        - id : 5305
        - responseElementType : "Service"
        - simultaneousViewsLimit : ""
        - price : 10 ←
    - servicePlanReferenceList
      - name : "oscar 2016"
    - extrafields
      - model : "SVOD"
      - id : 21
  
```

The application would display:

Watch with oscar 2016 subscription - \$10 / Month



39 List Service Plan Offers for a VOD Movie

39.1 Overview

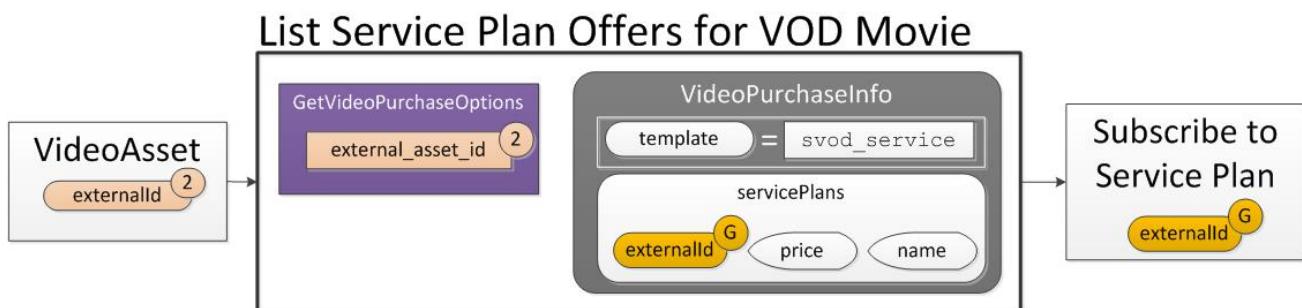
Users must be able to see a list of all Service Plan offers through which a movie can be purchased.



After clicking to purchase, the user should be shown details of the Service Plan before purchasing.



39.2 Requests and Responses



Request: [GetVideoPurchaseOptions](#) (from RiGHTv Front-end API)

This request gets data on all available purchase options for a specified movie asset, including Service Plans that include SVOD services through which the movie is available.

Parameter	Description
<i>external_asset_id</i>	The external ID of the asset.
<i>store_id</i>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [VideoPurchaseInfo](#) (from RiGHTv Front-end API)

The response lists all published pages on which the asset is available, each detailed in a **VideoPurchaseInfo** element. The **VideoPurchaseInfo** element for each page lists the Service Plans that include the SVOD services that include the page. Significant properties include:

Property	Description										
<i>servicePlans</i>	<p>Lists all Service Plans that include the specified page, each in a <i>ServicePlan</i> element. Significant parameters include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The Service Plan external ID.</td></tr> <tr> <td><i>name</i></td><td>The Service Plan name.</td></tr> <tr> <td><i>price</i></td><td>The Service Plan price.</td></tr> <tr> <td><i>storeProductReferenceId</i></td><td>The ID of the app store product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The Service Plan external ID.	<i>name</i>	The Service Plan name.	<i>price</i>	The Service Plan price.	<i>storeProductReferenceId</i>	The ID of the app store product reference.
Property	Description										
<i>externalId</i>	The Service Plan external ID.										
<i>name</i>	The Service Plan name.										
<i>price</i>	The Service Plan price.										
<i>storeProductReferenceId</i>	The ID of the app store product reference.										

39.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Subscribe to the Service Plan	119 The Purchase Process (p.312)

39.4 Example

https://<operator>-atp.tvaas.com/RTEFacade/GetVideoPurchaseOptions?external_asset_id=Bridge+of+Spies&language_code=en&client=json

This request calls for purchase option data on the movie *Bridge of Spies*.

A sample VideoInfo element is shown below. For the SVOD service shown, one Service Plan, with external ID *Gold_Plan001*, is listed as including that SVOD service. The cost of the Service Plan is \$15 per month.

The JSON response structure for service plans is as follows:

- JSON object:
 - metadata
 - response
 - 0
 - template : "svod_service"
 - externalId : "Bridge of Spies_SVOD"
 - 1
 - 2
 - template : "svod_service"
 - externalId : "Bridge of Spies_SVOD"
 - servicePlans
 - 0
 - attachments
 - excludedFromSelfService : false
 - servicePlanOption : "FIXED_BY_DURATION_END_DAY"
 - bouquets
 - externalId : "Gold_Plan001"
 - description : "oscars & basic pacakge"
 - fixedEndDate : ""
 - services
 - responseElementType : "ServicePlan"
 - removalDate : 1493556000000
 - activationOption : "IMMEDIATE"
 - price : 15
 - consumptionDurationInDays : 10
 - channelPackages
 - name : "Gold Plan"
 - activationDate : ""
 - extraFields
 - services

The application would display:

Watch with Gold Plan service plan - \$15/1 month

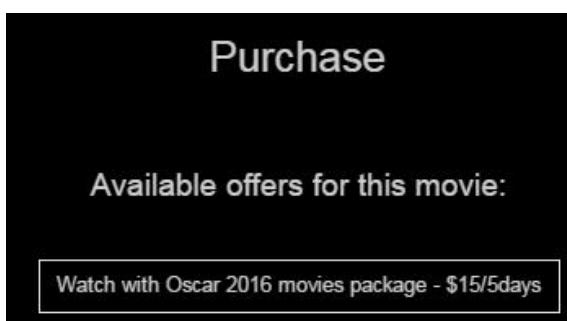


40 List VOD Movie Package Offers

40.1 Overview

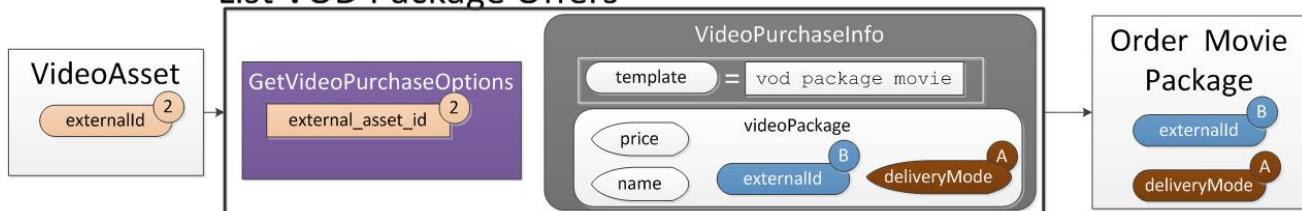
Users must be able to see details of all VOD Movie Packages through which a specific movie can be purchased.

In the example below, the user can purchase the movie by purchasing the Oscar 2016 movie package, for \$15, to have access to the movie for five days.



40.2 Requests and Responses

List VOD Package Offers



Request: [GetVideoPurchaseOptions](#) (from RiGHTv Front-end API)

This request gets data on all TVOD offers, SVOD services, Movie Packages and Service Plans through which the specified movie can be purchased.

Parameter	Description
<i>external_asset_id</i>	The external ID of the video asset in the Content Library.
<i>store_id</i>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [VideoPurchaseInfo](#) (from RiGHTv Front-end API)

The response lists all published pages on which the asset is available, each detailed in a *VideoPurchaseInfo* element. The *VideoPurchaseInfo* element for each page lists all the offers through which the page can be purchased.

For Movie Packages, the significant properties include:

Property	Description												
<i>template</i>	Indicates the purchase option type. For a Movie Package will be <i>vod package movie</i> .												
<i>videoPackage</i>	<p>Lists the Movie Packages through which the page can be purchased, each in an <i>VideoPackage</i> element. Significant parameters include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The external ID of the Movie Package.</td></tr> <tr> <td><i>name</i></td><td>The name of the Movie Package.</td></tr> <tr> <td><i>price</i></td><td>The price of the Movie Package.</td></tr> <tr> <td><i>storeProductReferenceId</i></td><td>The ID of the app store product reference.</td></tr> <tr> <td><i>estStoreProductReferenceId</i></td><td>The ID of the app store EST product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The external ID of the Movie Package.	<i>name</i>	The name of the Movie Package.	<i>price</i>	The price of the Movie Package.	<i>storeProductReferenceId</i>	The ID of the app store product reference.	<i>estStoreProductReferenceId</i>	The ID of the app store EST product reference.
Property	Description												
<i>externalId</i>	The external ID of the Movie Package.												
<i>name</i>	The name of the Movie Package.												
<i>price</i>	The price of the Movie Package.												
<i>storeProductReferenceId</i>	The ID of the app store product reference.												
<i>estStoreProductReferenceId</i>	The ID of the app store EST product reference.												

For more properties, see the RiGHTv Front-end API Developers Guide.

40.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Purchase a TVOD offer	119 The Purchase Process (p.312)



40.4 Example

https://<operator>-atp.tvaas.com/RTEFacade/GetVideoPurchaseOptions?external_asset_id=Bridge+of+Spies&language_code=en&client=json

This request calls for a list of all Movie Packages that include the movie *Bridge of Spies*.

A sample `VideoInfo` element is shown below. The template type `vod package movie` indicates this is a Movie Package. The Movie Package's external ID is *Oscar 2016*, and the price is \$15.

The screenshot shows a JSON editor interface with a tree view of a complex JSON object. The object structure is as follows:

- JSON
 - metadata
 - response
 - template : "vod package movie" ← Red arrow 1
 - externalId : "8d2b634e-6392-4285-8262-b4135850fd5d"
 - servicePlans
 - services
 - responseElementType : "VideoPurchaseInfo"
 - videoPackage
 - attachments
 - rentalPeriod : ""
 - externalId : "Oscar 2016" ← Red arrow 2
 - description : ""
 - packagePeriod : 5
 - responseElementType : "VideoPackage"
 - estPrice : -1
 - plannedPublishDate : ""
 - removalDate : ""
 - deliveryMode : "Streaming"
 - price : 15 ← Red arrow 3
 - name : "Oscar 2016"
 - extrafields
 - definition : ""
 - id : 5102
 - packagePeriodUnit : 4
 - rentalPeriodUnit : ""
 - status : 2
 - pricingOptions
 - seasonOfferOptions
 - name : "Bridge of Spies"
 - vodCards
 - definition : ""
 - supportedTerminalModels
 - id : 5104
 - status : 2

The application would display:

Watch with Oscar 2016 movies package - \$15/5days



41 Get Pricing Matrix Info

41.1 Overview

The *AggregatedVideo* entity lists the pricing matrix applicable to each offer, by pricing matrix ID. The user must be shown the details of a pricing matrix before purchasing.

41.2 Requests and Responses

Request: [GetPricingMatrix \(from RiGHTv Front-end API\)](#)

This request retrieves info on the specified pricing matrix. Significant parameters include:

Parameter	Description
<i>pricing_matrix_id</i>	The ID of the pricing matrix.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [PricingOption \(from RiGHTv Front-end API\)](#)

The returned entity presenting the details of the specified pricing matrix, including the following properties:

Property	Description									
<i>name</i>	The name of the pricing option.									
<i>rentalPeriod</i>	The rental period offered.									
<i>rentalPeriodUnit</i>	The unit in which <i>rentalPeriod</i> is specified. <table border="1" style="margin-top: 10px;"> <tr> <th>rentalPeriodUnit Value</th> <th>Interpretation</th> </tr> <tr> <td>2</td> <td>Minutes</td> </tr> <tr> <td>3</td> <td>Hours</td> </tr> <tr> <td>4</td> <td>Days</td> </tr> </table>		rentalPeriodUnit Value	Interpretation	2	Minutes	3	Hours	4	Days
rentalPeriodUnit Value	Interpretation									
2	Minutes									
3	Hours									
4	Days									



Property	Description	
<i>downloadPeriod</i>	The download period offered.	
<i>downloadPeriodUnit</i>	The unit in which <i>downloadPeriod</i> is specified.	
	rentalPeriodUnit Value	Interpretation
	2	Minutes
	3	Hours
	4	Days
<i>price</i>	The price of the offer.	
<i>type</i>	The pricing option type. Possible values: <i>streaming</i> , <i>download</i> , <i>converged</i> , <i>est</i> , <i>Est-Pre-Order</i>	

41.3 Related Tutorials

41.4 Example

https://<operator>.tvaas.com/RTEFacade/GetPricingMatrix?pricing_matrix_id=4024&client=json

This example will return the pricing matrix information for pricing matrix 4024, as below.

```
"response": [
  {
    "responseElementType": "PricingOption",
    "allowedViews": "",
    "downloadPeriod": "",
    "price": 5,
    "rentalPeriod": 48,
    "downloadPeriodUnit": "",
    "name": "Streaming",
    "id": 4025,
    "type": "streaming",
    "rentalPeriodUnit": 3
  }
]
```

Here, the pricing matrix includes information for streaming rental only - \$5. *rentalPeriodUnit* is set to 3, indicating units of hours. Thus *rentalPeriod* indicates that the rental period is 48 hours.



42 Get User Ratings

42.1 Overview

Users can rate movies and the average of all users' ratings of a particular movie should be displayed on the Movie screen as part of the movie metadata.

42.2 Requests and Responses

Get Movie Ratings



Request: [GetContentRatingList](#) (from COMPASS RT API)

This request returns the ratings of the specified content(s).

Parameter	Description
<i>content_id_list</i>	List the external content IDs of the movies for which ratings are to be retrieved.

For more parameters, see the Compass RT API Guide.

Response: [ContentRating](#) (from COMPASS RT API)

The returned entity provides the overall user rating of the content.

Property	Description
<i>rating</i>	Gives the average rating for the content item.
<i>ratersCount</i>	Gives the number of ratings performed.

42.3 Related Tutorials

To	Refer to Tutorial
Create a Movie Screen	33 Create a Movie Screen (p.106)

42.4 Example

https://<operator>.tvaas.com/compass/GetContentRatingList?content_id_list=Argo&client=json

This request calls for the user rating assigned to the movie Argo.

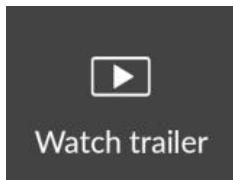
A sample return is shown below. In this case, the movie has an average rating of 3, though only 2 users have rated it.

```
[{"metadata": {  
    "request": "GetContentRatingList",  
    "fullLength": 1,  
    "timestamp": 1455718637654  
},  
 "response": [  
     {  
      "ratersCount": 2, ←  
      "rating": 3, ←  
      "id": "Argo"  
     }  
   ]  
}]
```

43 Playing a Trailer

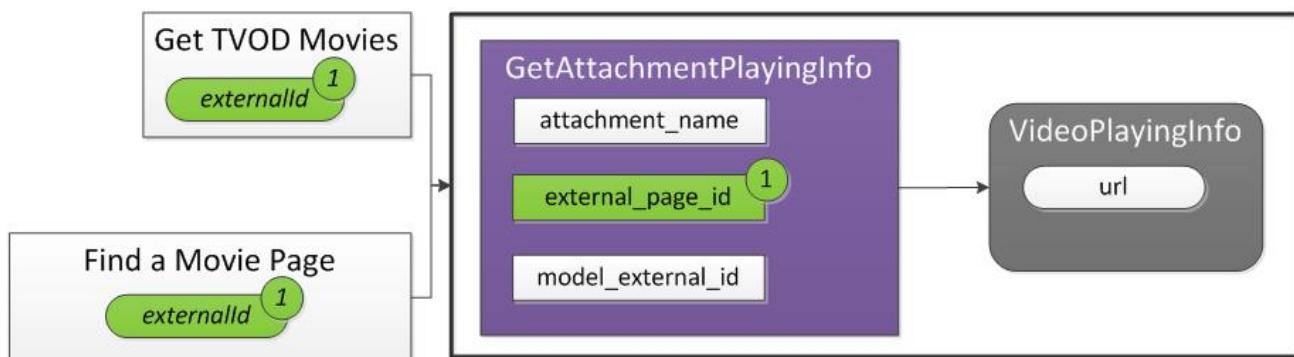
43.1 Overview

Trailers are provided for many movies. These should be playable from the movie screen. Anonymous users should be able to play trailers.



43.2 Requests and Responses

Trailers are defined as video attachments to a movie. To play a trailer, you first need to get its playback URL.



Request: [GetAttachmentPlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for the specified trailer movie. Significant parameters include:

Parameter	Description
<i>external_page_id</i>	The external ID of the movie page in the VOD Catalog. If arriving at the Movie Screen from a catalog page (see 18 Get TVOD Movies (p.52)), the externalID of a movie page will be known. If not, the ID of any VOD Catalog page that includes the specified movie asset can be used (see 44 Find a Movie Page (p.139)).
<i>model_external_id</i>	The type of device on which the device is to be played.
<i>attachment_name</i>	The name of the attachment, which is typically set to trailer.

Response: [VideoPlayingInfo](#) (from RiGHTv Front-end API)

This document is Viaccess SA or Orca Interactive intellectual property

Copying is strictly forbidden

Confidential – under NDA

This entity provides various properties about the specified video file. For getting the playback URL, the significant property is:

Property	Description
<i>url</i>	The playback URL of the video. See separate documentation for instructions on operating the player.

43.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Get the external ID of a movie page that includes the asset and a trailer	44 Find a Movie Page (p.139)

43.4 Example

https://<operator>.tvaas.com/RTEFacade/GetAttachmentPlayingInfo?attachment_name=Trailer&external_page_id=Argo&model_external_id=IPad&client=json

This request gets the playback URL for the `Trailer` of the movie `Argo`, to be played on an `IPad`.

A sample response is shown below. The playback url is indicated.

```
{
  "metadata": {
    "request": "GetAttachmentPlayingInfo",
    "timestamp": 1456215983922
  },
  "response": {
    "responseElementType": "VideoPlayingInfo",
    "identifier": "",
    "protocol": "http",
    "filename": "Test Trailer.mpg_EtsHlsClrS3",
    "port": "0",
    "ip": "",
    "id": 301,
    "position": 0,
    "url": "http://d3cz5x0y15zny5.cloudfront.net/dotscreen/HLS/Argo/MAINArgo.m3u8",
    "token": ""
  }
}
```



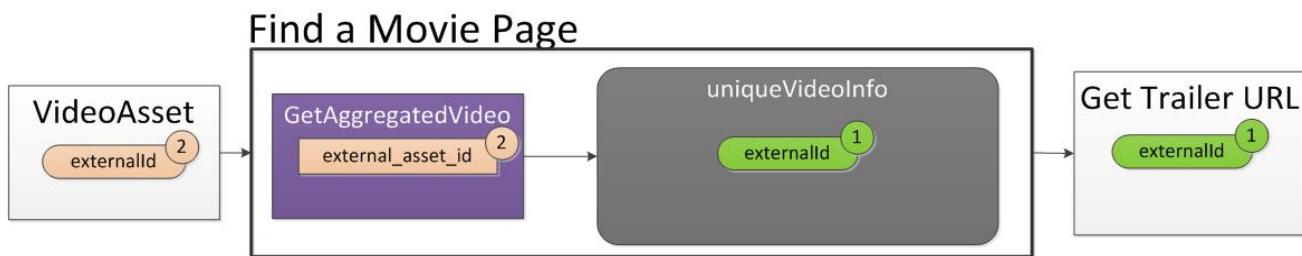

44 Find a Movie Page

44.1 Overview

The ID of a movie page in the VOD catalog is required as an input to a number of functions that may be executed from the Movie Screen. For example, the ID of a movie page must be provided when requesting a URL for a movie trailer.

The user reaches the Movie Screen from a TVOD catalog screen or from a list of recommendations. If reached from a TVOD catalog screen, the ID of a VOD Catalog movie page on which the movie is presented will be known. If reached from a recommendations list, no corresponding movie page will yet be known.

44.2 Requests and Responses



The GetAggregatedVideo function gets a list of all movie pages in the VOD Catalog that include the specified movie asset from the Content Library. The ID of any returned movie page can be used when needed, for example, when requesting a trailer URL.

Request: [GetAggregatedVideo](#) (from RiGHTv Front-end API)

This request gets data on all TVOD offers and SVOD packages that include the specified movie asset. Only one parameter is required.

Parameter	Description
<i>external_asset_id</i>	The external ID of the video asset in the Content Library.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: UniqueVideoInfo (from RiGHTv Front-end API)

The returned entity includes one or more UniqueVideoInfo elements, each presenting a unique offer in which the movie is available.

Property	Description
<i>externalId</i>	The ID of the Movie page in the VOD Catalog.

44.3 Related Tutorials

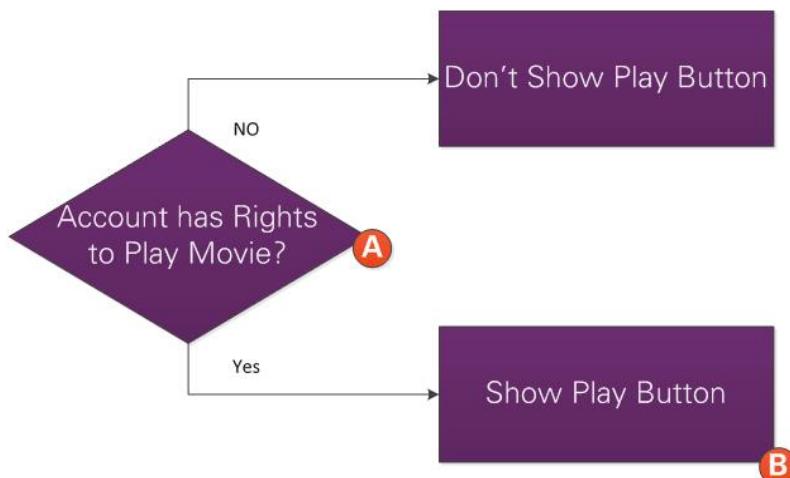
To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Play a movie trailer	43 Playing a Trailer (p.137)



45 Playing a Movie

45.1 Overview

Unlike trailers, only logged-in users can view movies. Playing a movie thus involves two steps as shown below.



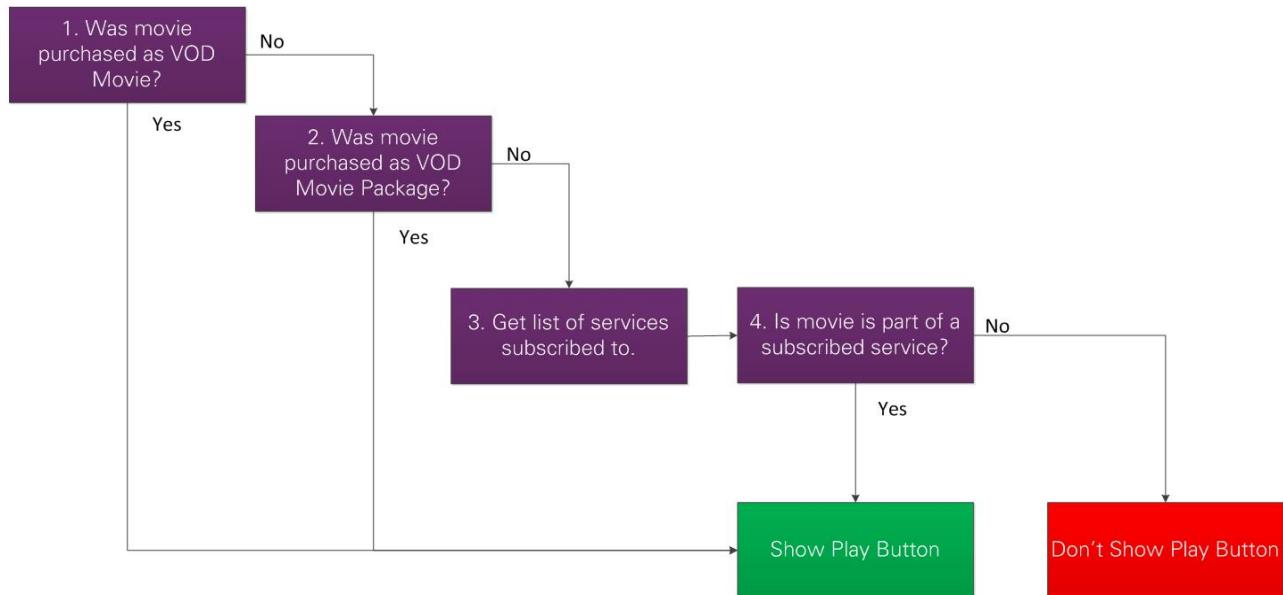
- Check if the account has rights to view the movie.
- If yes, display the Play button. To play a movie, you first need to get its playback URL.

45.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Check whether the account has the right to play the movie	46 Check if Account Has Rights to Play Movie (p.143)
Get the playback URL	49 Get Movie Playback URL (p.151)
Learn how to implement the media player	50 Using the Media Player (p.154)

46 Check if Account Has Rights to Play Movie

To check if an account has rights to play a particular movie, the following steps must be followed.



46.1 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Check whether the account has the right to play the movie through a purchased TVOD or VOD Movie Package offer	47 Check TVOD and VOD Movie Package Video Tickets (p.144)
Check whether the account has the right to play the movie through an SVOD offer to which it subscribes (either directly or through a Service Plan)	48 Check SVOD Subscriptions (p.148)

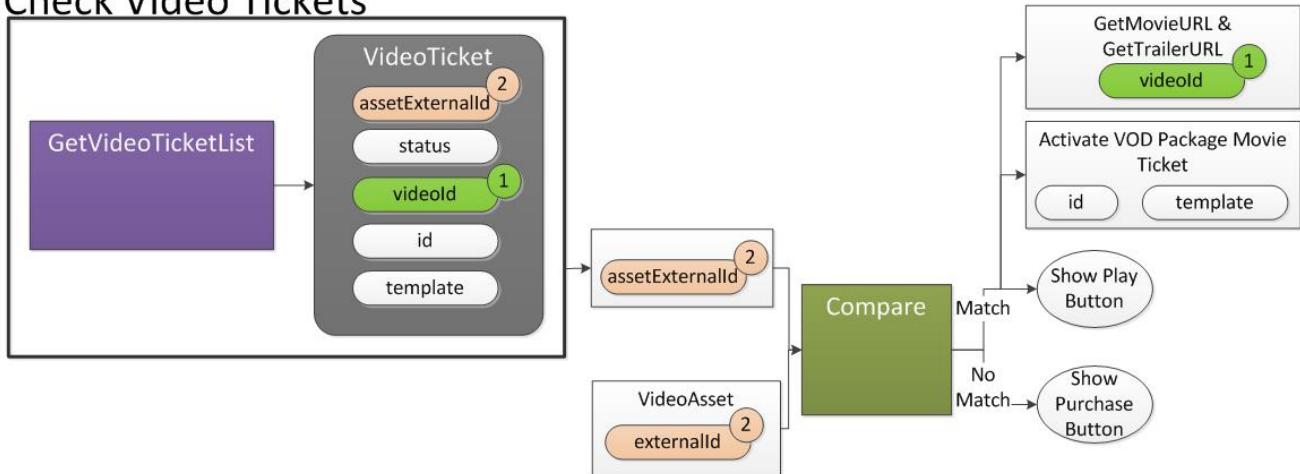
47 Check TVOD and VOD Movie Package Video Tickets

47.1 Overview

When determining whether to show the user the Play button for a movie, check whether the account has purchased access to that movie through a TVOD offer or as part of a VOD Movie Package.

47.2 Requests and Responses

Check Video Tickets



Request: [GetVideoTicketList](#) (from RiGHTv Front-end API)

Returns a list of movies purchased as VOD movies and as part of VOD packages. No parameters are required.

Request: [VideoTicket](#) (from RiGHTv Front-end API)

This entity lists movies and movie packages purchased by the account. It includes the following properties.

Property	Description
<i>assetExternalId</i>	The external ID of the asset in the Content Library.
<i>status</i>	Indicates if active or expired.
<i>videoid</i>	The external ID of the movie page in the VOD Catalog.
<i>id</i>	The ID of the ticket opened for the movie.
<i>template</i>	The template of the movie page in the VOD Catalog (vod_service / svod_service / vod package movie)

Check the external ID of the movie to be played against the values of *assetExternalId* in the listed VOD movie ticket.

47.3 Related Tutorials

Check whether the account has the right to play the movie through a

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Check whether the account has the right to play the movie	46 Check if Account Has Rights to Play Movie (p.143)
Active a VOD Package movie ticket	52 Activate VOD Package Movie Ticket (p.157)

47.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetVideoTicketList?client=json>

This call requests all VOD movie and VOD movie packages to which the user has access.

The sample response below shows that the account has active access to the movie Goodfellas, ordered as a VOD movie.



```
"template": "vod_service",  
"advisories": "",  
"prLevel": 4,  
"allowedEstDownloadingDevices": -1,  
"videoId": 308,  
"type": "transaction",  
"assetExternalId": "Goodfellas",  
"responseElementType": "VideoTicket",  
"attachment": {},  
"downloadPeriod": 7,  
"deliveryMode": "converged",  
"assetId": 5247,  
"id": 316023,  
"windowEnd": "",  
"videoPackageName": "",  
"rentalPeriodUnit": 3,  
"allowedViews": "",  
"windowStart": "",  
"videoPackageExternalId": "",  
"rentalPeriod": 12,  
"prName": "R",  
"videoExternalId": "Goodfellas",  
"downloadWindowEnd": 1460636490113,  
"downloadWindowStart": 1460031714846,  
"downloadPeriodUnit": 4,  
"name": "Goodfellas (es)",  
"currentEstDownloadingDevices": "",  
"status": "active"
```

The sample response below shows that the account has active access to the movie Argo, ordered as part of a VOD movie package. The video ticket is an active status.



```
"template": "vod package movie",  
"advisories": "",  
"prLevel": 4,  
"allowedEstDownloadingDevices": "",  
"videoId": 9201,  
"type": "transaction",  
"assetExternalId": "Argo",  
"responseElementType": "VideoTicket",  
"attachment": {},  
"downloadPeriod": "",  
"deliveryMode": "streaming",  
"assetId": 5274,  
"id": 271009,  
"windowEnd": 1463810472434,  
"videoPackageName": "MyMovies",  
"rentalPeriodUnit": 4,  
"allowedViews": "",  
"windowStart": 1459922472434,  
"videoPackageExternalId": "3456345634",  
"rentalPeriod": 45,  
"prName": "R",  
"videoExternalId": "f8dc1ae7-13dc-43d9-8df2-3587a082046b",  
"downloadWindowEnd": "",  
"downloadWindowStart": "",  
"downloadPeriodUnit": 0,  
"name": "Argo (es)",  
"currentEstDownloadingDevices": "",  
"status": "active"
```



48 Check SVOD Subscriptions

48.1 Overview

When determining whether to show the user the Play button for a movie, check whether the account has purchased access to that movie through an SVOD subscription.

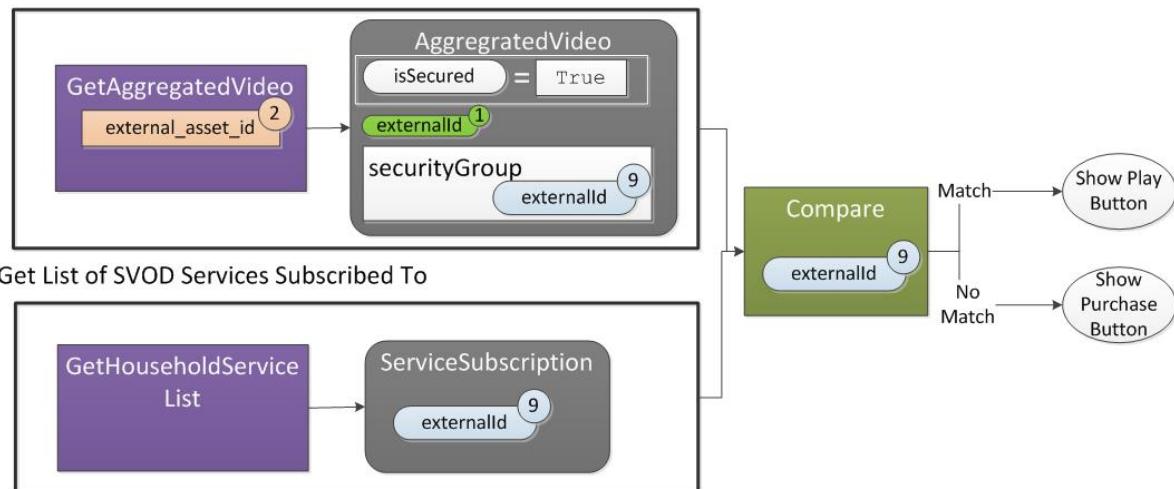
Note

The instructions provided here apply whether the account subscribed to an SVOD service directly, or through a Service Plan.

48.2 Requests and Responses

Was movie purchased through an SVOD offer?

Get List of SVOD Services that include Movie



Use *GetAggregatedVideo* to get a list of all SVOD services that include the specified asset. Use *GetHouseholdServiceList* to get a list of all *ServiceSubscriptions* to which the household is subscribed. If there is a match, the account does subscribe to at least one SVOD service that includes the specified movie.

48.2.1 Requests and Responses: GetAggregatedVideo

Request: [GetAggregatedVideo](#) (from RiGHTv Front-end API)

This request gets data on all TVOD offers and SVOD packages that include the specified movie asset.

Parameter	Description
external_asset_id	The external ID of the asset.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: UniqueVideoInfo (from RiGHTv Front-end API)

The returned entity includes one or more UniqueVideoInfo elements, each presenting a unique offer in which the movie is available.

Property	Description
isSecured	If set to true, indicates that the movie is offered in at least one SVOD service.
securityGroups	Lists the relevant SVOD services byexternalId.
externalId	The ID of the Movie page in the VOD Catalog.

48.2.2 Requests and Responses: GetHouseHoldServiceList

Request: GetHouseHoldServiceList (from RiGHTv Front-end API)

Returns a list of all services the account is subscribed to. No parameters are required.

Response: ServiceSubscription (from RiGHTv Front-end API)

Lists all the services subscribed to, including SVOD services. Significant parameters include:

Property	Description
externalId	The external ID of the service.

48.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Check whether the account has the right to play the movie	46 Check if Account Has Rights to Play Movie (p.143)

48.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetAggregatedVideo?external_asset_id=Cinderella&language_code=es&client=json`

This request calls for aggregated video data on the movie Cinderella. The response for one of the SVOD offers in which that movie is included is shown below. This movie is included in the DisneySVOD service.

```
{
  "isSecured": true,
  "attachments": [],
  "encodings": [
    {
      "responseElementType": "Encoding",
      "name": "HLS"
    }
  ],
  "externalId": "Cinderella",
  "responseElementType": "UniqueVideoInfo",
  "simultaneousViewsLimit": "",
  "removalDate": "",
  "extrafields": [],
  "pricingMatrixId": 4024,
  "definition": "",
  "securityGroups": [
    {
      "responseElementType": "SecurityGroup", ←
      "externalId": "Disney", ←
      "type": "Service"
    }
  ],
  "discountId": "",
  "status": 2
},
```

<https://<operator>.tvaas.com/RTEFacade/GetHouseholdServiceList?scene=tvaas&client=json>

This request gets a list of all services to which the account is subscribed. The response includes the Disney SVOD service.

```
{
  "responseElementType": "ServiceSubscription",
  "subscriptionExpiration": "",
  "externalId": "Disney",
  "model": "SVOD", ←
  "id": 4,
  "serviceName": "Disney"
}
```

As the account is subscribed to the Disney service, and the movie to be played, Cinderella, is part of the Disney service, conclude that the account has the right to view the movie and the play button should be shown.

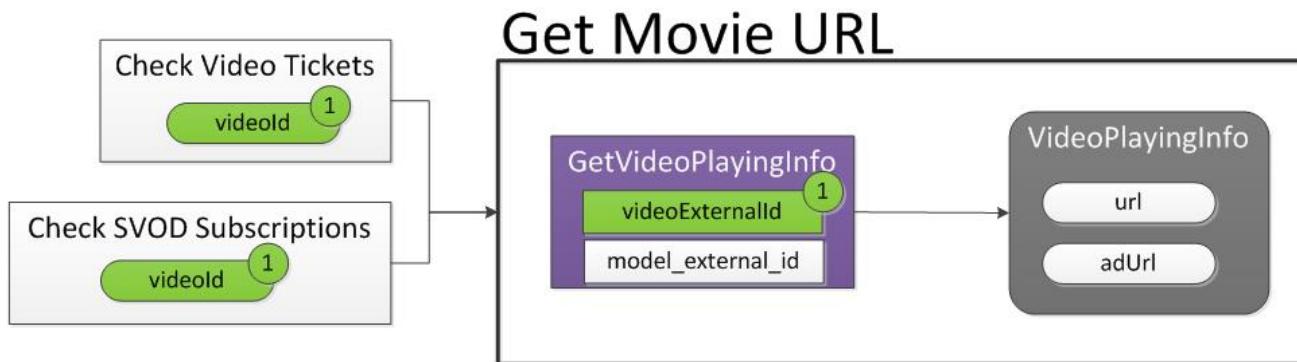


49 Get Movie Playback URL

49.1 Overview

When the user presses a Play button to watch a movie, the playback URL must be retrieved.

49.2 Requests and Responses



If the rights to play movie were purchased through an SVOD offer, a ticket for the individual movie must be opened. It is suggested that the ticket be opened when the user first attempts to view the movie. See [135 Opening a Ticket for an SVOD Movie \(p.357\)](#).

Note on Analytics: The Analytics Platform extracts data sent when requesting a playback URL for use in many of its charts and KPIs.

Request: [GetVideoPlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for the specified movie. Significant parameters include:

Parameter	Description
<code>video_external_id</code>	The external ID of the movie page in the VOD Catalog.
<code>resume</code>	Set to <code>true</code> or <code>false</code> , specifies whether to resume the movie from its last played position or not.
<code>device_model_id</code>	The type of device on which the device is to be played.

Response: [VideoPlayingInfo](#) (from RiGHTv Front-end API)

This entity provides various properties about the specified video file. For getting the playback URL, the significant property is:



Property	Description
<i>url</i>	The playback URL of the video. See separate documentation for instructions on operating the player.

49.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Learn how to implement the media player	50 Using the Media Player (p.154)
Open a ticket for an SVOD movie	135 Opening a Ticket for an SVOD Movie (p.357)

49.4 Example

https://<operator>.tvaas.com/RTEFacade/GetVideoPlayingInfo?video_external_id=Argo&resume=true&model_external_id=IPad&client=json

This request gets the playback URL for the movie Argo, to be played on an IPad.

Sample response for a successful call is shown below. Note the URL displayed.



```

{
  "metadata": {
    "request": "GetVideoPlayingInfo",
    "timestamp": 1456233125093
  },
  "response": {
    "responseElementType": "VideoPlayingInfo",
    "identifier": "",
    "protocol": "http",
    "filename": "Argo.mpg_EtsHlsClrS3",
    "port": "0",
    "ip": "",
    "id": 301,
    "position": 0,
    "url": "http://d3cz5x0y15zny5.cloudfront.net
      /dotscreen/HLS/Argo/MAINArgo.m3u8",
    "token": ""
  }
}

```

Sample response for an unsuccessful call is shown below. The output indicates failure, and informs that the account does not have a ticket for this movie.



```
[{"metadata": {  
    "request": "GetVideoPlayingInfo",  
    "timestamp": 1456217268803  
},  
"response": {  
    "code": "INVALID_TICKET", ←  
    "message": "No ticket for movie 301 in household 10300",  
    "status": "FAILURE"  
}  
}
```

50 Using the Media Player

For instructions on integrating the media player within a client application, see separate documentation provided.

50.1 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Mark the last position to allow the user to resume watching later	51 Mark Last Position (p.155)



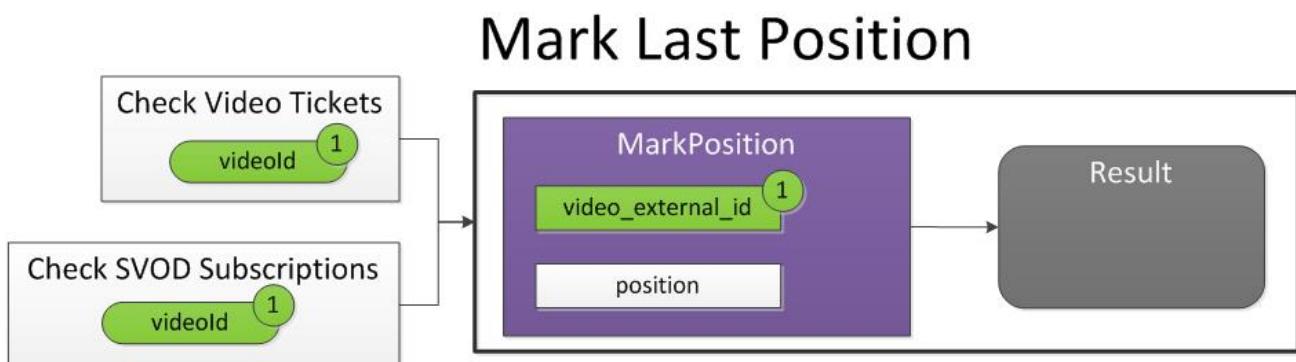
51 Mark Last Position

51.1 Overview

If a user stops playing a movie before it ends, the play position can be recorded to allow the user to continue viewing from the same point in the movie at another time.

Note on Analytics: For data collection purposes, marking the stopped position is used as an indication that the user has ended the viewing session. The stopped position must thus only be marked when the user closes the player, or automatically when the video stream stops. The last position must also be marked once the content has been played to the end. *MarkPosition* must be sent without caching.

51.2 Requests and Responses



Request: [MarkPosition](#) (from RiGHTv Front-end API)

This request stores (in the back end) the position where playback was paused. The following parameters are required:

Parameter	Description
<i>video_external_id</i>	The external ID of the VOD content in the catalog.
<i>position</i>	The position in milliseconds from the start of the movie

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicating whether the request was a SUCCESS or FAILURE.
<i>message</i>	A short message from the server.

51.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Learn how to implement the media player	45 Playing a Movie (p.141)

51.4 Example

https://<operator>.tvaas.com/RTEFacade/MarkPosition?video_external_id=Argo&position=25000&client=json

This request marks the position at 25000 milliseconds from the start of the movie Argo.

The response below shows that the position was successfully marked.

```
{
  "metadata": {
    "request": "MarkPosition",
    "timestamp": 1458460661348
  },
  "response": {
    "message": "Mark-Position operation finished successfully",
    "status": "SUCCESS"
  }
}
```

52 Activate VOD Package Movie Ticket

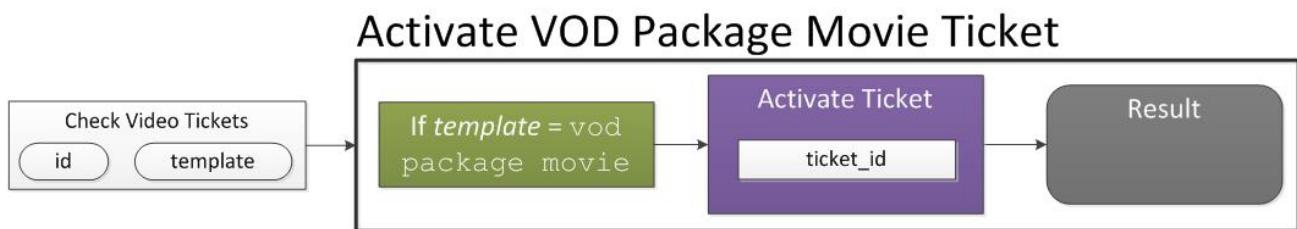
52.1 Overview

A video ticket is created automatically for each video bought or rented. When an individual movie is purchased, its ticket is activated automatically. Tickets for movies bought as part of a VOD movie package must be activated individually.

Note

Video tickets for VOD movie package movies should be activated when the Play or Download button on the movie's movie page is first clicked, and before requesting the playing or download URL.

52.2 Requests and Responses



Request: [ActivateTicket](#) (from RiGHTv Front-end API)

Call this request to activate a ticket in the VOD package.

Parameter	Description
<i>ticket_id</i>	The ID of the ticket representing the purchase of a VOD package.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

52.3 Related Tutorials

To	Refer to Tutorial
Order a VOD Movie Package	129 Ordering a VOD Movie Package (p.341)

52.4 Example

The request below gets a list of the account's tickets.

<https://<operator>.tvaas.com/RTEFacade/GetVideoTicketList?&client=json>

In the sample below, the movie has a ticket id 2701010. The movie can be identified as being part of a VOD movie package from the *template* property which indicates a vod package movie.

```
"template": "vod package movie", ←
"advisories": "",
"prLevel": 3,
"allowedEstDownloadingDevices": "",
"videoId": 9202,
"type": "transaction",
"assetExternalId": "Ant-Man",
"responseElementType": "VideoTicket",
"attachment": {},
"downloadPeriod": "",
"deliveryMode": "streaming",
"assetId": 5100,
"id": 271010, ←
>windowEnd": 1463810472557,
"videoPackageName": "MyMovies",
"rentalPeriodUnit": 4,
"allowedViews": "",
>windowStart": 1459922472557,
"videoPackageExternalId": "3456345634",
"rentalPeriod": 45,
"prName": "PG-13",
"videoExternalId": "7b2a4d40-00d8-4cf9-aba8-12d3d32e7152",
"downloadWindowEnd": "",
"downloadWindowStart": "",
"downloadPeriodUnit": 0,
"name": "Ant-Man (es)",
"currentEstDownloadingDevices": "",
"status": "active"
```



The action below activates the ticket with ticket id=271010.

`https://<operator>.tvaas.com/RTEFacade/ActivateTicket?ticket_id=271010&client=json`

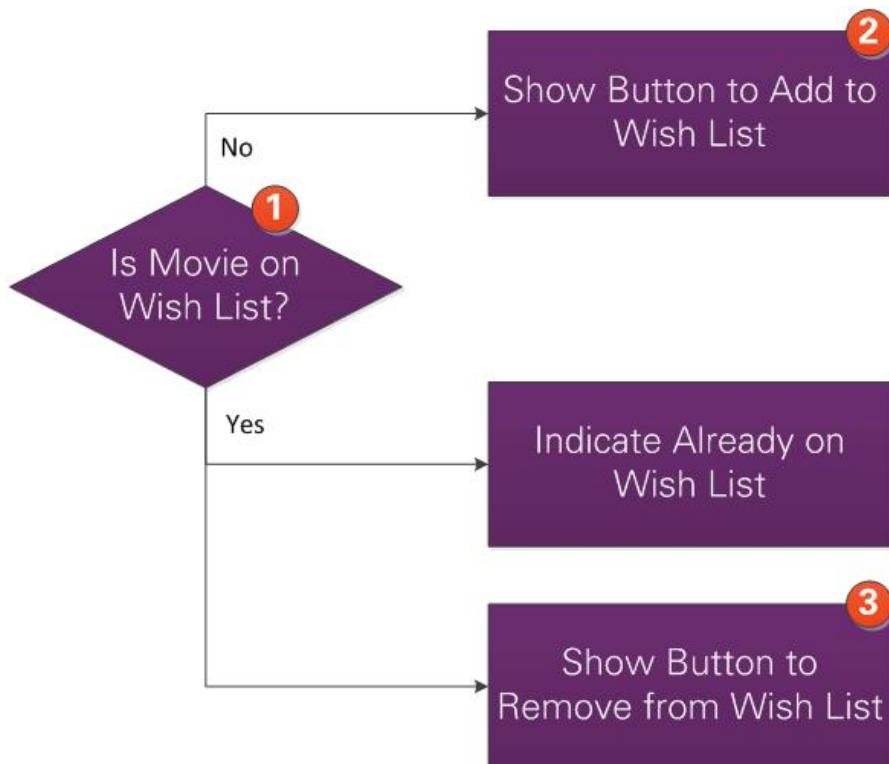
53 Manage a Wish List

53.1 Overview

A Wish List is a list of movies from a VOD catalog that the user selects and marks, so that later on they have direct access to either purchase or view them from a single location. The Movie Screen must allow the user to add the selected movie to the wish list, or remove it if previously added.

Wish lists are managed at the user level. The user must be logged in to be able to add to or remove from the wish list.

Tip: Show the Add to Wish List button even if the user is not logged in. Show the log in screen if an anonymous user clicks the button.



53.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Check whether a movie is already on the this wish list	54 Check the Wish List (p.162)
Add items to a wish list	55 Add to the Wish List (p.165)
Remove items from a wish list	56 Remove from the Wish List (p.167)
Display a wish list	156 Show My Wishlist (p.426)

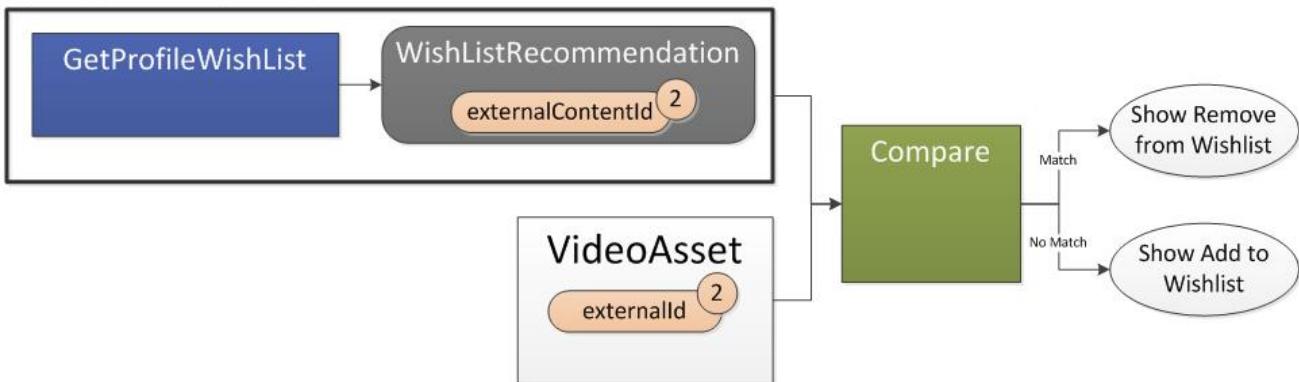
54 Check the Wish List

54.1 Overview

On the Movie screen, check whether the movie is already on the user's wishlist or not, to determine whether to display an Add to Wishlist or Remove from Wishlist button.

54.2 Requests and Responses

Check Wishlist



Request: [GetProfileWishList](#) (from COMPASS RT API)

Gets the profile's wish list. No parameters are required for checking whether the content item is already on the list.

Response: [WishListRecommendation](#) (from COMPASS RT API)

The returned entity lists all items in the profile's wish list. The significant property is:

Property	Description
<i>externalContentId</i>	The external ID of the movie asset in the Content Library.

54.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Manage a wish list	53 Manage a Wish List (p.160)
Add items to a wish list	55 Add to the Wish List (p.165)
Remove items from a wish list	56 Remove from the Wish List (p.167)
Display a wish list	156 Show My Wishlist (p.426)

54.4 Example

`https://<operator>.tvaas.com/compass/GetProfileWishList?client=json`

This request calls for the list of the profile's Wish List items.

The sample response shows that the movie Cinderella is on the user's wish list.

```
"images": [
  {
    "name": "HorizontalImage",
    "value": "/attachments/CinderellaOutHorizontalImage.jpg"
  },
  {
    "name": "VerticalImage",
    "value": "/attachments/CinderellaVerticalImage.jpg"
  }
],
"prLevel": 3,
"availabilities": [],
"blend": "Default",
"prName": "TV-14",
"ContentProperties": [],
"recommended": 1,
"addToListTime": 1456033917566,
"engine": "promotions",
"genres": [
  {
    "name": "Fantasy",
    "externalId": "Fantasy",
    "id": "Fantasy"
  },
  {
    "name": "Drama",
    "externalId": "Drama",
    "id": "Drama"
  },
  {
    "name": "Family",
    "externalId": "Family",
    "id": "Family"
  }
],
"name": "Cinderella",
"externalContentId": "Cinderella",
"id": 22301,
"contentType": "Movie"
```

55 Add to the Wish List

If the item is not on the Wish List, allow the user to add it.

55.1 Requests and Responses



Request: [AddToWishList](#) (from COMPASS RT API)

Adds a content item to the profile's wish list. You can include additional parameters to inform the content discovery system how the user found this movie:

Parameter	Description
<i>external_content_id</i>	The external ID of the content.
<i>recommended</i>	Set to True or False , if the movie was clicked from a list of recommendations or not.
<i>engine</i>	Specifies the recommendation engine that generated the recommendation, such as <i>most_popular</i> or <i>user_rating</i> .

For more parameters see the COMPASS RT API.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the *AddtoWishList* action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates success or failure.

55.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Manage a wish list	53 Manage a Wish List (p.160)
Check whether a movie is already on the this wish list	54 Check the Wish List (p.162)
Remove items from a wish list	56 Remove from the Wish List (p.167)
Display a wish list	156 Show My Wishlist (p.426)

55.3 Example

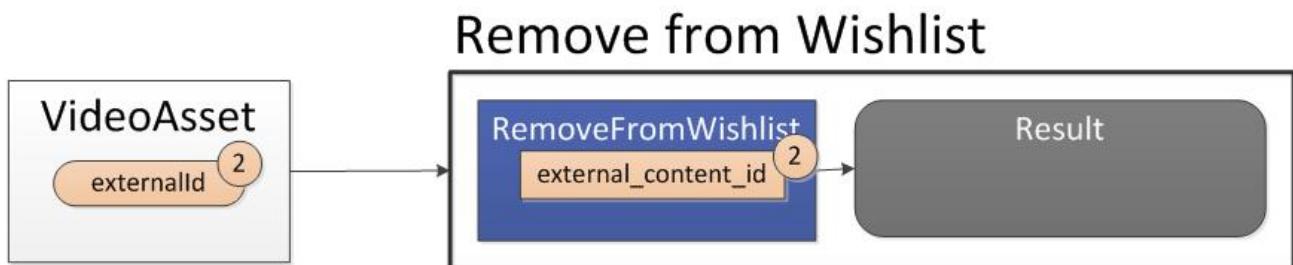
`https://<operator>.tvaas.com/compass/AddToWishList?external_content_id=Argo&client=json`

This request adds the movie Argo to the profile's wish list.

56 Remove from the Wish List

If the item is on the Wish List, allow the user to remove it.

56.1 Requests and Responses



Request: [RemoveFromWishList](#) (from COMPASS RT API)

Removes the content item from the profile's wish list.

Parameter	Description
<i>external_content_id</i>	The external ID of the content item to be removed from the wishlist.

For more parameters see the COMPASS RT API.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the RemoveFromWishList action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates success or failure.

56.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Manage a wish list	53 Manage a Wish List (p.160)

Check whether a movie is already on the this wish list	54 Check the Wish List (p.162)
Add items to a wish list	55 Add to the Wish List (p.165)
Display a wish list	156 Show My Wishlist (p.426)

56.3 Example

`https://<operator>.tvaas.com/compass/RemoveFromWishList?external_content_id=Argo&client=json`

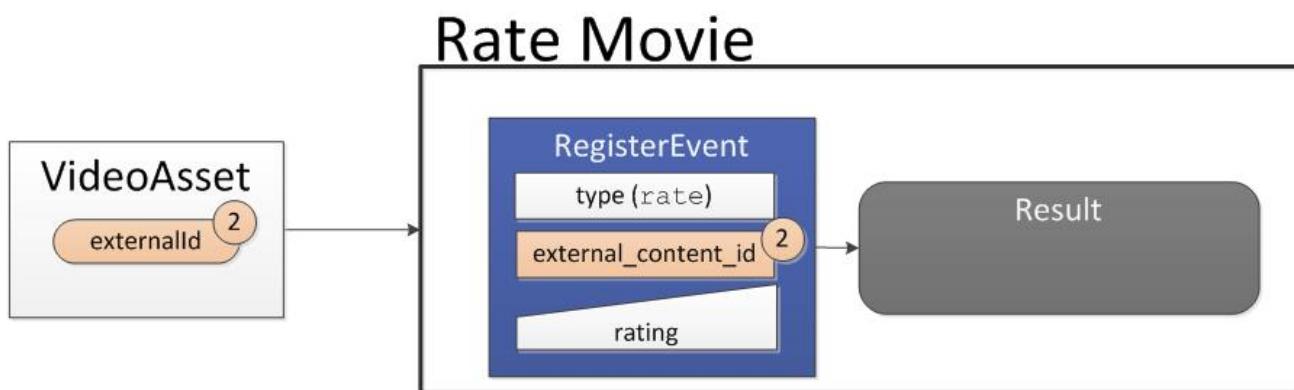
This request removes the movie Argo from the profile's wish list.

57 Rate a Movie

57.1 Overview

Users should be able to rate the movie on the Movie Screen. Ratings are on a scale of 1 (poor) to 5 (good). Only logged in users can rate a movie.

57.2 Requests and Responses



Request: [RegisterEvent](#) (from COMPASS RT API)

Registers the rating the user assigns. Significant parameters include:

Parameter	Description
<i>external_content_id</i>	The name of the movie.
<i>type</i>	Set to <i>rate</i> .
<i>rating</i>	The rating on a scale of 1 to 5.

For more parameters see the COMPASS RT API.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the *RegisterEvent* action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates success or failure.

57.3 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)

57.4 Example

`https://<operator>.tvaas.com/compass/RegisterEvent?type=rate&external_content_id=Argo&rating=3&client=json`

This request registers a rating of 3 for the movie Argo.

58 Get Recommended Movies

58.1 Overview

A Movie screen should list of recommended movies.

The screenshot shows a movie detail page for 'Bridge of Spies'. At the top, there's a navigation bar with links for Home, Subscription VOD, Rent or buy VOD, VOD packages, Live TV, and TV Guide. The 'Home' link is underlined. On the right side of the header are a search icon, a 'Sign in' button, and a language selection dropdown set to 'EN'. Below the header, a back arrow and the title 'Bridge of Spies' are displayed. To the left is a large movie poster featuring a close-up of a man's face with an American flag in the background. To the right of the poster, movie details are listed: 'Available in Oscar 2016, eST, Rent, oscar 2016', 'Compatible devices PC, AndroidSmartphone, iPad, iPhone, AndroidTablet', a 3-star rating, 'Directed by Steven Spielberg', '2015 | PG-13 | Drama, History, Thriller', and 'Cast: Tom Hanks'. A detailed plot summary follows: 'In the cold war, a lawyer, James B. Donovan is recruited by the CIA and involved in an intense negotiation mission to release and exchange a CIA U-2 spy-plane pilot, Francis G. Powers. The pilot was arrested alive after his plane was shot down by the Soviet Union during a mission and stays in the company of a KGB intelligence officer, Rudolf Abel, who was arrested for espionage in the US.' Below the plot summary are several interactive buttons: 'Purchase', 'Watch trailer', 'Add to wishlist', 'My Rating', 'Share with Facebook', and 'Share with Twitter'. At the bottom, a 'Recommended' section displays four movie thumbnails: 'Forrest Gump' (Drama, Family), 'The Revenant' (Thriller, Drama, Adventure), 'Toy story 3' (Comedy, Adventure, Animation), and 'A History of Violence' (Crime, Thriller, Drama).

Similarly, an Episode screen should show a list of recommended series seasons.





58.2 Requests and Responses

The *GetVideoRecommendationList* call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.

58.3 Recommendation Requests – Movie Screen

The table below provides best-practice examples of recommendation calls for a Movie screen.

Recommendations List	Type	Blend	Content Type
Related Content	related_content		
<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?type=related_content&params=external_content_id:Argo&max_results=100&max_pr_level=15&from=0&count=15&device_models=IPad&random=false&client=json</code>			
Recommendations		more_recommendations	Movie
<code>https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=more_recommendations&params=video_id:Argo&content_type=Movie&subscription=false&max_results=100&max_pr_level=15&from=0&count=15&device_models=IPad&random=false&client=json</code>			

58.4 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)

59 Create an Episode Screen

59.1 Overview

The Episode screen is essentially a Movie screen, as described in [33 Create a Movie Screen \(p.106\)](#). Only the recommendation calls differ.

The screenshot shows the Viaccess-Orca website interface. At the top, there's a navigation bar with links for Home, Subscription VOD, Rent or buy VOD, VOD packages, Live TV, and TV Guide. On the far right, there are 'Sign in' and 'EN' language selection buttons. Below the navigation, a search bar is present. The main content area is titled 'Breaking Bad S04EP01'. To the left of the title is a promotional image for the episode, featuring a man in a green shirt and white shorts standing in a desert landscape. The title 'Breaking Bad' is prominently displayed in the center of the image. Below the image, text indicates it's available for rent on compatible devices (PC, Android, Smartphone, iPad, iPhone, Android Tablet), was released in 2012, is rated TV-14, and is a Drama, Crime genre. It also mentions the cast (David Crane, Bryan Cranston, Aaron Paul) and a plot summary about Walt and Jesse being held captive by Gus. Below the plot summary are social sharing buttons for Purchase, Watch trailer, Add to wishlist, My Rating, Share with Facebook, and Share with Twitter. Further down, a 'Recommended' section lists other TV shows: Law and Order (Drama, Crime), The Wire (Drama, Crime), The Sopranos S2 (Drama, Crime), and Weeds (Comedy, Crime, Drama). Each recommended show has a small thumbnail image and its title and genre below it.

59.2 Related Tutorials

To	Refer to Tutorial
Create a Movie screen	33 Create a Movie Screen (p.106)
Create a Season screen	61 Create a VOD Season Screen (p.179)
Create a Series screen	60 Create a TVOD TV Show Series Screen (p.175)
Get a list of recommendations	62 Get Recommended TVOD Seasons (p.184)



60 Create a TVOD TV Show Series Screen

60.1 Overview

A TV Show Series screen displays information about a TV Show series. It should include:

1. An image and description.
2. A list of seasons within the series. See [22 Get Seasons in a TVOD TV Show Series \(p.66\)](#).
3. A list of recommended content. See [62 Get Recommended TVOD Seasons \(p.184\)](#).

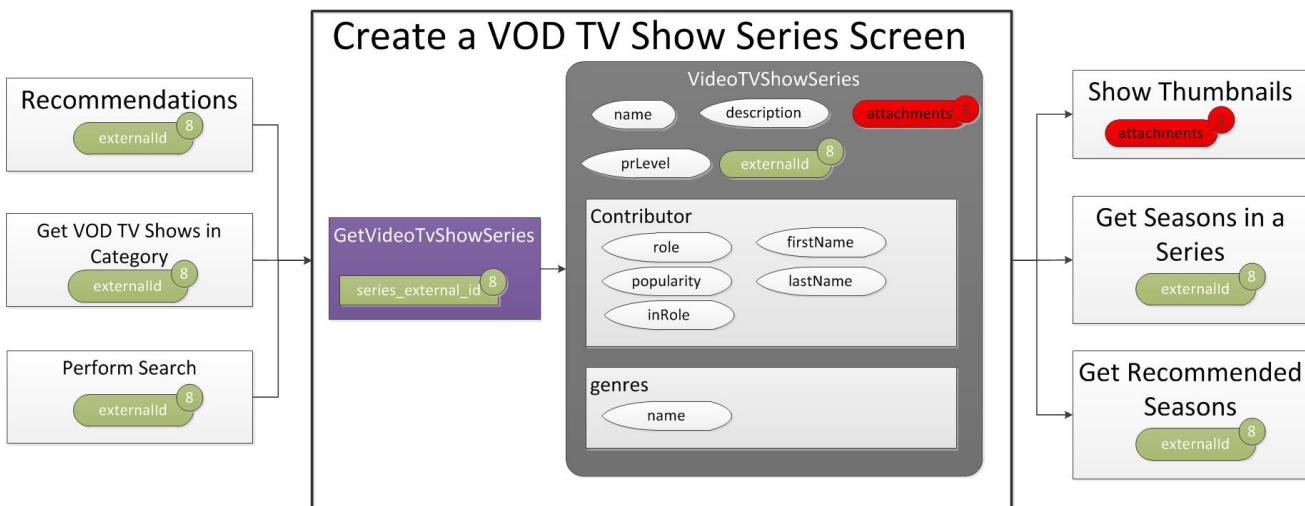
The screenshot shows a mobile application interface for a TV show series. At the top, there is a navigation bar with tabs: Home, Subscription VOD, Rent or buy VOD, VOD packages, Live TV, and TV Guide. Below the navigation bar, the title "Breaking Bad" is displayed in a large, bold font. To the left of the title is a back arrow icon labeled "Back".

1. Image and Description: On the left side of the screen is a promotional image for "Breaking Bad". The image features a man in a green shirt and shorts standing in a desert landscape, holding a gun. The title "Breaking Bad" is overlaid on the image. Below the image, text reads "Change the Equation" and "Series Premiere Jan 20 Sundays 10pm AMC". To the right of the image is a purple circular badge with the number "1". Below the badge is a detailed description of the show: "When chemistry teacher Walter White is diagnosed with Stage III cancer and given only two years to live, he decides what he has nothing to lose. He lives with his teenage son, who has cerebral palsy, and his wife, in New Mexico. Determined to ensure that his family will have a secure future, Walt embarks on a career of drugs and crime. He proves to be remarkably proficient in this new world as he begins manufacturing and selling methamphetamine with one of his former students. The series tracks the impacts of a fatal diagnosis on a regular, hard working man, and explores how a fatal diagnosis affects his morality and transforms him into a major player of the drug trade."

2. Seasons: Below the description, there is a section titled "Seasons". It shows a grid of five thumbnail images, each representing a season of "Breaking Bad". The first thumbnail is labeled "4" and the second is labeled "5". To the right of the thumbnails is a purple circular badge with the number "2".

3. Recommended: Below the seasons section, there is a section titled "Recommended". It shows a grid of four thumbnail images, each representing a recommended TV show. The first two thumbnails are for "The Sopranos" (Season 2) and the last two are for "Six Feet Under" (Season 2). Each thumbnail includes the show's name, season number, and genre. To the right of the thumbnails is a purple circular badge with the number "3".

60.2 Requests and Responses



Request: GetVideoTvShowSeries (from RiGHTv Front-end API)

This request gets all metadata on the relevant TV Show.

Parameter	Description
<i>series_external_id</i>	The external ID of the TV Show in the Content Library.

For more parameters, see the RiGHTv Front-end Integration Guide.

Response: VideoTVShowSeries (from RiGHTv Front-end API)

The returned entity includes a long list of properties, including:

Parameter	Description
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>name</i>	The name of the TV Show.
<i>description</i>	A description of the TV Show.
<i>genres</i>	Lists the names of all genres assigned to the TV Show.
<i>prlevel</i>	The parental rating level of the TV Show.
<i>externalId</i>	The external ID of the TV Show in the Content Library.

60.3 Related Tutorials

To	Refer to Tutorial
Get a list of seasons in the TV Show	22 Get Seasons in a TVOD TV Show Series (p.66)
Get a list of recommendations	62 Get Recommended TVOD Seasons (p.184)
Display attachment images	149 Display Images (p.402)

60.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetVideoTvShowSeries?series_external_id=Weeds&language_code=en&client=json`

This request calls for metadata on the series Weeds.

A sample response is shown below.

```
"metadata": {
    "request": "GetVideoTvShowSeries",
    "timestamp": 1516617698377
},
"response": {
    "isMiniSeries": false,
    "studio": "",
    "prLevel": 0,
    "attachments": [
        {
            "responseElementType": "Attachment",
            "assetId": "",
            "name": "VerticalImage",
            "assetName": "",
            "value": "/attachments/Vertical.jpg"
        },
        {
            "responseElementType": "Attachment",
            "assetId": "",
            "name": "HorizontalImage",
            "assetName": "",
            "value": "/attachments/Horizontal.jpg"
        }
    ],
    "year": 2012,
    "prName": "NR",
    "externalId": "Weeds",
}
```

```
"description": "Widowed suburbanite Nancy Botwin (Mary-Louise Parker) starts growing and selling marijuana to make enough money to support her family after her husband's unexpected death leaves her in big debt. Nancy has associates in her pot-growing business, which include Andy, who is her brother-in-law, and pot-smoking City Councilman Doug Wilson. Then there's the business of trying to keep the whole thing secret from people like her neighbor Celia.",  
"shortDescription": "",  
"countries": [],  
"contentProviderName": "",  
"numberOfSeasons": 2,  
"responseElementType": "VideoTvShowSeries",  
"genres": [],  
"name": "Weeds",  
"extrafields": [],  
"parentalAdvisories": "",  
"contributors": [],  
"shortName": "",  
"contentProviderExternalId": "",  
"seriesCategoryInfoList": [  
  {  
    "template": "TV_Shows_Series",  
    "responseElementType": "TvShowCategoryInfo",  
    "attachments": [],  
    "extrafields": [],  
    "externalId": "Weeds1",  
    "description": "",  
    "status": 2  
  }  
]
```



61 Create a VOD Season Screen

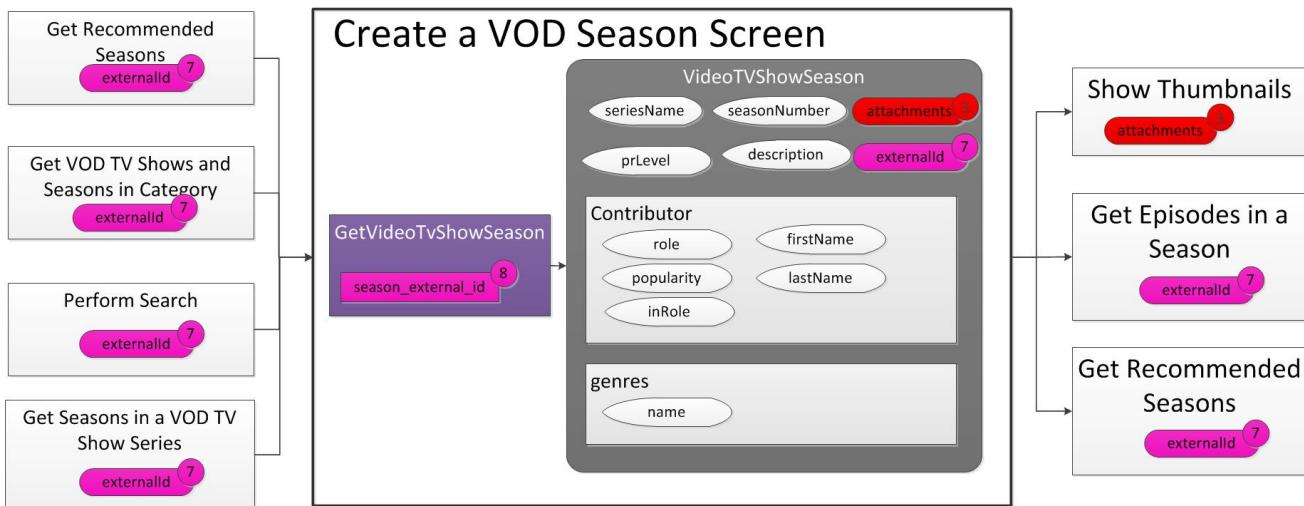
61.1 Overview

A Season screen displays information about a TV Show season. It should include:

1. An image and description.
2. A list of episodes within the season. See [23 Get Episodes in a TVOD Season \(p.71\)](#).
3. A list of recommended content. See [62 Get Recommended TVOD Seasons \(p.184\)](#).

The screenshot shows a VOD season screen for the TV show 'Breaking Bad'. At the top, there's a navigation bar with links: Home, Subscription VOD, Rent or buy VOD, VOD packages, Live TV, and TV Guide. Below the navigation is a back button labeled '< Back'. The main content area features a large thumbnail for 'Breaking Bad' S04EP01, which includes a promotional image of Walter White and the text 'Change the Equation' and 'Series Premiere Jan 20 Sundays 10pm AMC'. To the right of the thumbnail is a purple circular badge with the number '1'. Below the thumbnail is a detailed description of the season: 'Crime, Drama' followed by a paragraph of text. Further down, there are sharing options: 'Share with Facebook' (with a 'f' icon) and 'Share with Twitter' (with a bird icon). The next section, 'Episodes', contains two episode thumbnails: 'Breaking Bad S04EP01' and 'Breaking Bad S04EP02', each with a purple circular badge containing the number '2'. The final section, 'Recommended', lists four other TV shows with their thumbnails: 'Law and Order' (Drama, Crime), 'The Wire' (Drama, Crime), 'The Sopranos S2' (Drama, Crime), and 'Weeds' (Comedy, Crime, Drama). Each recommended show has a purple circular badge with the number '3'.

61.2 Requests and Responses



Request: [GetVideoTvShowSeason](#) (from RiGHTv Front-end API)

This request gets all metadata on the relevant season.

Parameter	Description
<i>season_external_id</i>	The external ID of the season in the Content Library.

For more parameters, see the RiGHTv Front-end Integration Guide.

Response: [VideoTvShowSeason](#) (from RiGHTv Front-end API)

The returned entity includes a long list of properties, including:

Parameter	Description
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>description</i>	A description of the season.
<i>genres</i>	Lists the names of all genres assigned to the season.
<i>prlevel</i>	The parental rating level of the season.
<i>externalId</i>	The external ID of the season in the Content Library.

Parameter	Description													
<i>Contributors</i>	A list of contributors to the season.													
	<table border="1"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>role</i></td> <td>Actor, Director, etc.</td> </tr> <tr> <td><i>popularity</i></td> <td>The numeric value indicating the popularity of the contributor in the context of the specific content item.</td> </tr> <tr> <td><i>firstName</i></td> <td>Contributor first name</td> </tr> <tr> <td><i>lastName</i></td> <td>Contributor last name</td> </tr> <tr> <td><i>inRole</i></td> <td>Information about the contributor's role in the movie</td> </tr> </tbody> </table>	Property	Description	<i>role</i>	Actor, Director, etc.	<i>popularity</i>	The numeric value indicating the popularity of the contributor in the context of the specific content item.	<i>firstName</i>	Contributor first name	<i>lastName</i>	Contributor last name	<i>inRole</i>	Information about the contributor's role in the movie	
Property	Description													
<i>role</i>	Actor, Director, etc.													
<i>popularity</i>	The numeric value indicating the popularity of the contributor in the context of the specific content item.													
<i>firstName</i>	Contributor first name													
<i>lastName</i>	Contributor last name													
<i>inRole</i>	Information about the contributor's role in the movie													
<i>seriesName</i>	The name of the TV show series.													
<i>seasonNumber</i>	The number of the season in the series. The name of the season to display should be constructed as: <i>seriesName</i> : Season <i>seasonNumber</i> .													

61.3 Related Tutorials

To	Refer to Tutorial
Get a list of episodes in the season	22 Get Seasons in a TVOD TV Show Series (p.66)
Get a list of recommendations	62 Get Recommended TVOD Seasons (p.184)
Display attachment images	149 Display Images (p.402)

61.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetVideoTvShowSeason?season_external_id=Weeds_season_7&language_code=en&client=json`

This request calls for metadata on season 7 of the series Weeds.



A partial response is shown below:

```
"metadata": {  
    "request": "GetVideoTvShowSeason",  
    "timestamp": 1516618273146  
},  
"response": {  
    "studio": "",  
    "prLevel": 3,  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "VerticalImage",  
            "assetName": "",  
            "value": "/attachments/weeds-season-7-three-years-later.jpg"  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "HorizontalImage",  
            "assetName": "",  
            "value": "/attachments/weeds_horizontal.jpg"  
        }  
    ],  
    "seasonCategoryInfoList": [  
        {  
            "template": "TV_Shows_Season",  
            "responseElementType": "TvShowCategoryInfo",  
            "attachments": [],  
            "extrafields": [],  
            "externalId": "52574354",  
            "description": "",  
            "status": 2  
        },  
    ]  
}
```

```
[  
    {  
        "template": "TV_Shows_Season",  
        "responseElementType": "TvShowCategoryInfo",  
        "attachments": [],  
        "extrafields": [],  
        "externalId": "Weeds7",  
        "description": "",  
        "status": 2  
    }  
,  
    "year": 2011,  
    "seriesName": "Weeds",  
    "prName": "TV-14",  
    "externalId": "Weeds_season_7",  
    "description": "The seventh season of Weeds premiered on June 27  
    , 2011 at 10 E/9 C on the television cable network Showtime,  
    and consisted of 13 episodes.",  
    "shortDescription": "",  
    "seasonNumber": 7,  
    "countries": [],  
    "seriesExternalId": "Weeds",  
    "contentProviderName": "",  
    "numberOfEpisodes": "",  
    "responseElementType": "VideoTvShowSeason",  
    "genres": [],  
    "name": "Weeds season 7",  
    "extrafields": [],  
    "parentalAdvisories": "",  
    "contributors": [],  
    "shortName": "",  
    "contentProviderExternalId": ""  
]
```



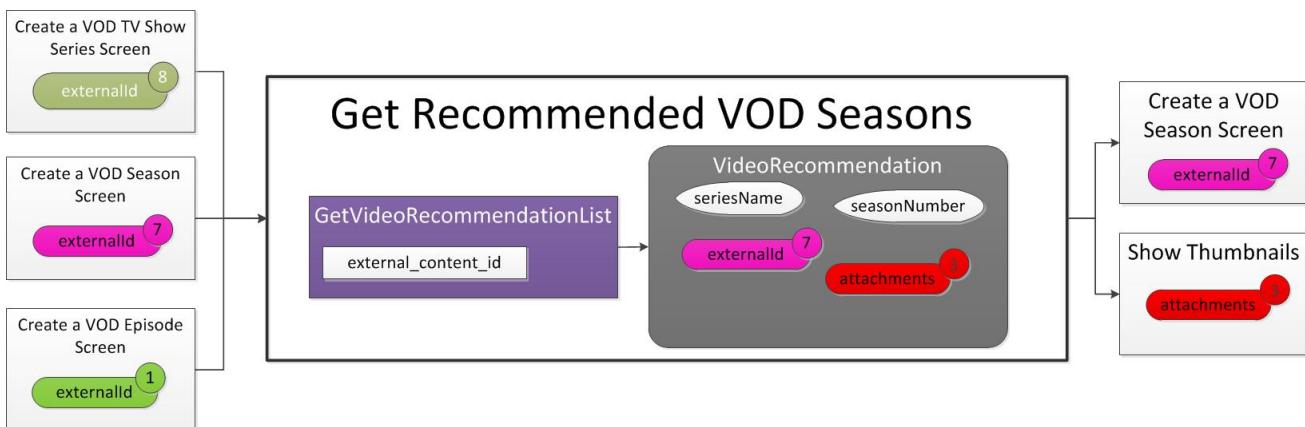
62 Get Recommended TVOD Seasons

62.1 Overview

TV Show Series, Sseason and Episode screens should all list recommended content.

62.2 Requests and Responses

The *GetVideoRecommendationList* call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.



Request: [GetVideoRecommendationList \(from COMPASS RT API\)](#)

The request for recommendations for Seasons to display on a TV Show Series screen, a Season screen and an Episode screen is as follows:

```
https://<operator>.tvaas.com/compass/
GetVideoRecommendationList? blend=more_recommendations&params=external_
content_id:{content_id}&content_type=Season&subscription=false&max_
results={max_results}&max_pr_level={max_pr_level}&random=
{random}&language_code={language_code}&client=json
```

The parameters are described below:

Parameter	Description								
<i>blend</i>	The name of a blend according to which the recommendation results from various recommendation engines are blended together. Set to <i>more_recommendations</i> .								
<i>external_content_id</i>	The ID of the item based on which recommendations should be retrieved. The table below explains which content ID should be used depending on the screen being displayed: <table border="1" data-bbox="520 1051 1432 1432"> <thead> <tr> <th>Screen</th><th>Content ID</th></tr> </thead> <tbody> <tr> <td>TV Show Series</td><td>The content ID of the first season in the series.</td></tr> <tr> <td>Season</td><td>The content ID of the season.</td></tr> <tr> <td>Episode</td><td>The content ID of the episode (video asset).</td></tr> </tbody> </table>	Screen	Content ID	TV Show Series	The content ID of the first season in the series.	Season	The content ID of the season.	Episode	The content ID of the episode (video asset).
Screen	Content ID								
TV Show Series	The content ID of the first season in the series.								
Season	The content ID of the season.								
Episode	The content ID of the episode (video asset).								
<i>content_type</i>	The type of content to recommend. Set to <i>Season</i> .								
<i>subscription</i>	Set to <i>false</i> to include seasons to which the account is not subscribed. Set to <i>true</i> to include only seasons to which the account is already subscribed.								

Response: [VideoRecommendation \(from COMPASS RT API\)](#)

The returned entity includes a long list of properties, including:



Parameter	Description
<i>images</i>	The <i>name</i> and <i>value</i> (path and file name) of the posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
<i>externalContentId</i>	The external content ID of the recommended content.
<i>seriesName</i>	The name of the TV Show Series that the season is part of.
<i>contentSeq</i>	The number of the season in the series. Build the name of the season for display as: <i>seriesName</i> : Season <i>contentSeq</i>

62.3 Example

`https://<operator>.tvaas.com/compass/GetVideoRecommendationList?blend=more_recommendations&subscription=false&random=false&client=json&content_type=season¶ms=external_content_id%3AWeeds_season_7`

This request calls for recommended seasons based on season 7 of the series Weeds.

The response for one season returned is displayed below:

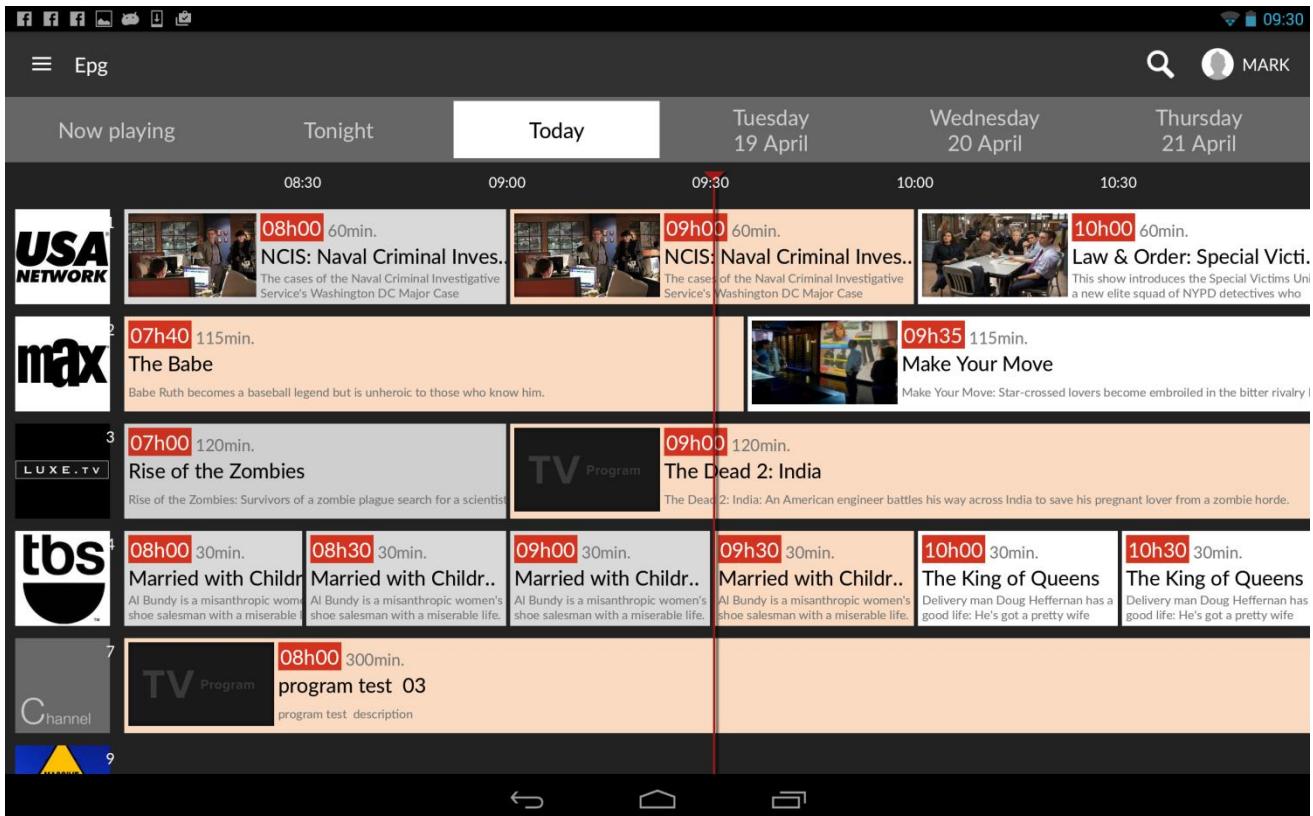


```
"contentSeq": 2,  
"images": [  
    {  
        "name": "HorizontalImage",  
        "value": "/attachments/TDBar640.jpg"  
    },  
    {  
        "name": "VerticalImage",  
        "value": "/attachments/  
            /430894_mkt_pa_truedetective_s2_vince_po.jpg"  
    }  
],  
"prLevel": 5,  
"seriesName": "True detective",  
"availabilities": [],  
"prName": "TVMA",  
"seriesRatersCount": 0,  
"type": "Fallback",  
"ContentProperties": [],  
"recommendationReasons": [],  
"genres": [  
    {  
        "name": "Crime",  
        "externalId": "Crime",  
        "id": "Crime"  
    }  
],  
"name": "True detective season 2",  
"externalContentId": "True_detective_season_2",  
"id": 24234,  
"externalSeriesId": "True_detective",  
"contentType": "Season",  
"seriesRating": 0
```

63 Create a TV Guide

63.1 Overview

The TV Guide (also called an EPG – Electronic Program Guide), displays the program line-up of all channels along a timeline.



63.2 Basic Display Guidelines

The following are recommended guidelines for constructing the TV Guide:

- The TV GUIDE should be displayed as a grid.
- Display each channel on its own row. The row should show an icon for the channel, followed by the list of scheduled programs along the timeline. See [64 Get a List of Channels \(p.190\)](#) and [65 Get List of Channel Programs \(p.193\)](#).
- For each program, its name and thumbnail should be displayed. See [65 Get List of Channel Programs \(p.193\)](#).
- If gaps exist between programs in the timeline, they should be filled with text such as “Information Unavailable”.
- Channels should be listed in order according to the channel number.
- Display all channels, except for Adult channels which should only be displayed to a logged-in user.
- Optionally, only display channels appropriate for the device (terminal) currently in use.

- Channels to which a logged-in user is subscribed should be indicated as such. See [66 Identify Subscribed Channels \(p.197\)](#).
- Optionally, indicate programs for which Start Over is supported. See [163 Identify Start Over Enabled Programs \(p.454\)](#).
- Optionally, indicate programs for which Catch-Up is supported. See [173 Identify Catch-Up Enabled Programs \(p.483\)](#). If the operator offers Catch-Up functionality, the user should be able to scroll the TV Guide back in time to be able to access past programs.

63.3 Additional Display Guidelines

The following are further ideas for improving the user's experience of the TV Guide:

- Limit the timeline to a period of the next seven days.
- Allow navigation to a specific day, possibly providing a single-day view of the TV Guide.
- Highlight programs currently being broadcast.

63.4 Functionality Requirements

The following are recommended guidelines for functionality to provide in the TV Guide:

- User can select any program. If the user:
 - Is subscribed to the channel, the Program screen is shown.
 - Is not subscribed to the channel, a screen is shown indicating that the channel is not subscribed to, and lists the packages through which the channel can be subscribed to.
- User can add the channel to their Favorites list.

63.5 Related Tutorials

To	Refer to Tutorial
Get a list of channels	64 Get a List of Channels (p.190)
Get a list of programs on each channel	65 Get List of Channel Programs (p.193)
Get a list of channels to which the account is subscribed	66 Identify Subscribed Channels (p.197)
List offers through which access to a live channel can be purchased	72 List Offers for a Channel (p.211)
Play a live channel	76 Playing Live Programs (p.220)
Indicate which programs support Start Over	163 Identify Start Over Enabled Programs (p.454)
Indicate which programs support Cath-Up	173 Identify Catch-Up Enabled Programs (p.483)

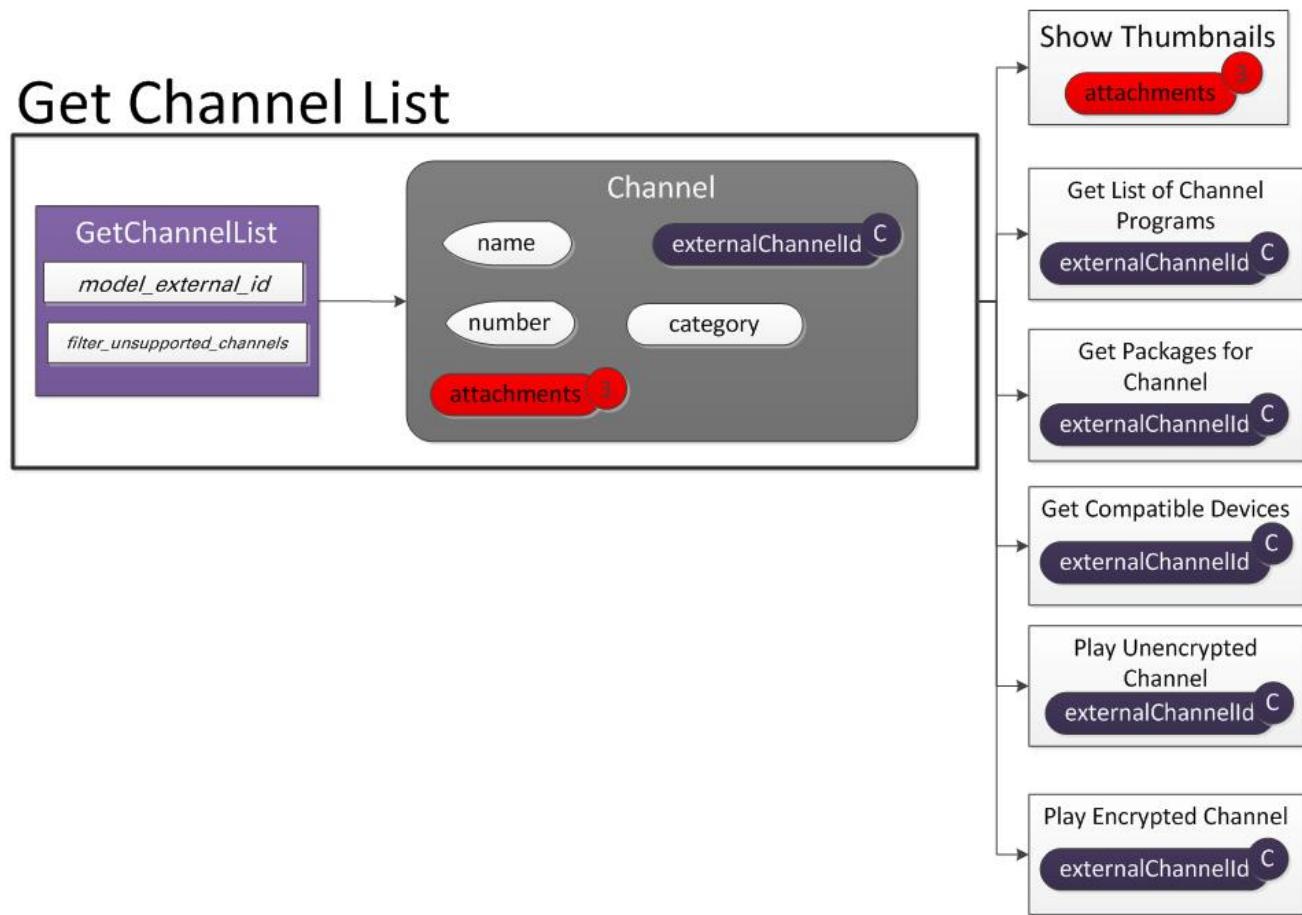


64 Get a List of Channels

64.1 Overview

The first step in building a TV Guide is to get a list of all available channels.

64.2 Requests and Responses



Request: [GetChannelList](#) (from RiGHTv Front-end API)

Gets a list of channels with a range of information about each one. Significant parameters include:

Property	Description
<i>model_external_id</i>	The ID of the device in use.
<i>filter_unsupported_channels</i>	If set to true, limits the list of channels returned to include only those supported by the device in use.
<i>max_pr_level</i>	Specifies the maximum parental rating level of channel to retrieve. Specified as a number from 0 (lowest) to 999 (highest).

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: Channel (from RiGHTv Front-end API)

Lists all channels that match the parameters specified by *GetChannelList*, and provides a range of information on each channel listed. Properties relevant to the TV Guide include:

Property	Description
<i>externalChannelId</i>	The ID the channel.
<i>name</i>	The name of the channel
<i>number</i>	The number of the channel
<i>attachment</i>	An array of attachments to the channel, which will typically include a poster for use as an icon on the TV GUIDE.
<i>category</i>	The category of the channel, to allow for channel grouping by category.

For more properties, see the RiGHTv Front-end API Developers Guide.

64.3 Related Tutorials

64.4 Performance Guidelines

Excessive calls for channel information can harm performance and they should thus be issued efficiently. Some recommendations are provided below:

- The application should keep all TV Guide information in a cache which should refresh periodically.
- The application should only call for and display as many channels as can be listed at one time. If the user scrolls down, call for and display the next group of channels.

64.5 Example

`https://<operator>.tvaas.com/RTEFacade/GetChannelList?model_external_id=IPad&filter_unsupported_channels=true&client=json`



This request gets a list of all channels that are viewable on an iPad.

An extract of a sample response is shown below.

```
{  
    "prLevel": 0,  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "VerticalImage",  
            "assetName": "",  
            "value": "/attachments/USA_networks_vertical.jpg"  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "HorizontalImage",  
            "assetName": "",  
            "value": "/attachments/usa-network-horizontal.jpg"  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "SquareImage",  
            "assetName": "",  
            "value": "/attachments/Usa_square(1).jpg"  
        }  
    ],  
    "externalChannelId": "France24En",  
    "flags": 44,  
    "description": "France24En",  
    "bitrate": "",  
    "interactiveUrl": "",  
    "responseElementType": "Channel",  
    "number": 1,  
    "simultaneousViewsLimit": 3,  
    "affiliation": "",  
    "contentDefinition": "",  
    "id": 8000,  
    "serviceId": "",  
    "identifier": "",  
    "ip": "",  
    "prName": "NR",  
    "originalNetworkId": "",  
    "encoding": "HLS",  
    "url": "http://olive.fr.globecast.tv/live/ramdisk/france24_fra/hls_video/index.m3u8",  
    "transportStreamId": "",  
    "port": 0,  
    "sourceType": "HLS",  
    "name": "Channel 1",  
    "extrafields": [],  
    "category": "News",  
    "longName": ""  
},
```

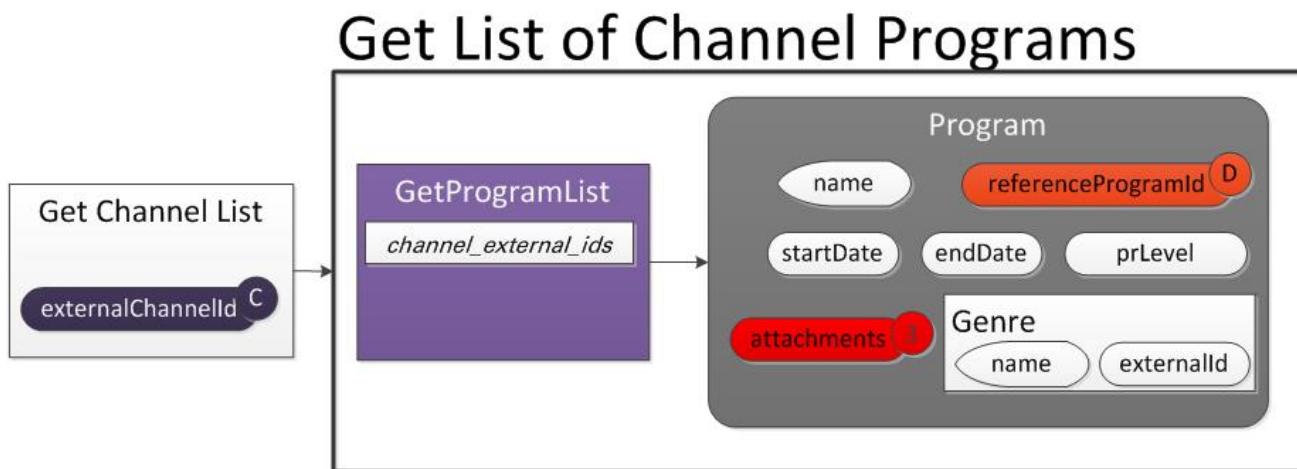


65 Get List of Channel Programs

65.1 Overview

The TV Guide must show the programs to be shown (and optionally that have been shown) per channel.

65.2 Requests and Responses



Request: [GetProgramLists](#) (from RiGHTv Front-end API)

Gets a list of all programs scheduled for the specified channel(s), together with a range of information on each one. Significant parameters include:

Property	Description
<i>channel_external_ids</i>	A list of the ID's of all channels for which program info is required.
<i>start_date</i>	The earliest date from which programs should be returned. Specified in milliseconds from 1/1/1970).
<i>end_date</i>	The latest date from which programs should be returned. Specified in milliseconds from 1/1/1970).

Note

To take advantage of data caching and thereby reduce the load of requests on the RiGHTv server, it is important that standardized start and end dates be used, so that all requests performed around the same time are sent identically. For example: Start and end of day.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: Program (from RiGHTv Front-end API)

Lists all programs that match the parameters specified by *GetProgramLists* action, and provides a range of information on each program listed. Properties relevant to the TV GUIDE include:

Property	Description
<i>referenceProgramId</i>	The ID of the program. This is used to identify the program in other calls.
<i>name</i>	The name of the program.
<i>startDate, endDate</i>	The start and end dates of the program, specified in milliseconds from 1/1/1970.
<i>prLevel</i>	The parental rating of the program (as a number from 0 – lowest, to 999 – highest).
<i>attachment</i>	An array of attachments to the program, which will typically include a poster for use as an icon on the TV Guide.
<i>genres</i>	A list of genres assigned to the episode. For each, its genre <i>name</i> and <i>externalId</i> are returned.

For more properties, see the RiGHTv Front-end API Developers Guide.

65.3 Related Tutorials

To	Refer to Tutorial
Create a TV guide	63 Create a TV Guide (p.188)
Get a list of channels	64 Get a List of Channels (p.190)
Get a list of channels to which the account is subscribed	66 Identify Subscribed Channels (p.197)
List offers through which access to a live channel can be purchased	72 List Offers for a Channel (p.211)
Play a live channel	76 Playing Live Programs (p.220)



To	Refer to Tutorial
Indicate which programs support Start Over	163 Identify Start Over Enabled Programs (p.454)
Indicate which programs support Catch-Up	173 Identify Catch-Up Enabled Programs (p.483)
Display image attachments	149 Display Images (p.402)

65.4 Performance Guidelines

Excessive calls for program information can harm performance and they should thus be issued efficiently. Some recommendations are provided below:

- The application should keep all program information in a cache which should refresh periodically.
- The application should only call for, and then display, the program list for the current day. If the user slides to or selects a future date, a new call should be issued for the relevant day.

65.5 Example

```
https://<operator>.tvaas.com/RTEFacade/GetProgramLists?channel_external_ids=France24Ar; France24En&start_date=1459403282082&end_date=1483167180000&scene=tvaas&client=json
```

This request gets a list of all programs lined up for channels France24An and France24En, between 31/03/2016 and 31/12/2016.

An extract of a sample response is shown below.



```

    "advisories": "",
    "prLevel": 0,
    "attachments": [],
    "endDate": 1460451000000, ←
    "seriesName": "",
    "seriesNumberOfEpisodes": "",
    "prName": "NR",
    "referenceProgramId": "ae7952ee5aonn34cicihpghf0p",
    "flags": 0,
    "description": "The Signal: Three friends believe they've tracked a
                    hacker to a shed in the Nevada desert.",
    "seriesSeason": "",
    "episodeId": "",
    "responseElementType": "Program",
    "price": 0,
    "genres": [],
    "name": "The Signal",
    "id": 12300,
    "shortName": "",
    "startDate": 1460445000000
  },
  {
    "advisories": "",
    "prLevel": 0,
    "attachments": [
      {
        "responseElementType": "Attachment",
        "assetId": "",
        "name": "VerticalImage",
        "assetName": "",
        "value": "/j3e8p6goe4hvo5rau5062kpvmu_VerticalImage.jpg" ←
      },
      {
        "responseElementType": "Attachment",
        "assetId": "",
        "name": "HorizontalImage",
        "assetName": "",
        "value": "/j3e8p6goe4hvo5rau5062kpvmu_HorizontalImage.jpg"
      }
    ],
    "endDate": 1460459400000, ←
    "seriesName": "",
    "seriesNumberOfEpisodes": "",
    "prName": "NR",
    "referenceProgramId": "j3e8p6goe4hvo5rau5062kpvmu",
    "flags": 0,
    "description": "An underdog lawyer takes on a fraudulent Insurance company.",
    "seriesSeason": "",
    "episodeId": "",
    "responseElementType": "Program",
    "price": 0,
    "genres": [
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Crime",
        "externalId": "Crime",
        "id": 4041
      },
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Drama",
        "externalId": "Drama",
        "id": 5057
      },
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Thriller",
        "externalId": "Thriller",
        "id": 12255
      }
    ],
    "name": "The Rainmaker",
    "id": 12296, ←
    "shortName": "",
    "startDate": 1460451000000 ←
  }
]

```

66 Identify Subscribed Channels

66.1 Overview

The TV Guide should indicate which channels the logged-in user is subscribed to.

Customers cannot subscribe to individual channels. Instead, they subscribe to channel packages which include multiple channels.

To build a list of channels to which the logged-in account is subscribed:

1. Get a list of channel packages to which the account is subscribed. (See [67 Get Account Channel Packages \(p.198\)](#)).
2. Get a list of channels in each subscribed package. (See [68 Get Channels in Package \(p.200\)](#)).

Note: The instructions provided here apply whether the account subscribed to a Channel Package directly, or through a Service Plan.

66.2 Related Tutorials

To	Refer to Tutorial
Create a TV guide	63 Create a TV Guide (p.188)
Get a list of Channel Packages to which the account is subscribed	67 Get Account Channel Packages (p.198)
Get a list of channels included in each Channel Package	68 Get Channels in Package (p.200)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)
Create a Program screen	69 Create a Program Screen (p.202)



67 Get Account Channel Packages

67.1 Overview

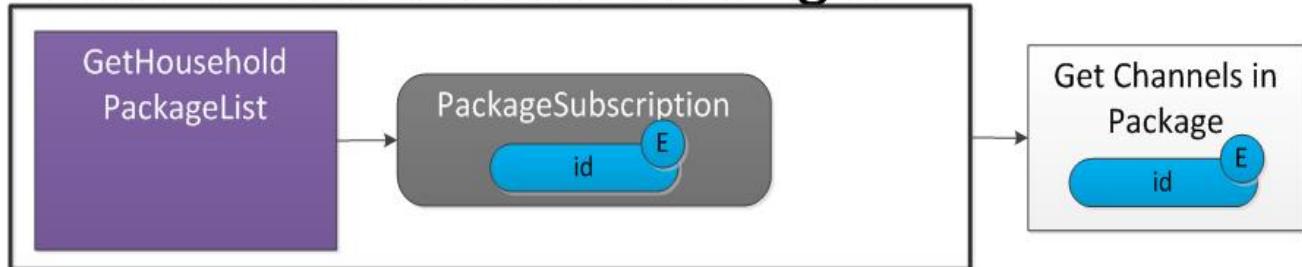
Users cannot subscribe to individual channels, instead they subscribe to channel packages which include multiple channels. This tutorial shows how to get a list of subscribed channel packages.

Note

The instructions provided here apply whether the account subscribed to a Channel Package directly, or through a Service Plan.

67.2 Requests and Responses

Get Account Channel Packages



Request: [GetHouseholdPackageList](#) (from RiGHTv Front-end API)

Gets a list of all channel packages to which the account is subscribed. No parameters are required.

For available parameters, see the RiGHTv Front-end API Developers Guide.

Response: [PackageSubscription](#) (from RiGHTv Front-end API)

Lists all channel packages to which the account is subscribed. The only parameter returned is:

Property	Description
<i>id</i>	The ID of the package subscription.

67.3 Related Tutorials

To	Refer to Tutorial
Create a TV guide	63 Create a TV Guide (p.188)
Identify channels to which the account is subscribed	66 Identify Subscribed Channels (p.197)
Get a list of channels included in each Channel Package	68 Get Channels in Package (p.200)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)
Create a Program screen	69 Create a Program Screen (p.202)

67.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHouseholdPackageList?client=js on>

This request gets a list of all channel packages to which the account is subscribed.

An extract of a sample response is shown below. The account is subscribed to channel package 8005.

```
{
  "metadata": {
    "request": "GetHouseholdPackageList",
    "fullLength": 1,
    "timestamp": 1459419191714
  },
  "response": [
    {
      "responseElementType": "PackageSubscription",
      "id": 8005
    }
  ]
}
```

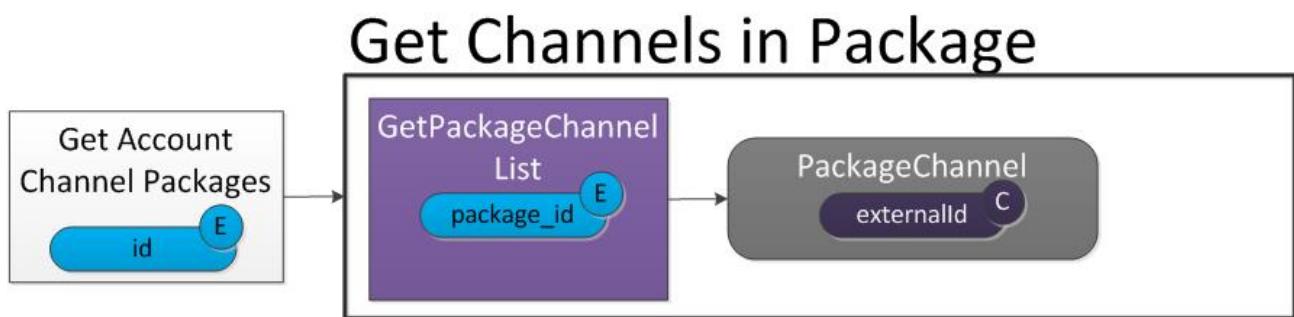


68 Get Channels in Package

68.1 Overview

This tutorial shows how to get a list of all channels in a specified channel package.

68.2 Requests and Responses



Request: [GetPackageChannelList](#) (from RiGHTv Front-end API)

Gets a list of all channel packages that the account is subscribed to. Only one parameter is required:

Parameter	Description
<i>package_id</i>	The ID of the package subscription.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [PackageChannel](#) (from RiGHTv Front-end API)

Lists all channel packages to which the user is subscribed. The only significant parameter returned is:

Property	Description
<i>externalId</i>	The id of a channel in the channel package.

To obtain a list of all channels subscribed to, use *GetPackageChannelList* for each package the account is subscribed to.

68.3 Related Tutorials

To	Refer to Tutorial
Create a TV guide	63 Create a TV Guide (p.188)
Identify channels to which the account is subscribed	66 Identify Subscribed Channels (p.197)
Get a list of Channel Packages to which the account is subscribed	67 Get Account Channel Packages (p.198)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)
Create a Program screen	69 Create a Program Screen (p.202)

68.4 Example

https://<operator>.tvaas.com/RTEFacade/GetPackageChannelList?package_id=8005&client=json

This request gets a list of all channels included in the channel package with id 8005.

An extract of a sample response is shown below, showing two channels included in the channel package.

```
{
  "metadata": {
    "request": "GetPackageChannelList",
    "fullLength": 10,
    "timestamp": 1459419816141
  },
  "response": [
    {
      "responseElementType": "PackageChannel",
      "number": 1,
      "name": "",
      "externalId": "France24En",
      "id": 8000 ←
    },
    {
      "responseElementType": "PackageChannel",
      "number": 2,
      "name": "",
      "externalId": "France24Ar",
      "id": 8009 ←
    }
  ]
}
```



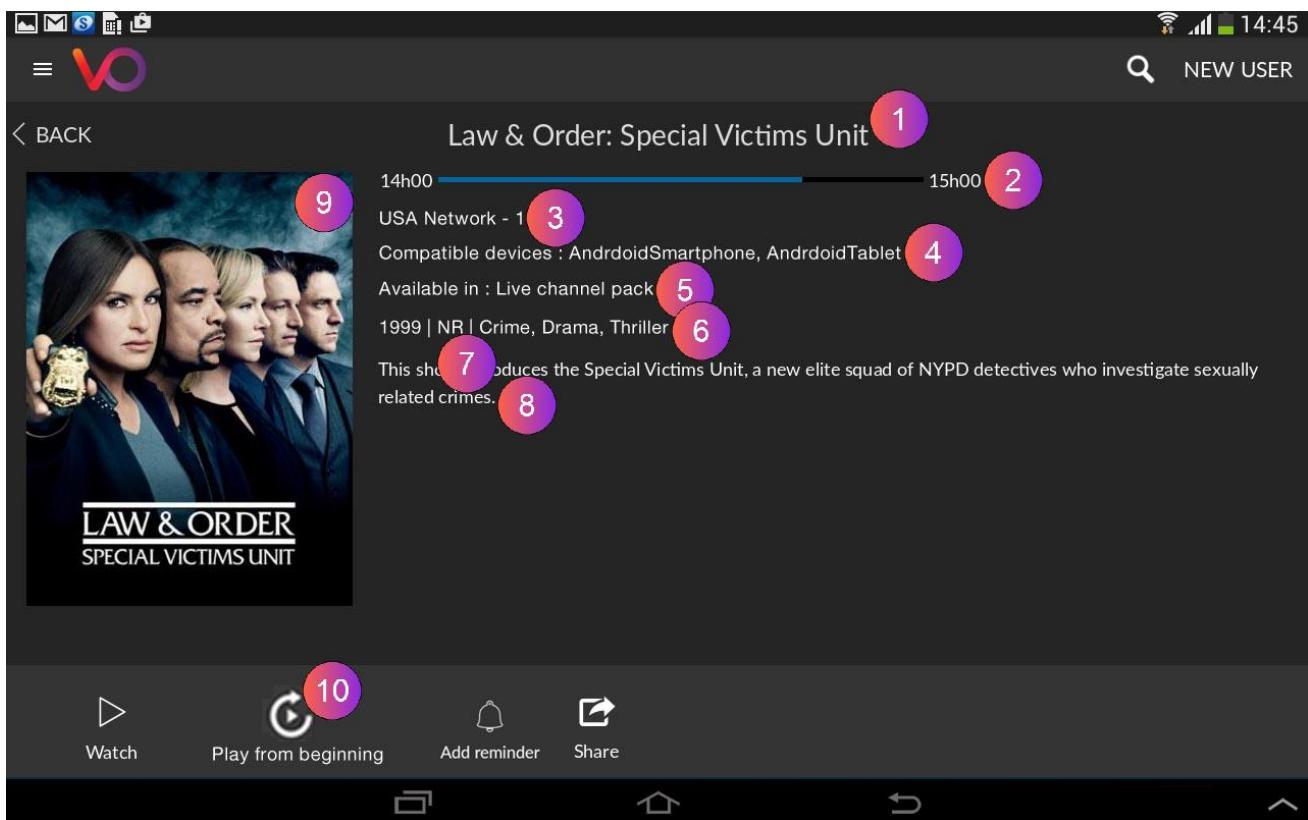
69 Create a Program Screen

69.1 Overview

A Program screen displays detailed information about a live program. Users typically reach the program screen by selecting the program from the TV Guide (see [63 Create a TV Guide \(p.188\)](#)).

69.2 Basic Display Guidelines

The Program screen should display relevant information about the program.

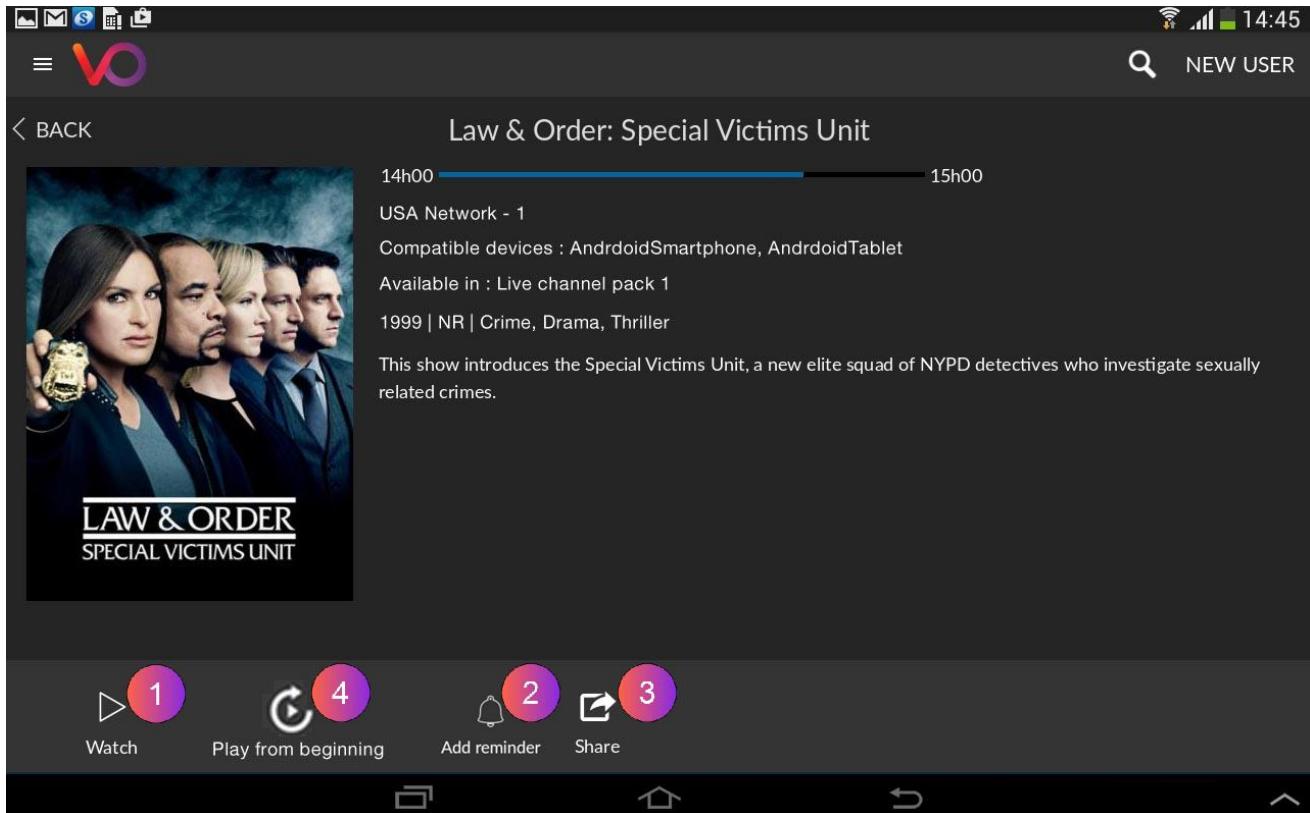


The following should be included:

1. Name (See [70 Get Program Metadata \(p.205\)](#).)
2. Start and end times
3. Channel information
4. A list of compatible devices.
5. A list of Channel Packages and Service Plans through which the program is available. (See [72 List Offers for a Channel \(p.211\)](#).)
6. Genres
7. Parental rating
8. Detailed description
9. Image for the program. (If not available, substitute with the channel image, or some default image.)

10. Whether Start Over is supported. (See [163 Identify Start Over Enabled Programs \(p.454\).](#))
11. Whether Catch-Up is supported. (See [173 Identify Catch-Up Enabled Programs \(p.483\).](#))
12. Whether a reminder has been set. (See [157 Show My Reminders \(p.428\).](#))

69.3 Functionality Requirements



From the Program screen the user should be able to:

1. Begin viewing the program if currently being broadcast. (See [76 Playing Live Programs \(p.220\).](#))
2. Set a reminder for a program that is scheduled for the future. (See [74 Setting Program Reminders \(p.218\)\).](#)
3. Share the episode. (See [194 Sharing Via Facebook \(p.524\)\).](#)
4. Begin viewing the program in Start Over mode, but only if the program supports that and only while the program is being broadcast. (See [163 Identify Start Over Enabled Programs \(p.454\)](#) and [164 Playing a Live Channel through Start Over \(p.458\).](#))
5. Begin viewing the program in Catch-Up mode, but only if the program supports that and only after the broadcast of the program has ended. (See [173 Identify Catch-Up Enabled Programs \(p.483\)](#) and [176 Playing a Program through Catch-Up \(p.492\).](#))

69.4 Related Tutorials

To	Refer to Tutorial
Get most of the program metadata	70 Get Program Metadata (p.205)
Get a list of devices on which the program can be viewed	73 Get Compatible Devices (p.216)
Get a list of recommended content to place on the program screen	75 Get Recommended Live Content (p.219)
Play a live program	76 Playing Live Programs (p.220)
Indicate if program supports Start Over	163 Identify Start Over Enabled Programs (p.454)
Indicate if program supports Catch-Up	173 Identify Catch-Up Enabled Programs (p.483)
Indicate if a reminder has been set for the program	157 Show My Reminders (p.428)



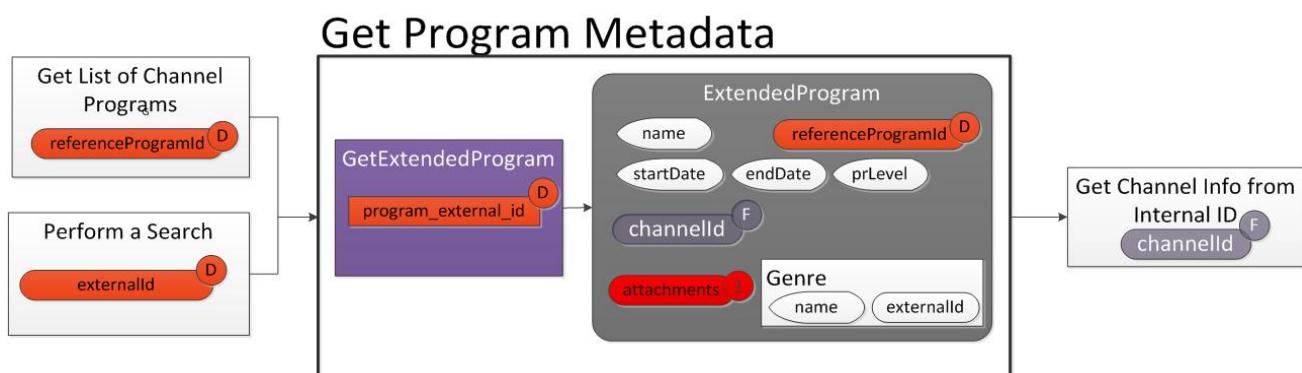
70 Get Program Metadata

70.1 Overview

The Program Screen should display a range of information about the program, such as

1. Name
2. Start and end times
3. Channel information
4. Genres
5. Detailed description

70.2 Requests and Responses



Request: [GetExtendedProgram](#) (from RiGHTv Front-end API)

Gets detailed information on a specified program. Significant parameters include:

Parameter	Description
<code>program_external_id</code>	The external ID of the program.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [ExtendedProgram](#) (from RiGHTv Front-end API)

For the Program screen, significant properties include:

Property	Description
name	The name of the program.
Description	A detailed description of the program.
startDate, endDate	The start and end dates and times, specified in milliseconds from 1/1/1970.
prName	The name of the parental rating level.
prLevel	The numeric level of the parental rating level, where 0 is the lowest rating and 999 is the highest.
attachments	An array of attachments. See 149 Display Images (p.402) for more information.
genres	A list of genres assigned to the episode. For each, its genre <i>name</i> and <i>externalId</i> are returned.
channelId	The internal id of the channel. To learn how to get the channel name for display and the channel external ID for use in other requests, see 71 Get Channel Info from Internal ID (p.208) .

For more properties, see the RiGHTv Front-end API Developers Guide.

70.3 Related Tutorials

To	Refer to Tutorial
Create a Program screen	69 Create a Program Screen (p.202)
Get the name of the channel on which the program is broadcast	71 Get Channel Info from Internal ID (p.208)



70.4 Example

https://<operator>.tvaas.com/RTEFacade/GetProgramLists?channel_external_ids=France24Ar;France24En&start_date=1459403282082&end_date=1483167180000&scene=tvaas&client=json

This request gets a list of all programs lined up for channel France24En, between 31/03/2016 and 31/12/2016.

A sample response is shown below.

```
{
  "advisories": "",
  "prLevel": 0,
  "attachments": [ ←
    {
      "responseElementType": "Attachment", ←
      "assetId": "",
      "name": "VerticalImage",
      "assetName": "",
      "value": "/bsgo5s8vk70tqgjclch5qabmnk_VerticalImage.jpg"
    },
    {
      "responseElementType": "Attachment",
      "assetId": "",
      "name": "HorizontalImage",
      "assetName": "",
      "value": "/bsgo5s8vk70tqgjclch5qabmnk_HorizontalImage.jpg"
    }
  ],
  "endDate": 1460847840000, ←
  "seriesName": "",
  "seriesNumberOfEpisodes": "",
  "prName": "NR", ←
  "referenceProgramId": "bsgo5s8vk70tqgjclch5qabmnk",
  "flags": 0,
  "description": "Childhood best friends Maggie Caruso and Emma Crawford have shared countless adventures growing up together. Now, Maggie and Emma are in store for one of their biggest adventures - raising a baby.", ←
  "seriesSeason": "",
  "episodeId": "",
  "responseElementType": "Program",
  "price": 0,
  "genres": [
    {
      "responseElementType": "Genre",
      "parentName": "",
      "name": "Comedy", ←
      "externalId": "Comedy",
      "id": 3162
    }
  ],
  "name": "Playing House", ←
  "id": 14402,
  "shortName": "",
  "startDate": 1460845980000 ←
},
```

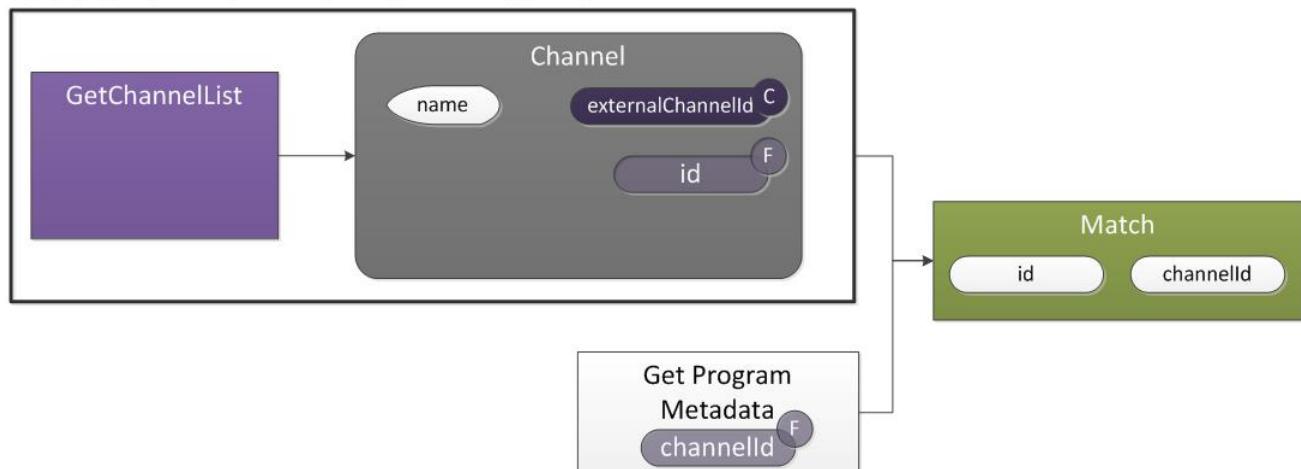
71 Get Channel Info from Internal ID

71.1 Overview

The *GetExtendedProgram* request returns the internal channel ID of the channel on which the program can be viewed. This tutorial explains how to obtain the channel name for display, and the external channel ID for use in other requests.

71.2 Requests and Responses

Get Channel Info from Internal ID



Request: [GetChannelList](#) (from RiGHTv Front-end API)

Gets detailed information on a specified channel. For getting channel information based on internal ID, no parameters need be specified.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Channel](#) (from RiGHTv Front-end API)

For each channel, the name, external ID and internal ID are listed. To find the name and external ID of a channel for which the internal ID is known, match the known internal ID (*channelID* returned by *GetExtendedProgram*) to the *id* property returned by *GetChannelList*.

For the Program screen, significant properties include:

Property	Description
<i>name</i>	The name of the channel.
<i>externalChannelId</i>	The external ID of the channel.
<i>id</i>	The internal ID of the channel.

For more properties, see the RiGHTv Front-end API Developers Guide.

71.3 Related Tutorials

To	Refer to Tutorial
Create a program's metadata	70 Get Program Metadata (p.205)

71.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetProgramLists?channel_external_ids=France24En&start_date=1459403282082&end_date=1483167180000&scene=tvaas&client=json`

This request gets a list of all programs lined up for channel France24En, between 31/03/2016 and 31/12/2016.

A sample response is shown below.

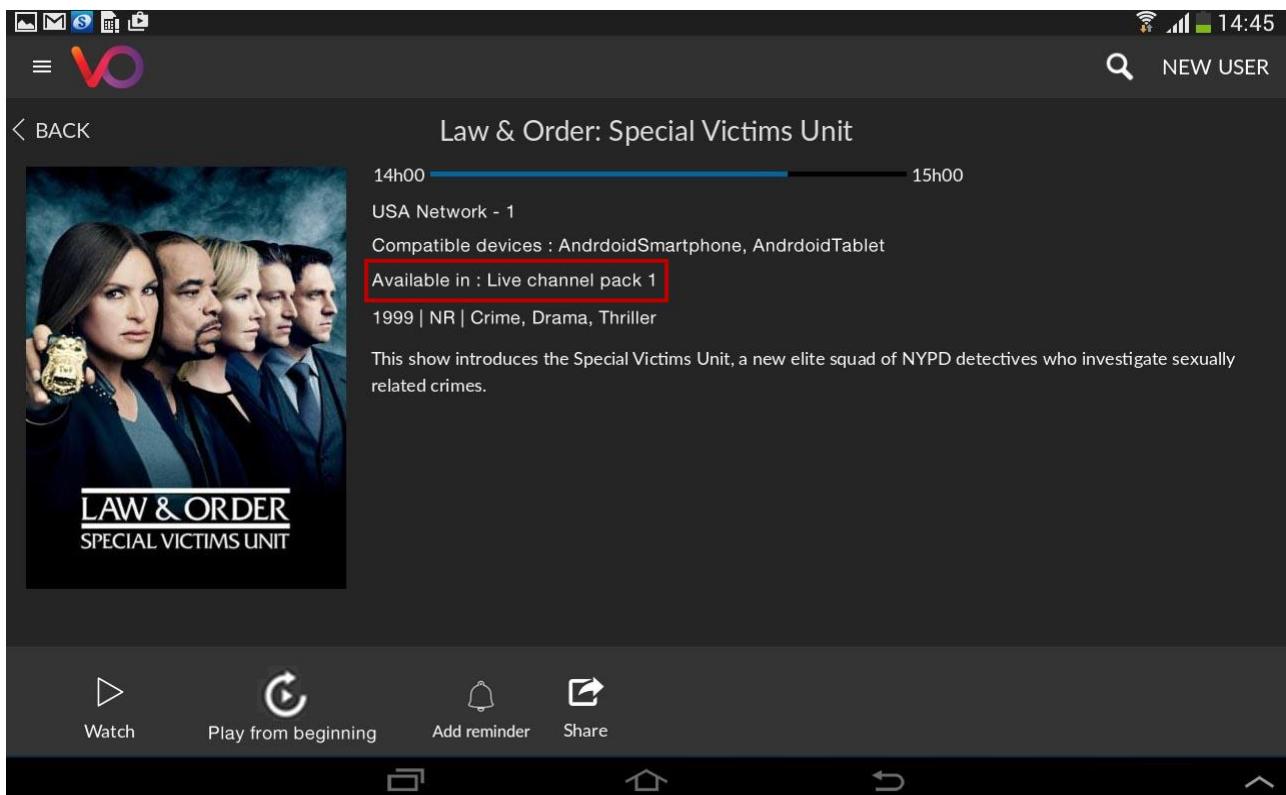
```

"response": [
  {
    "responseElementType": "ProgramList",
    "channelExternalId": "France24En",
    "programs": [
      {
        "advisories": "",
        "prLevel": 0,
        "attachments": [],
        "endDate": 1469739600000,
        "seriesName": "",
        "seriesNumberOfEpisodes": "",
        "prName": "",
        "referenceProgramId": "",
        "flags": 0,
        "description": "",
        "seriesSeason": "",
        "episodeId": "",
        "responseElementType": "Program",
        "price": 0,
        "genres": [],
        "name": "",
        "id": -1,
        "shortName": "",
        "startDate": 0,
        "channelId": 0
      },
      {
        "advisories": "",
        "prLevel": 0,
        "attachments": [
          {
            "responseElementType": "Attachment",
            "assetId": "",
            "name": "VerticalImage",
            "assetName": "",
            "value": "/6m3i5at4317ksf5tbbdubuq4e2_VerticalImage.jpg"
          },
          {
            "responseElementType": "Attachment",
            "assetId": "",
            "name": "HorizontalImage",
            "assetName": "",
            "value": "/6m3i5at4317ksf5tbbdubuq4e2_HorizontalImage.jpg"
          }
        ]
      }
    ],
    "endDate": 1469744100000,
    "seriesName": "",
    "seriesNumberOfEpisodes": "",
    "prName": "NR",
    "referenceProgramId": "6m3i5at4317ksf5tbbdubuq4e2",
    "flags": 0,
    "description": "An elite team of police forensic evidence",
    "seriesSeason": "",
    "episodeId": "",
    "responseElementType": "Program",
    "price": 0,
    "genres": [
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Crime",
        "externalId": "Crime",
        "id": 4041
      },
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Drama",
        "externalId": "Drama",
        "id": 5057
      },
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Mystery",
        "externalId": "Mystery",
        "id": 5073
      },
      {
        "responseElementType": "Genre",
        "parentName": "",
        "name": "Thriller",
        "externalId": "Thriller",
        "id": 12255
      }
    ],
    "name": "CSI: Crime Scene Investigation",
    "id": 14252,
    "shortName": "",
    "startDate": 1469739600000,
    "channelId": 8000
  }
]

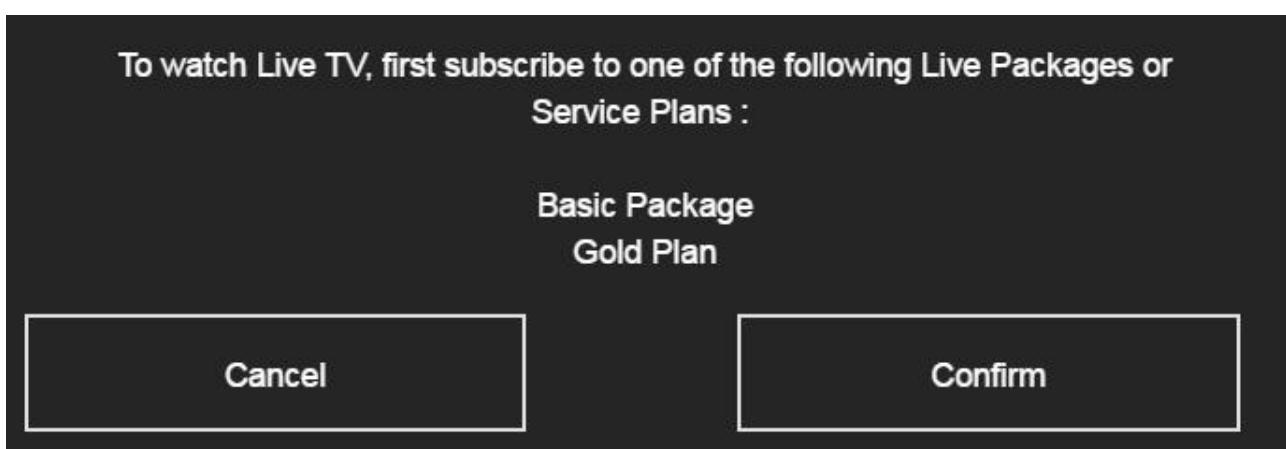
```

72 List Offers for a Channel

A Program screen should list all Channel Package offers through which the channel on which the program is broadcast can be viewed. The screen should also list all Service Plans that include those Channel Packages.



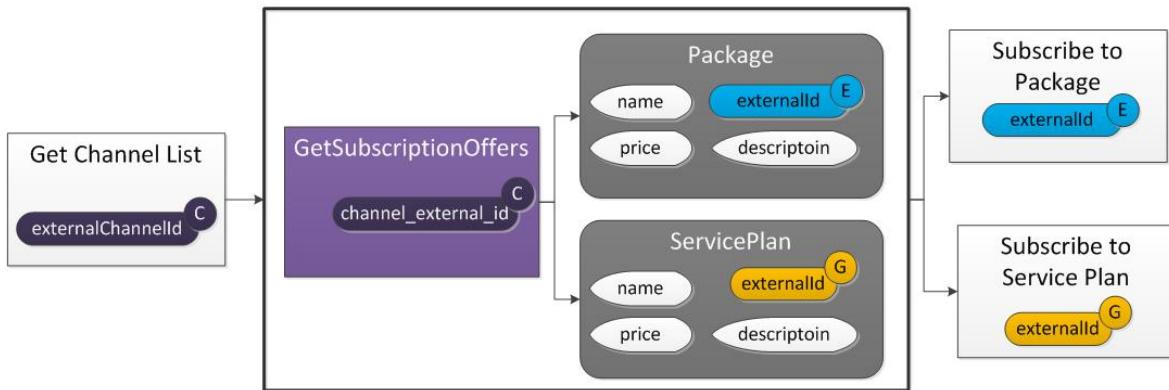
Similarly, if the customer attempts to play a channel to which the account is not yet subscribed from the Now Playing screen, a list of Channel Packages and Service Plans through which the channel can be purchased should be displayed.



On clicking Confirm, the My Offers/All Offers screen should be displayed (see [158 My Offers / All Offers Screen \(p.429\)](#)).

72.1 Requests and Responses

Get Offers for Channel



Request: [GetSubscriptionOffers](#) (from RiGHTv Front-end API)

This request returns a list of all Channel Packages that include a specific channel, as well as Service Plans that include the returned Channel Packages.

Significant parameters include:

Parameter	Description
<i>channel_external_id</i>	The channel external ID.
<i>store_id</i>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end Integration API Guide.

Response: [OffersInfo \(from RiGHTv Front-end API\)](#)

The response lists all Channel Packages in which the specified channel is included, as well as Service Plans that include those Channel Packages. Significant properties include:

Property	Description												
<i>channelPackageList</i>	<p>Lists all Channel Packages that include the channel, each in a <i>Package</i> element. Significant parameters include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The Channel Package external ID.</td></tr> <tr> <td><i>name</i></td><td>The Channel Package name.</td></tr> <tr> <td><i>description</i></td><td>Description of the Channel Package.</td></tr> <tr> <td><i>price</i></td><td>Price of the Channel Package.</td></tr> <tr> <td><i>storeProductReferenceId</i></td><td>The ID of the app store product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The Channel Package external ID.	<i>name</i>	The Channel Package name.	<i>description</i>	Description of the Channel Package.	<i>price</i>	Price of the Channel Package.	<i>storeProductReferenceId</i>	The ID of the app store product reference.
Property	Description												
<i>externalId</i>	The Channel Package external ID.												
<i>name</i>	The Channel Package name.												
<i>description</i>	Description of the Channel Package.												
<i>price</i>	Price of the Channel Package.												
<i>storeProductReferenceId</i>	The ID of the app store product reference.												
<i>servicePlanList</i>	<p>Lists all Service Plans that include the Channel Packages returned, each in a <i>ServicePlan</i> element. Significant parameters include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The Service Plan external ID.</td></tr> <tr> <td><i>name</i></td><td>The Service Plan name.</td></tr> <tr> <td><i>description</i></td><td>Description of the Service Plan.</td></tr> <tr> <td><i>price</i></td><td>Price of the Service Plan.</td></tr> <tr> <td><i>storeProductReferenceId</i></td><td>The ID of the app store product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The Service Plan external ID.	<i>name</i>	The Service Plan name.	<i>description</i>	Description of the Service Plan.	<i>price</i>	Price of the Service Plan.	<i>storeProductReferenceId</i>	The ID of the app store product reference.
Property	Description												
<i>externalId</i>	The Service Plan external ID.												
<i>name</i>	The Service Plan name.												
<i>description</i>	Description of the Service Plan.												
<i>price</i>	Price of the Service Plan.												
<i>storeProductReferenceId</i>	The ID of the app store product reference.												

For more parameters, see the RiGHTv Front-end Integration API Guide.



72.2 Related Tutorials

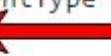
To	Refer to Tutorial
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

72.3 Example

https://<operator>.tvaas.com/RTEFacade/GetSubscriptionOffers?channel_external_id=Tvaas-S-France0&client=json&language_code=en

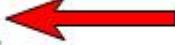
This request calls for a list of offers through which the channel Tvaas-S-France0 is available.

The channelPackageList response is shown below. The channel is available through the channel package called Basic Package, which costs 5 USD.(The channel package itself can be purchased as part of the Gold Plan service plan.)

```
"channelPackageList": [
  {
    "responseElementType": "Package",
    "price": 5, 
    "servicePlanReferenceList": [
      {
        "responseElementType": "ServicePlanReference",
        "name": "Gold Plan", 
        "externalId": "Gold_Plan001"
      },
      ],
      "name": "Basic Package",
      "extrafields": [],
      "description": "Basic Package",
      "externalId": "Trial_Pack",
      "id": 8005
    },
  ]
```

The servicePlanList response is shown below. The channel is available through the Gold Plan service plan, which costs 15 USD.



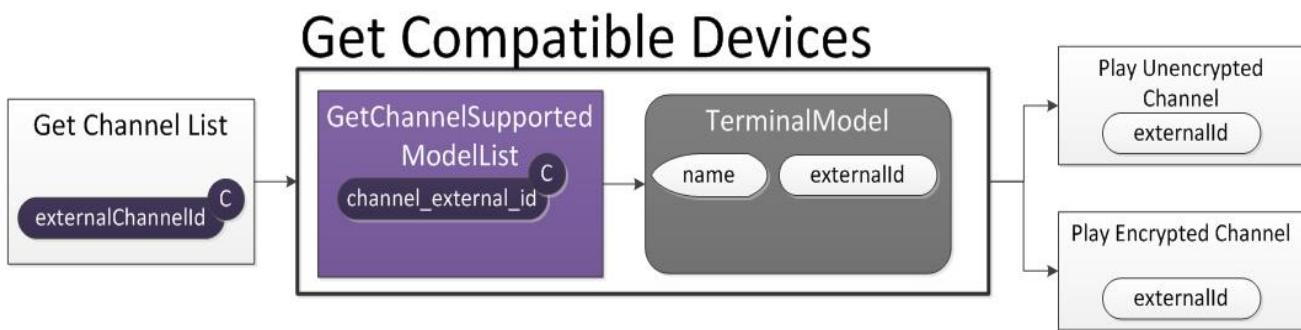
```
"servicePlanList": [  
    {  
        "attachments": [],  
        "excludedFromSelfService": false,  
        "servicePlanOption": "FIXED_BY_DURATION_END_DAY",  
        "bouquets": [],  
        "externalId": "Gold_Plan001",  
        "description": "oscars & basic pacakge",  
        "fixedEndDate": "",  
        "services": [ ],  
        "responseElementType": "ServicePlan",  
        "removalDate": 1493556000000,  
        "activationOption": "IMMEDIATE",  
        "price": 15,   
        "consumptionDurationInDays": 10,  
        "channelPackages": [ ],  
        "name": "Gold Plan",   
        "activationDate": "",  
        "extraFields": []  
    },
```

73 Get Compatible Devices

73.1 Overview

The program screen should display a list of device types on which the channel may be viewed.

73.2 Requests and Responses



Request: [GetChannelSupportedModelList \(from RiGHTv Front-end API\)](#)

Gets a list of devices on which the channel may be viewed. Significant parameters include:

Parameter	Description
<i>channel_external_id</i>	The ID of the channel.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [TerminalModel \(from RiGHTv Front-end API\)](#)

Lists all device (terminal) types on which the channel may be viewed. Significant properties include:

Property	Description
<i>name</i>	The name of the device. This is the string that should be displayed to the user.
<i>externalId</i>	The external ID of the device

For more properties, see the RiGHTv Front-end API Developers Guide.

73.3 Related Tutorials

To	Refer to Tutorial
Create a Program screen	69 Create a Program Screen (p.202)

73.4 Example

https://<operator>.tvaas.com/RTEFacade/GetChannelSupportedModelList?channel_external_id=France24En&client=json

This request gets a list of all devices on which channel France24En can be viewed.

The response extract below indicates that the channel can be viewed on an Android smartphone.

```
"response": [
  {
    "responseElementType": "TerminalModel",
    "vodCasNames": [
      "Viaccess-Purple-DRM"
    ],
    "liveVsName": "RiGHTvLive",
    "vodVsName": "amazon-s3",
    "categoryExternalId": "Smartphone",
    "highestDefinitionForVideo": "",
    "name": "AndroidSmartphone", ←
    "externalId": "AndroidSmartphone", ←
    "vodCasName": "Viaccess-Purple-DRM",
    "liveCasName": "Viaccess-Purple-DRM",
    "vodVsNames": [
      "amazon-s3"
    ]
  }
],
```



74 Setting Program Reminders

74.1 Overview

Users should be able to set a reminder for a program to notify them a specified time before the program is set to start playing.

Reminders must be managed entirely by the client application, including:

- Storing reminders
- Ringing an alarm for a scheduled reminder
- Cancelling reminders

The back-end does not provide specific support for reminders.

74.2 Related Tutorials

To	Refer to Tutorial
Create a Program screen	69 Create a Program Screen (p.202)
Show a list of reminders set	157 Show My Reminders (p.428)

75 Get Recommended Live Content

75.1 Overview

The Program Screen should list recommended programs.

75.2 Requests and Responses

The `GetLiveRecommendationList` call is used to create lists of promoted and recommended content. See [150 Get List of Recommendations \(p.406\)](#) for more information.

75.3 Recommendation Calls

The table below provides best-practice examples of recommendation calls for this screen.

Recommendation List	Type
Related Programs	related_content
<pre>https://<operator>.tvaas.com/compass/GetLiveRecommendationList? blend=related_programs&params=program_id:%7bprogram_id%7d&max_results=%7bmax_ results%7d& max_pr_level=%7bmax_pr_level%7d& device_models=%7bdevice_model%7d&client=json</pre>	

75.4 Related Tutorials

To	Refer to Tutorial
Create a Program screen	69 Create a Program Screen (p.202)
Learn more about recommendations	150 Get List of Recommendations (p.406)

76 Playing Live Programs

76.1 Overview

Once the user has elected to watch a live program, the application must retrieve the playback URL.

76.2 Related Tutorials

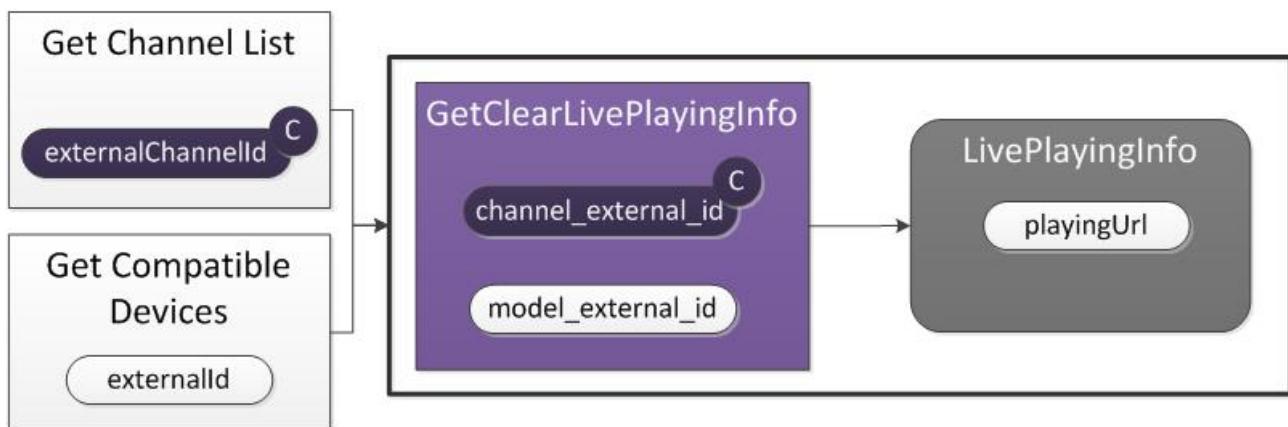
To	Refer to Tutorial
Get the URL for an unencrypted channel	77 Play Unencrypted Channel (p.221)
Get the URL for an encrypted channel	78 Playing Encrypted Channels (p.223)
Use the media player	50 Using the Media Player (p.154)

77 Play Unencrypted Channel

77.1 Overview

An unencrypted channel may be offered for free, or to subscribers only. Because unencrypted channels may be offered to anonymous users, a special request that does not require user log-in must be used.

77.2 Requests and Responses



Request: [GetClearLivePlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for an unencrypted channel. This request does not require account log-in.

Parameter	Description
channel_external_id	The ID of the channel to play. (This corresponds to the <code>externalChannelID</code> parameter in the <code>Channel</code> entity returned by the <code>GetChannelList</code> action.)
model_external_id	The ID of the model of device being used. (This corresponds to the <code>external_id</code> parameter returned in response to the <code>GetTerminalModelList</code> action.)

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [LivePlayingInfo](#) (from RiGHTv Front-end API)

Provides the playback URL for the requested channel.

Property	Description
playingUrl	The playback URL.

For more properties, see the RiGHTv Front-end API Developers Guide.

77.3 Related Tutorials

To	Refer to Tutorial
Play a live program	76 Playing Live Programs (p.220)

77.4 Example

https://<operator>.tvaas.com/RTEFacade/GetClearLivePlayingInfo?channel_external_id=France24En&model_external_id=IPad&client=json

This request gets the playback URL for watching the channel France24En on an IPad.

The response extract below shows the playback url in the *playingUrl* property.

```
{
  "metadata": {
    "request": "GetClearLivePlayingInfo",
    "timestamp": 1459428045101
  },
  "response": {
    "duration": "",
    "responseElementType": "LivePlayingInfo",
    "casToken": "",
    "parentalRatings": [],
    "resumePosition": "",
    "channelId": 8000,
    "playingUrl": "http://olive.fr.globecast.tv/live/ramdisk /france24_fra/hls_video/index.m3u8", ←
    "casTokenExpiration": "",
    "urlExpiration": 1459514445101
  }
}
```



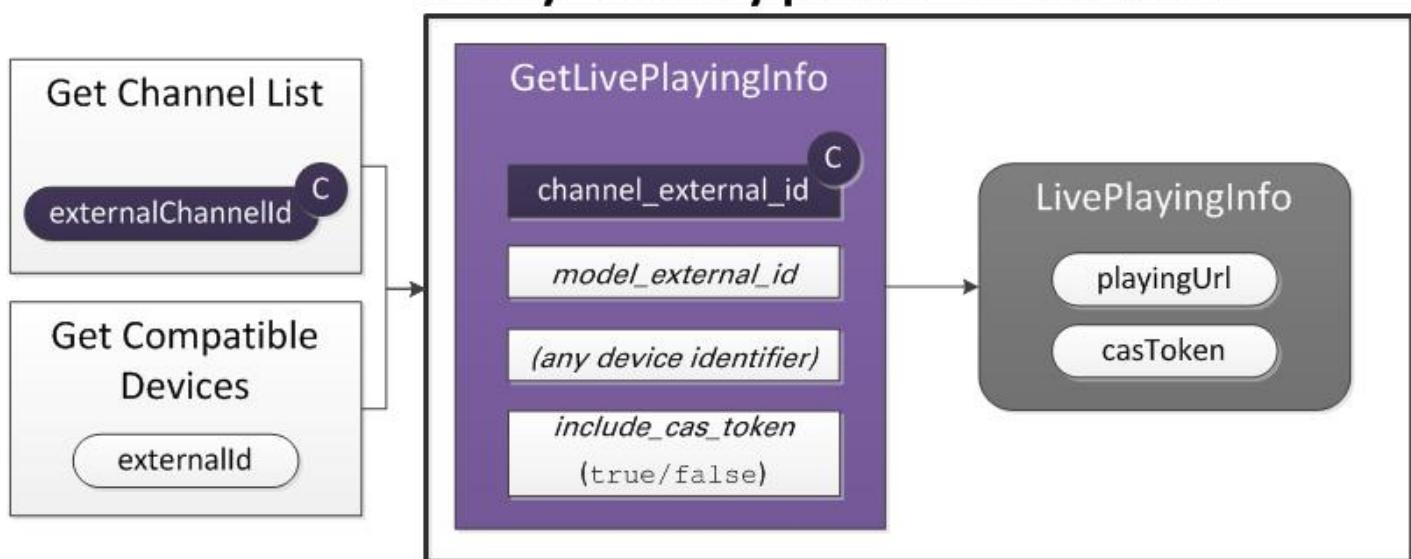
78 Playing Encrypted Channels

78.1 Overview

Any channel that is encrypted or that requires a license to be viewed requires the use of the action *GetLivePlayingInfo*, instead of *GetClearLivePlayingInfo*.

78.2 Requests and Responses

Play Encrypted Channel



Request: [GetLivePlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for an encrypted channel. This request requires that the user be logged in. The following parameters are required:

Parameter	Description
<code>channel_external_id</code>	The ID of the channel to play.
<code>model_external_id</code>	The ID of the model of device being used.

One of: <ul style="list-style-type: none">• <i>mac_address</i>• <i>serial_number</i>• <i>smart_card_id</i>	One the device (terminal) identifier for the requesting device.
include_cas_token	If set to <code>true</code> requests that the CAS token should be calculated and included in the result. The default value is <code>false</code> . A CAS token is required if a license is needed to play the channel.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [LivePlayingInfo](#) (from RiGHTv Front-end API)

Provides the playback URL for the requested channel, and if requested, the CAS token required for viewing licensed content.

Property	Description
playingUrl	The playback URL.
casToken	The CAS token, if requested.

For more properties, see the RiGHTv Front-end API Developers Guide.

78.3 Related Tutorials

To	Refer to Tutorial
Play a live program	76 Playing Live Programs (p.220)

78.4 Example

```
https://<operator>.tvaas.com/RTEFacade/GetLivePlayingInfo?channel_external_id=France24En&include_cas_token=true&serial_number=MOZILLA_D7361EEB-B15C-03A9-0237-190D83CDDA31&client=json
```

This request gets the playback URL for the channel France24En for a specified device in the account, requesting a CAS token for licensed content.

The response extract below shows the playback url in the *playingUrl* property.

```
http:\V\olive.fr.globecast.tv\live\ramdisk\france24_fra\hls_video\index.m3u8
```



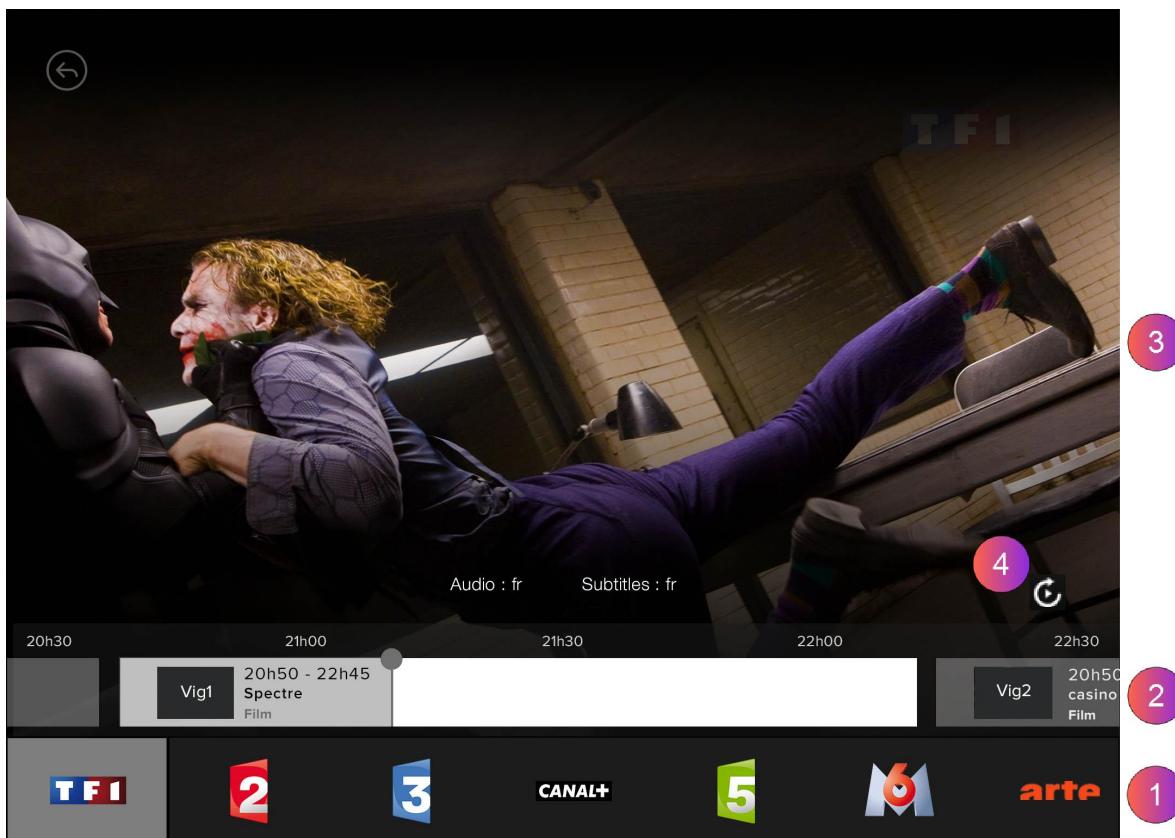
79 Create a Now Playing Screen

79.1 Overview

Whereas the TV Guide displays a grid showing all channels and their program line-up, the Now Playing screen is designed to allow users to flip between channels quickly.

The Now Playing screen should display:

1. A banner of channels, represented by their icons. Each is selectable.
2. The timeline for the currently selected channel.
3. The player, playing the current program on the selected channel if subscribed to.
4. A button to indicate if Start Over is supported, and to launch Start Over playback when pressed, or a button to indicate if Catch-Up is supported, and to launch Catch-Up playback when pressed.



79.2 Implementation

All of the actions required to create a Now Playing screen are referenced below.

79.3 Related Tutorials

To	Refer to Tutorial
Get a list of channels, use <i>GetChannelList</i> to get the <i>Channel</i> entity.	64 Get a List of Channels (p.190)
Get the channel icons, read from the <i>Channel</i> entity.	64 Get a List of Channels (p.190)
Get a list of programs per channel for showing the timeline, use <i>GetProgramLists</i> .	65 Get List of Channel Programs (p.193)
Determine if the account has access to a channel.	66 Identify Subscribed Channels (p.197)
Get the playback URL for the selected channel, use <i>GetClearLivePlayingInfo</i> (for unencrypted channels) or <i>GetLivePlayingInfo</i> (for encrypted channels).	76 Playing Live Programs (p.220)
List Channel Packages and Service Plans through which an account can purchase access to a channel	72 List Offers for a Channel (p.211).
Determine whether Start Over is supported	163 Identify Start Over Enabled Programs (p.454)
Get the playback URL for the Start Over for the selected channel	164 Playing a Live Channel through Start Over (p.458)
Determine whether Catch-Up is supported	173 Identify Catch-Up Enabled Programs (p.483).
Get the playback URL for the Catch-Up for the selected channel	176 Playing a Program through Catch-Up (p.492)

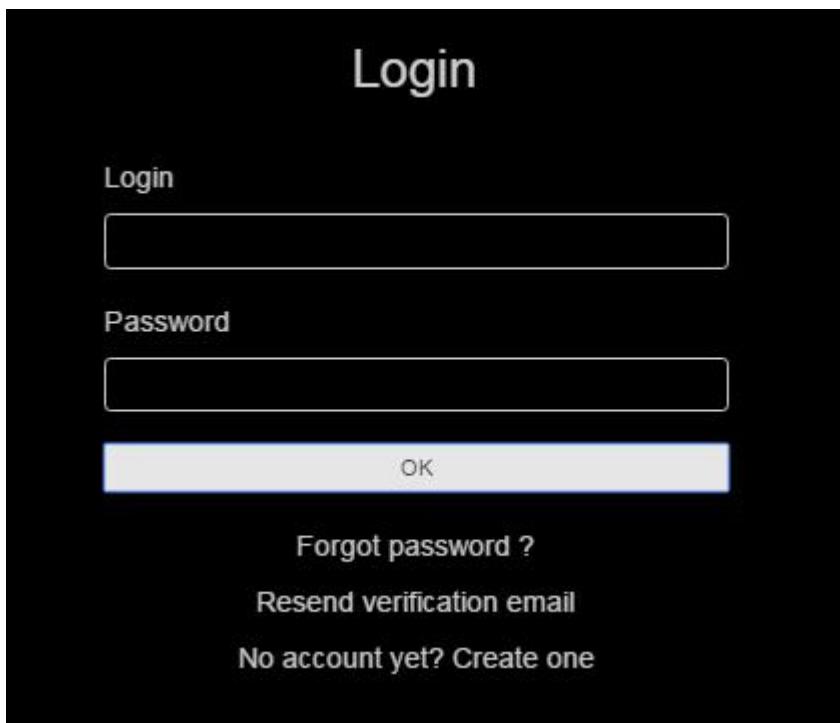


80 Create a Login Screen

80.1 Overview

As explained in [2.2.2 Anonymous vs. Authenticated Users \(p.2\)](#), some app functionality may be made available without having to log in. However, to purchase content and view that content, users must be required to log in.

A typical Login screen is shown below.



The Login screen must allow users to:

- Enter a username and password
- Request a password reset (Forgot password)
- Request a new verification email (in order to verify a new account)
- Create a new account

80.2 Related Tutorials

For implementation instructions, refer to the tutorials referenced below:

To	Refer To Tutorial
Create a new account	81 Create an Account (p.229) . 81 Create an Account (p.229)
Log in an existing user	86 Logging In and Out (p.240) . 86 Logging In and Out (p.240)
Reset an account password	106 Updating and Resetting Account Passwords (p.283) .

81 Create an Account

81.1 Overview

The application should include a Create New Account facility. Users will provide personal information, their e-mail address (which will also function as the user name), and a password of their choice.

Create account

Your name (max 30 characters)

Email (will also be used as your username)

Password (Use 6-20 characters and at least 1 digit)

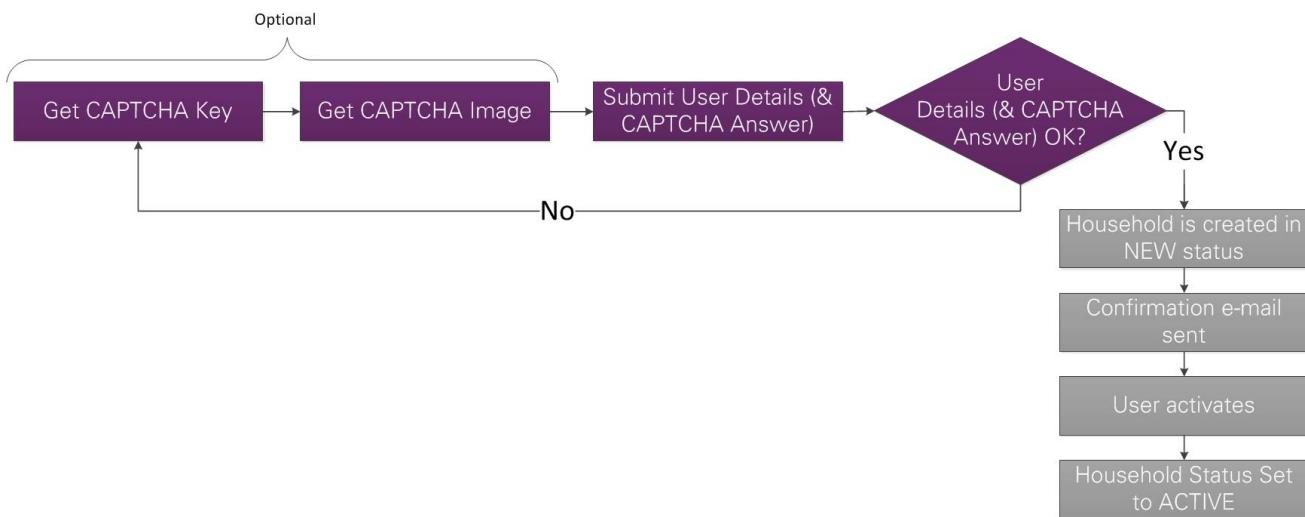
Create

Note

Optionally, the user can be subjected to a Captcha challenge.



The following process should be followed. The gray boxes indicate the rest of the process for information only:



The user can request that the confirmation email be resent. See [82 Resend Verification Email \(p.231\)](#)

81.2 Related Tutorials

To	Refer To Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Get a CAPTCHA key	83 Get CAPTCHA Key (p.233)
Get a CAPTCHA image	84 Get CAPTCHA Image (p.235)
Submit the account creation form	85 Submit User Details and Capture Challenge (p.237)
Resend the account confirmation email	82 Resend Verification Email (p.231)

82 Resend Verification Email

82.1 Overview

A new user is sent a verification email after registering, which includes a link to click to complete the account activation process.

The user should be able to request that the email be resent.

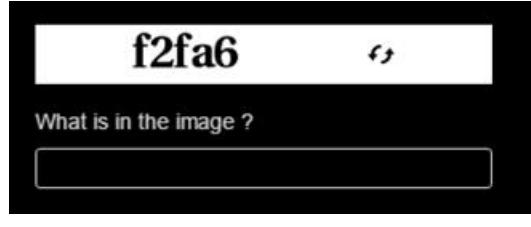
The image shows a dark-themed login interface. At the top is a large "Login" button. Below it are fields for "Login" and "Password". A central "OK" button is positioned between them. At the bottom of the screen are three links: "Forgot password?", "Resend verification email" (with a cursor icon indicating it is clickable), and "No account yet? Create one".

The user should then be requested to enter the email address entered when registering.

The image shows a dark-themed "Resend" screen. It features a text input field labeled "Email (will also be used as your username)". Below the input field is a "Send" button.

Note

Optionally, the user can be subjected to a Captcha challenge.



82.2 Requests and Responses

The ResendConfirmationEmail request is used to request that the verification email be resent.

Request: [ResendConfirmationEmail](#) (from RiGHTv Front-end API)

This request resends the verification email. The following parameters are required:

Parameter	Description
<i>username</i>	The account name (e-mail address)
The following parameters are only relevant when implementing a Captcha challenge:	
<i>captcha_key</i>	The Captcha key obtained using the GetCaptchaKey request.
<i>captcha_challenge</i>	The user's answer to the Captcha challenge.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

82.3 Related Tutorials

To	Refer To Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Get a CAPTCHA key	83 Get CAPTCHA Key (p.233)
Get a CAPTCHA image	84 Get CAPTCHA Image (p.235)
Submit the account creation form	85 Submit User Details and Capture Challenge (p.237)



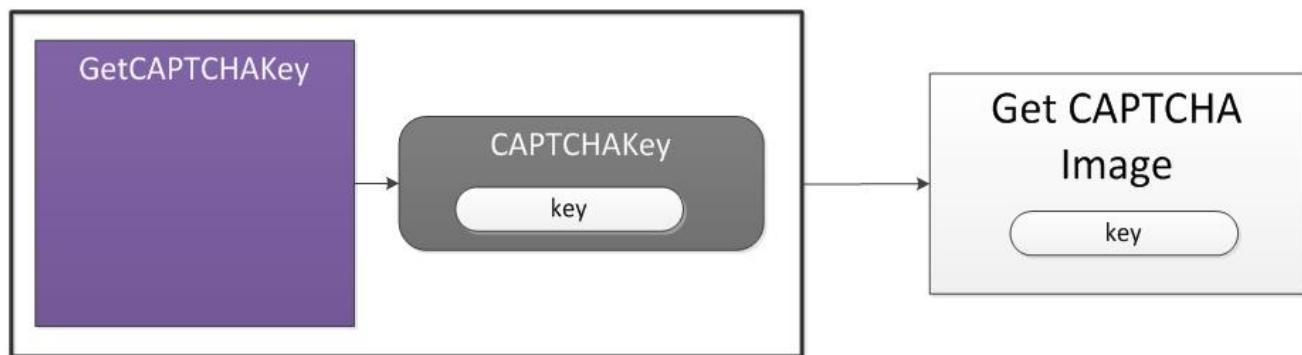
83 Get CAPTCHA Key

83.1 Overview

The first step in account creation is the generation of a CAPTCHA key that will be used to retrieve a CAPTCHA image.

83.2 Requests and Responses

Get CAPTCHA Key



Request: [GetCAPTCHAKey](#) (from RiGHTv Front-end API)

This request obtains a unique CAPTCHA key, which is simply a numeric key. No parameters are required.

Response: [CAPTCHAKey](#) (from RiGHTv Front-end API)

This entity provides the unique CAPTCHA key.

Property	Description
key	The unique CAPTCHA key.

83.3 Related Tutorials

To	Refer To Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Create a new account	81 Create an Account (p.229)
Get a CAPTCHA image	84 Get CAPTCHA Image (p.235)

83.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetCaptchaKey?client=json>

This request calls for a new CAPTCHA key.

The key is returned.

```
{  
  "metadata": {  
    "request": "GetCaptchaKey",  
    "timestamp": 1456037179846  
  },  
  "response": {  
    "responseElementType": "CaptchaKey",  
    "key": "1456037181866"  
  }  
}
```

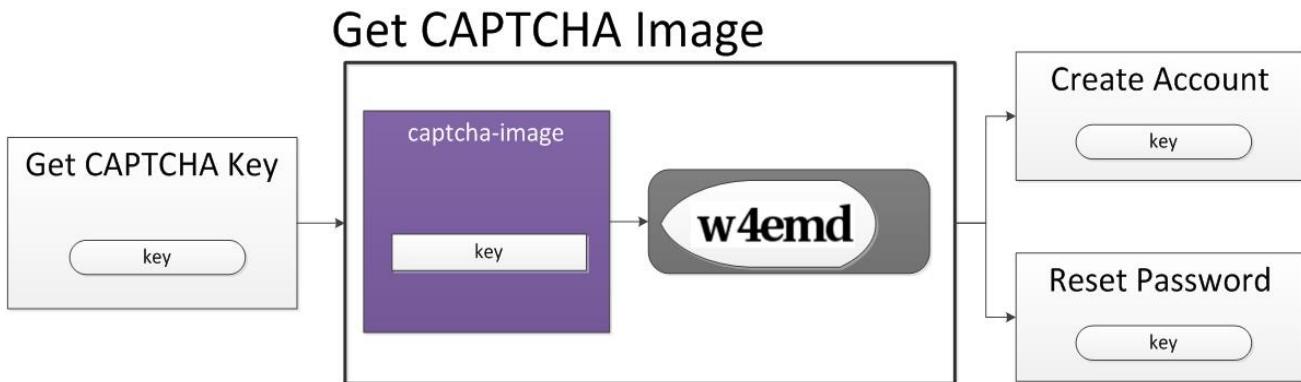


84 Get CAPTCHA Image

84.1 Overview

The CAPTCHA image to be shown on the Create Account screen must be retrieved using the generated CAPTCHA key.

84.2 Requests and Responses



Request: captcha-image (from RiGHTv Front-end API)

This call retrieves CAPTCHA images. Only one parameter need be specified:

Parameter	Description
key	The CAPTCHA key obtained using the GetCAPTCHAKey request.

This CAPTCHA image is returned directly.

84.3 Related Tutorials

To	Refer To Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Create a new account	81 Create an Account (p.229)
Get a CAPTCHA image	84 Get CAPTCHA Image (p.235)

84.4 Example

`https://<operator>.tvaas.com/RTEFacade/captcha-image?key=1456037181866`

This request calls for a new CAPTCHA image that matches the key 1456037181866.



A sample response is shown below.

a6ncc

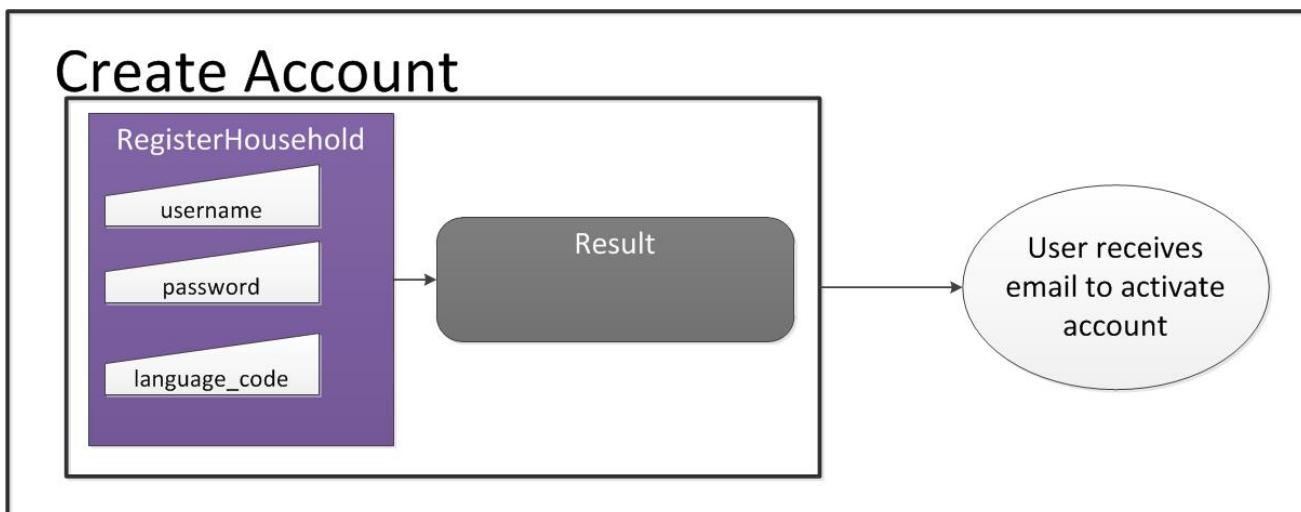
85 Submit User Details and Capture Challenge

85.1 Overview

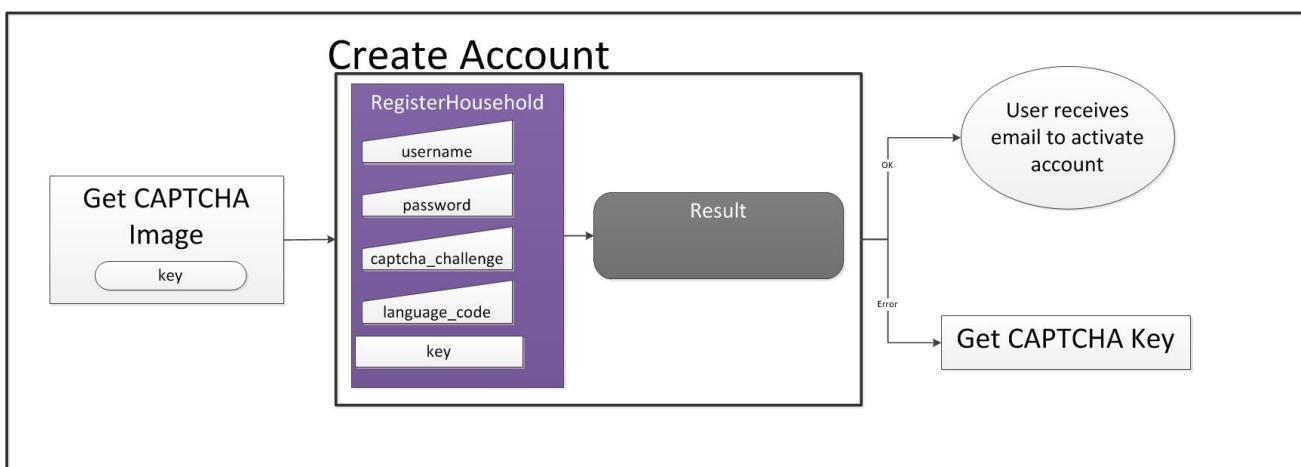
When creating a new account, the user is required to provide various personal details. Optionally, the user can be subjected to a Captcha challenge. The personal details, and the Captcha challenge answer if implemented, are submitted in a single request.

85.2 Requests and Responses

Process Flow without Captcha Challenge



Process Flow with Captcha Challenge



Request: [RegisterHousehold](#) (from RiGHTv Front-end API)

This request sends all the information required to open a new online account. The following parameters must be included:

Parameter	Description
<i>username</i>	A valid e-mail address.
<i>password</i>	A password of the user's choice of 6 to 20 characters, including at least one digit.
<i>language_code</i>	<i>The language of the new account (if not provided, the system default is used).</i>
The following parameters are only relevant when implementing a Captcha challenge:	
<i>captcha_key</i>	The key obtained by the <i>GetCAPTCHAKey</i> Request.
<i>captcha_challenge</i>	The user's answer to the CAPTCHA image.

Additional information such as address, telephone numbers, etc., can be sent too. See the RiGHTv Front-end API guide.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE .
<i>message</i>	A short message from the server

Note

Each CAPTCHA key can only be submitted once. If for any reason the RegisterHousehold action fails, a new CAPTCHA key and image must be obtained.

85.3 Related Tutorials

To	Refer To Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Create a new account	81 Create an Account (p.229)



85.4 Example

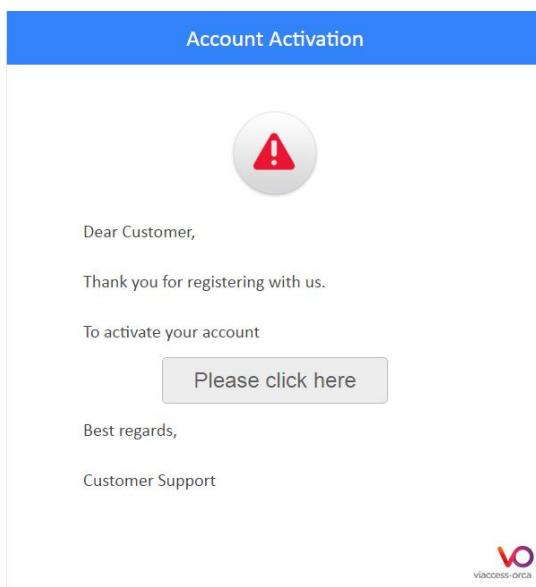
https://<operator>.tvaas.com/RTEFacade/RegisterHousehold?username=reuserg@gmail.com&password=testabc1&captcha_key=1456037181866&captcha_challenge=a6ncc&client=json

This request call registers a new account with email `reuserg@gmail.com` and password `testabc1`. The Catpcha key and the user's answer are both included.

Sample response of a successful RegisterHousehold call is shown below.

```
[{"metadata": {  
    "request": "RegisterHousehold",  
    "timestamp": 1456038915178  
},  
 "response": {  
    "message": "Household Registered Successfully",  
    "status": "SUCCESS"  
}}
```

After registering a new online account, the user receives an e-mail to activate the account.



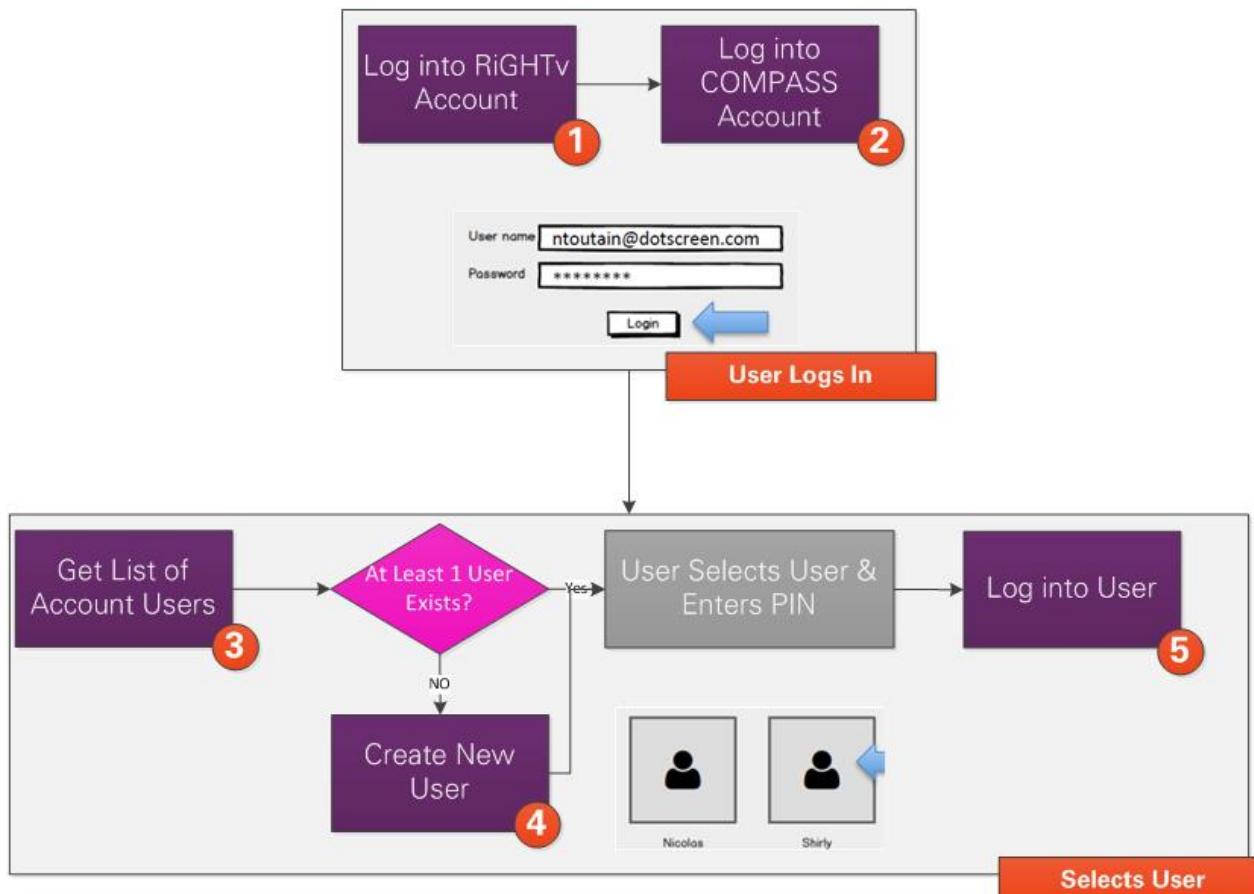
86 Logging In and Out

86.1 Overview

From the user's point of view, logging in is a two-step process: logging into the app, and selecting an individual user (profile). If no user exists, a new user must be created.

However, the application must in fact log the account into both the RiGHTv and COMPASS systems.

The process illustrated below implements those two user actions.



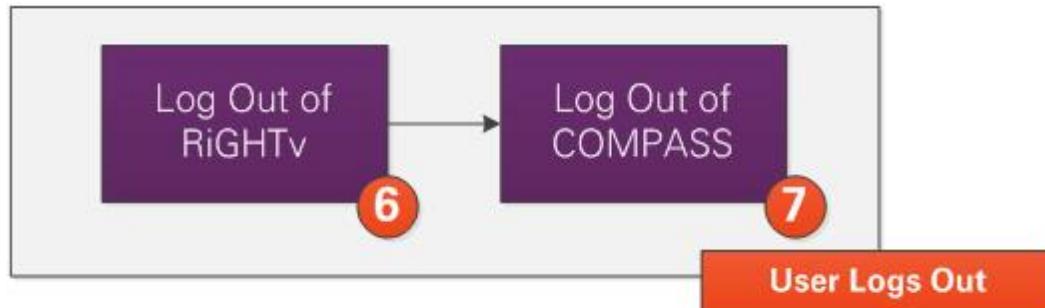
1. Log into RiGHTv account.
2. Log into the COMPASS account.
3. Get a list of users in the account, so as to be able to display the list to the user.
4. If no users exist yet, create one.

5. Log into the selected user.

Once logged in, a customer should be able to choose a different user at any time. The action calls are identical to when selecting a user after logging in. This process is illustrated below.



When a user logs out, the app must log out of the RiGHTv account and out of the COMPASS account, as illustrated below.



6. Log Out of RiGHTv

7. Log Out of COMPASS

86.2 Related Tutorials

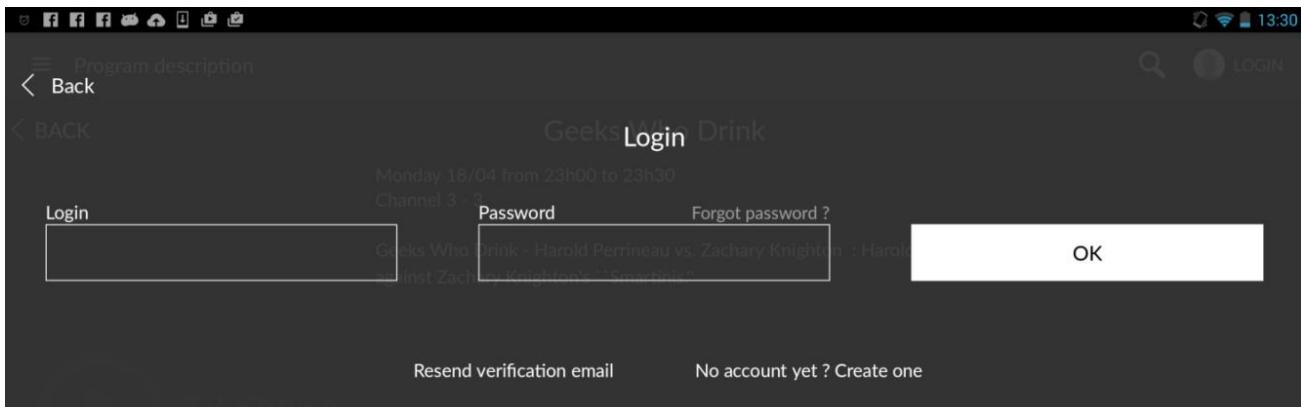
To	Refer to Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Create a new account	81 Create an Account (p.229)
Log the account into RiGHTv	87 Log into RiGHTv Account (p.243)
Log the account into COMPASS	88 Log into COMPASS Account (p.246)
Get a list of individual users within the account	89 Get User List (p.248)

Log in the user	90 Log in the User (p.251)
Log out of RiGHTv	91 Log Out of RiGHTv (p.253)
Log out of COMPASS	92 Log Out of COMPASS (p.255)

87 Log into RiGHTv Account

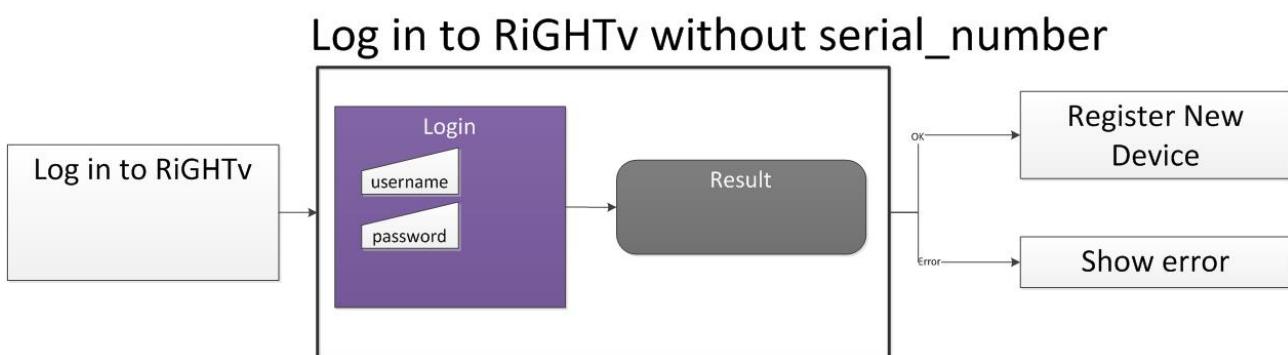
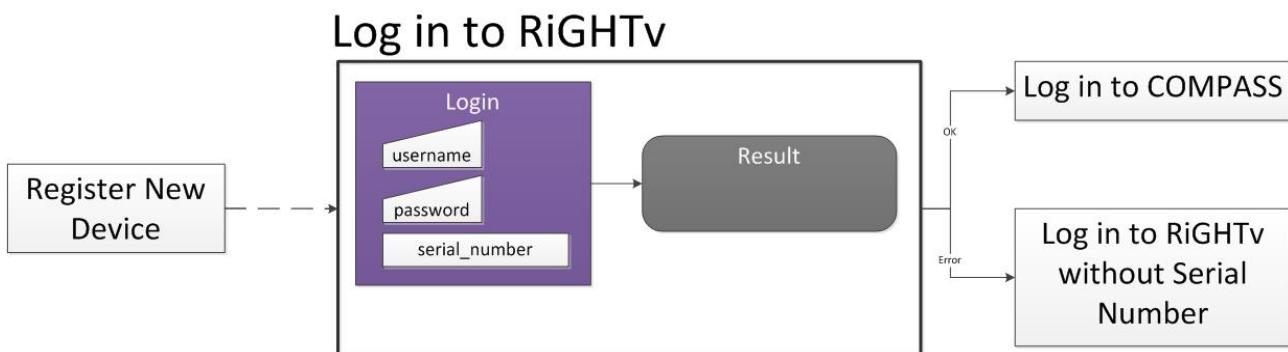
87.1 Overview

Users log into the app (and thus into RiGHTv) by entering their account name and password.



Note on Analytics: Although it is possible to log in without sending device identifying information, that information is required for the TV Business Analytics Platform.

87.2 Requests and Responses



Request: Login (from RiGHTv Front-end API)

To log into the RiGHTv account, the following parameters must be provided:

Parameter	Description
<i>username</i>	The account username, which is the email address supplied when creating the account
<i>password</i>	The account password.
<i>serial_number</i>	The ID of the device being used, as set in the console. This parameter is not absolutely required but it is advisable to include.

Response: Result (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

Note: If login fails, retry without submitting the serial number. If that works, register the new device. See [113 Device Management \(p.298\)](#).

87.3 Related Tutorials

To	Refer to Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Log in from the login screen	86 Logging In and Out (p.240)
Register a device	113 Device Management (p.298)

87.4 Example

`https://<operator>.tvaas.com/RTEFacade/Login?username=1&password=1&serial_number=4BFA9FAE-2753-4938-B342-D30394D7C773&client=json`

This request logs into the RiGHTv account for *username* 1, with *password* 1 and device *serial_number* 4BFA9FAE-2753-4938-B342-D30394D7C773.



The returned response for a successful log-in is shown below.

```
{  
  "metadata": {  
    "request": "Login",  
    "timestamp": 1456319364215  
  },  
  "response": {  
    "message": "'Login' action finished successfully"  
  },  
  "status": "SUCCESS"  
}  
}
```



The response shown below indicates a problem with the username/password combination.

```
{  
  "metadata": {  
    "request": "Login",  
    "timestamp": 1456319530731  
  },  
  "response": {  
    "code": "INVALID_CREDENTIALS",  
    "message": "Login failed.",  
    "status": "FAILURE"  
  }  
}
```

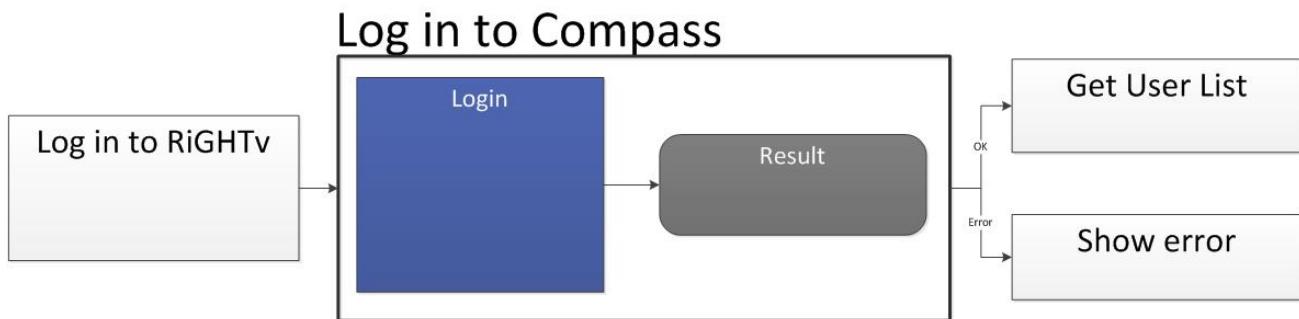


88 Log into COMPASS Account

88.1 Overview

After having logged into the user's RiGHTv account, the application must log into the corresponding COMPASS account. The account must first be logged into RiGHTv (see [87 Log into RiGHTv Account \(p.243\)](#)).

88.2 Requests and Responses



Request: [LogIn](#) (from COMPASS RT API)

This request logs the account (household) into COMPASS. No parameters are required.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .
<i>message</i>	A short message from the server

88.3 Related Tutorials

To	Refer to Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Log in from the login screen	86 Logging In and Out (p.240)

Register a device	113 Device Management (p.298)
Get a list of individual users within the account	89 Get User List (p.248)

88.4 Example

<https://<operator>.tvaas.com/compass/Login?client=json>

This request logs into the COMPASS account.

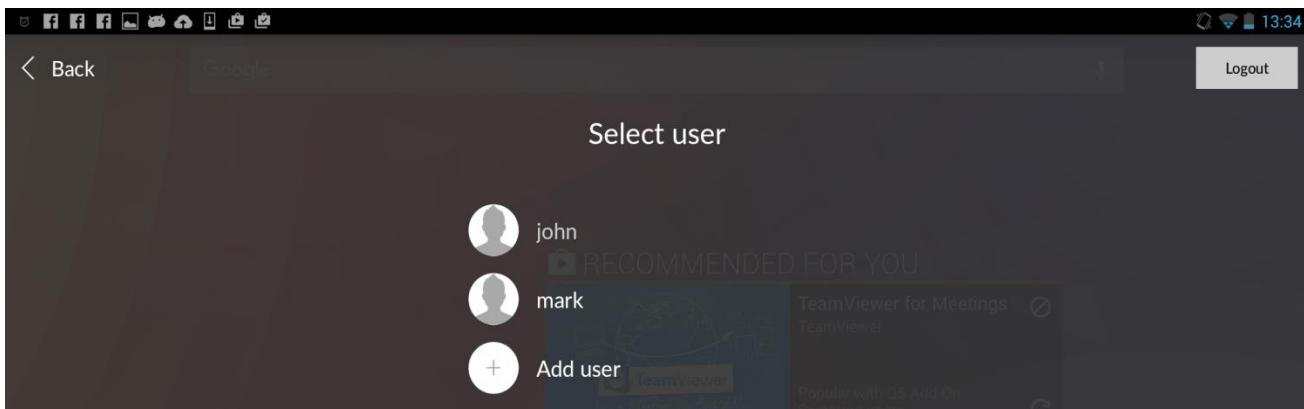
The response below indicates successful log-in to COMPASS.

```
{  
  "metadata": {  
    "request": "Login",  
    "timestamp": 1456319764663  
  },  
  "response": {  
    "message": "identity  
    =nNi98DTDPgS8kgvxCd76094I%2BeJwZBsGMAH1R6W8%2B  
    QeABSWV6qa4ayxTo%2B0QdAntrZvbtpB%2B7gx8eLZo5FmW  
    g%3D%3D",  
    "properties": {},  
    "status": "SUCCESS" ←  
  }  
}
```

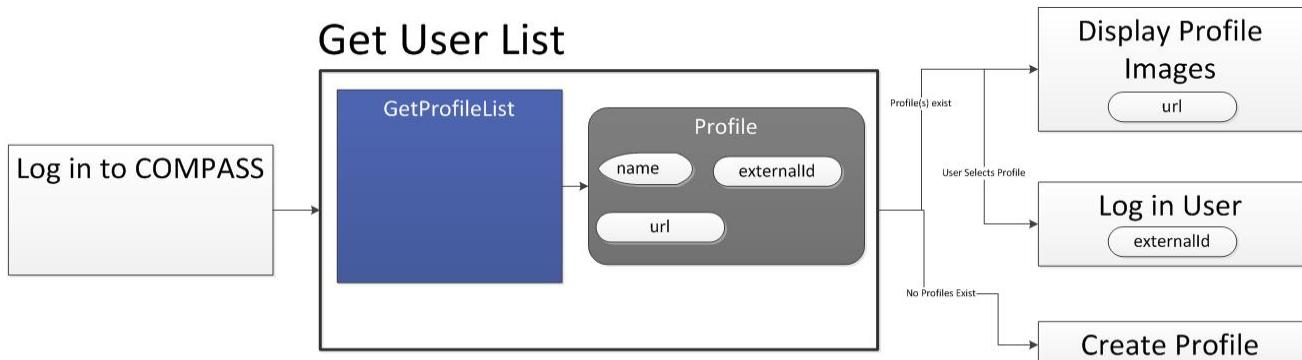
89 Get User List

89.1 Overview

After having logged in to the account, the application should show a list of existing users from which to choose. The app must thus retrieve the list of existing users. Typically the name and image of each user is shown. If no users are found, the application must prompt the user to create one.



89.2 Requests and Responses



Request: [GetProfileList](#) (from COMPASS RT API)

This request gets the list of existing users in the account. No parameters are required.

Response: **Profile** (from COMPASS RT API)

Provides information about each user in the account . For displaying the list of users, the most relevant properties are:

Property	Description
<i>name</i>	The name of the user to be displayed in the user list
<i>URL</i>	The path to the uploaded image for the user.
<i>externalid</i>	The ID number of the user. This is needed for the next step when logging into the user.
<i>image</i>	The name of the user's image file. (See 99 Get a User's Image (p.270) to learn how to use this property.)

89.3 Related Tutorials

To	Refer to Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Log in from the login screen	86 Logging In and Out (p.240)
Log account into COMPASS	88 Log into COMPASS Account (p.246)
Display the users' images	99 Get a User's Image (p.270)
Create and update users in the account	93 Creating and Updating Users (p.257)
Log the user into COMPASS	90 Log in the User (p.251)



89.4 Example

<https://<operator>.tvaas.com/compass/GetProfileList?client=json>

This request gets the list of users in the account.

The response for a single user is shown below.

```
"name": "Colin Coltrain",
"externalId": "10702",
"pinCodeExist": 0,
"id": 10702,
"profileProperties": [],
"suspended": 0
},
{
  "image": "",
  "images": [
    {
      "name": "ProfileImage",
      "description": "",
      "id": 154,
      "type": "Internal",
      "url": "/GetProfileImage?image_id=154" ←
    }
  ]
}
```

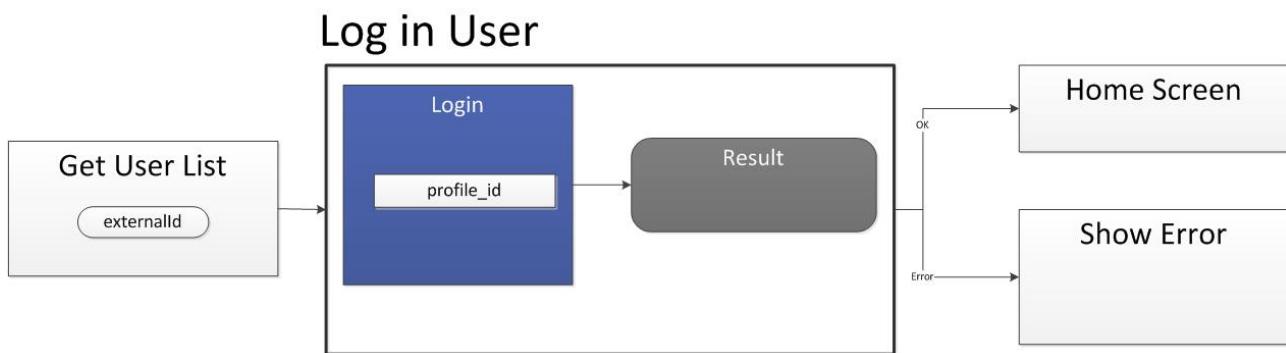


90 Log in the User

90.1 Overview

After the customer selects their user from the list shown, the application should log that particular user in.

90.2 Requests and Responses



Request: [Login \(from COMPASS RT API\)](#)

This request logs in a particular user. The following parameter must be provided:

Parameter	Description
<i>profile_ID</i>	The ID of the user, obtained from the <i>externalId</i> property returned by the <i>GetProfileList</i> action (see 89 Get User List (p.248)).

Response: [Result \(from COMPASS RT API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .
<i>message</i>	A short message from the server

90.3 Related Tutorials

To	Refer to Tutorial
Create a Login screen	80 Create a Login Screen (p.227)
Log in from the login screen	86 Logging In and Out (p.240)
Log account into COMPASS	88 Log into COMPASS Account (p.246)
Get a list of users	89 Get User List (p.248)

90.4 Example

https://<operator>.tvaas.com/compass/Login?profile_id=10702&client=json

This request logs into profile 10702.

The response below indicates successful log-in to the selected profile.

```
{
  "metadata": {
    "request": "Login",
    "timestamp": 1456321584102
  },
  "response": {
    "message": "identity
=gbw39DuaTjVKsEvJeYe6ushZozbgbV%2FBN
yedkaFTeBknlicvcmg6GvT3DXzFrgnCTEeyS
8rOCeqiNfxRG7a9Eg%3D%3D",
    "properties": {},
    "status": "SUCCESS"
  }
}
```

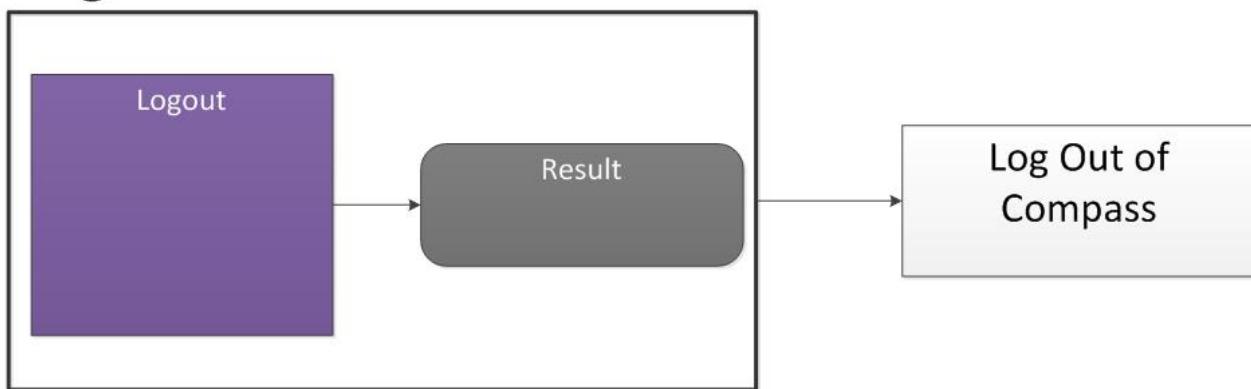
91 Log Out of RiGHTv

91.1 Overview

When the user chooses to log out the application, the application must log out of the RiGHTv account.

91.2 Requests and Responses

Log Out of RiGHTv



Request: [Logout](#) (from RiGHTv Front-end API)

This request logs out of the RiGHTv account. No parameters are required.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

91.3 Related Tutorials

To	Refer to Tutorial
Learn about logging in and logging out	86 Logging In and Out (p.240)

91.4 Example

<https://<operator>.tvaas.com/RTEFacade/Logout?client=json>

This request logs out of RiGHTv.

The response below indicates successful log-out of RiGHTv.

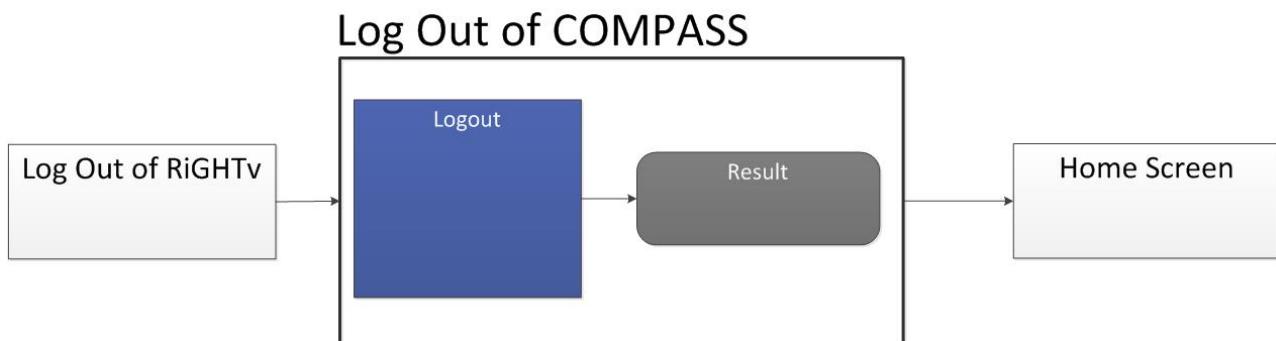
```
[{"metadata": {  
    "request": "Logout",  
    "timestamp": 1460444259375  
},  
 "response": {  
    "message": "'Logout' action finished successfully",  
    "status": "SUCCESS"  
}}
```

92 Log Out of COMPASS

92.1 Overview

When the user chooses to log out the app, the app must log out of the COMPASS account.

92.2 Requests and Responses



Request: [Logout](#) (from COMPASS RT API)

This request logs out of the COMPASS account. No parameters are required.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

92.3 Related Tutorials

To	Refer to Tutorial
Learn about logging in and logging out	86 Logging In and Out (p.240)

92.4 Example

<https://<operator>.tvaas.com/compass/Logout?client=json>

This request logs out of COMPASS.

The response below indicates successful log-out of COMPASS.

```
[{"metadata": {  
    "request": "Logout",  
    "timestamp": 1460444203867  
},  
"response": {  
    "message": "'Logout' action finished successfully",  
    "properties": {},  
    "status": "SUCCESS"  
}  
}
```

93 Creating and Updating Users

93.1 Overview

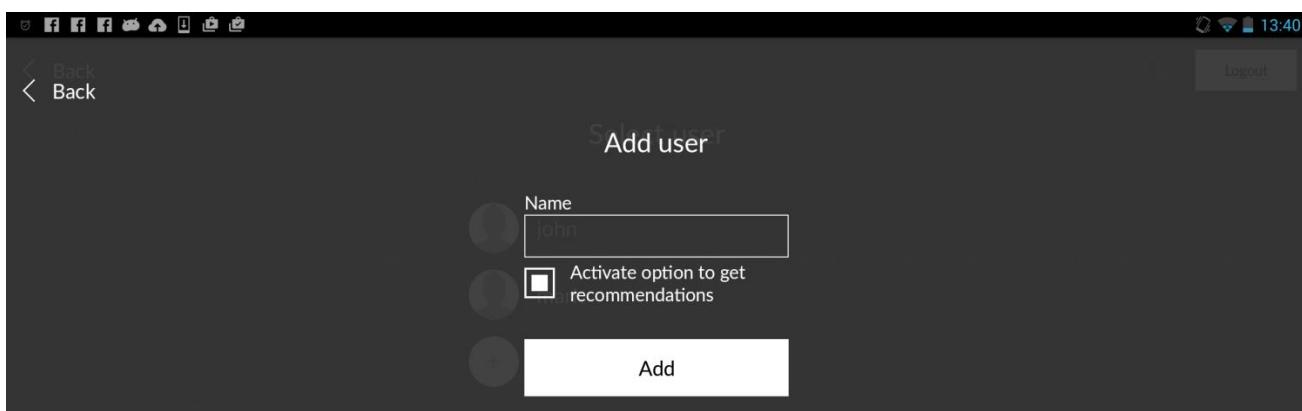
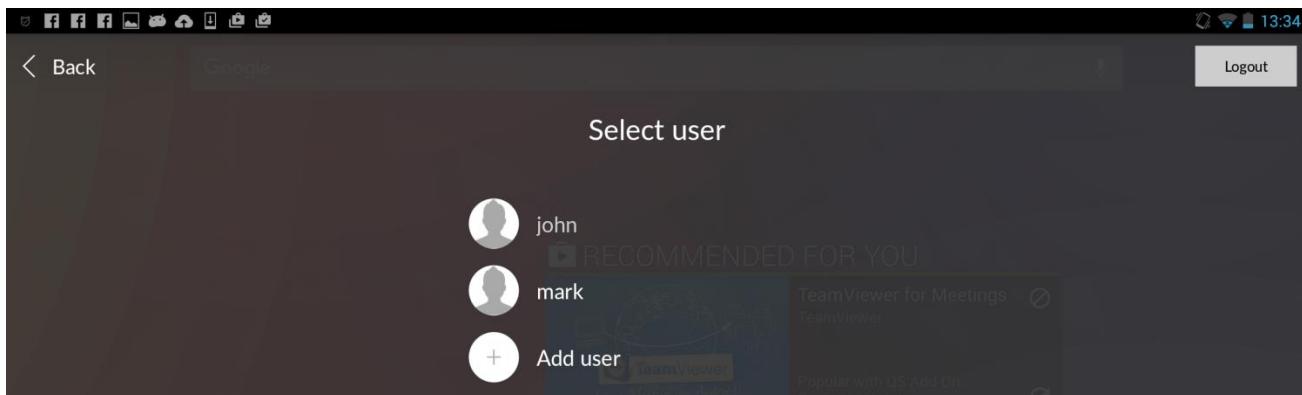
Once an account (household) has been created, the application must allow a logged-in user to define multiple users, representing individuals within the account.

After logging in, the app will show a list of existing users, and give the option to add more. After first log on, no users will exist yet and the application must prompt the user to create one.

Note

The user must be logged in before a new profile can be created.

This tutorial provides instructions for creating and updating users.



93.2 Related Tutorials

To	Refer To Tutorial
Create a new user in the account	94 Create a New User (p.259)
Change the name of a user	95 Update a User's Name (p.262)
Upload and replace a user's image	98 Managing User Images (p.269)
Register the user's selection of Usage Data opt-in/out status	103 Managing User Usage Data Opt-In/out Status (p.278)
Update the user's purchase pin	109 Purchase PIN Management (p.291)

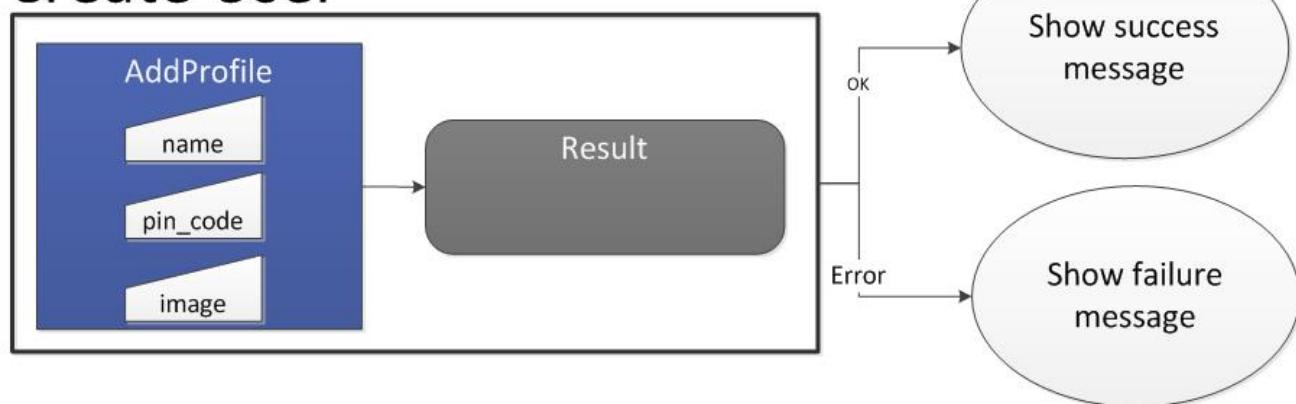
94 Create a New User

94.1 Overview

The application must provide the ability to create new users within the logged-in account, after having logged in.

94.2 Requests and Responses

Create User



Request: [Addprofile \(from COMPASS RT API\)](#)

This request creates a new user, linked to the logged-in account. The minimal required parameters are listed below.

Parameter	Description
<i>name</i>	The name of the user.
<i>image</i>	The <i>name</i> and <i>value</i> (path and file name) of the user's image. See 98 Managing User Images (p.269) to learn how to retrieve the image files. for more information about profile images

See the COMPASS RT API documentation for additional parameters.

Response: [Result \(from COMPASS RT API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

94.3 Related Tutorials

To	Refer To Tutorial
Create or update users in the account	93 Creating and Updating Users (p.257)

94.4 Example

`https://<operator>.tvaas.com/compass/AddProfile?name=tstProfile&pin_code=1234&client=json`

This request adds a new user `tstProfile`, with pin code 1234.

Sample response

The profile was created successfully:

```
{  
  "metadata": {  
    "request": "AddProfile",  
    "timestamp": 1456052605294  
  },  
  "response": {  
    "message": "Add profile operation finished successfully",  
    "properties": {  
      "profileId": "11202"  
    },  
    "status": "SUCCESS"  
  }  
}
```

A failure message is shown below. Profile creation failed as the account was not logged in.

```
{  
  "metadata": {  
    "request": "AddProfile",  
    "timestamp": 1456052461098  
  },  
  "response": {  
    "code": "INVALID_CREDENTIALS",  
    "message": "No anonymous user request is allowed: AddProfile",  
    "properties": {},  
    "status": "FAILURE"  
  }  
}
```



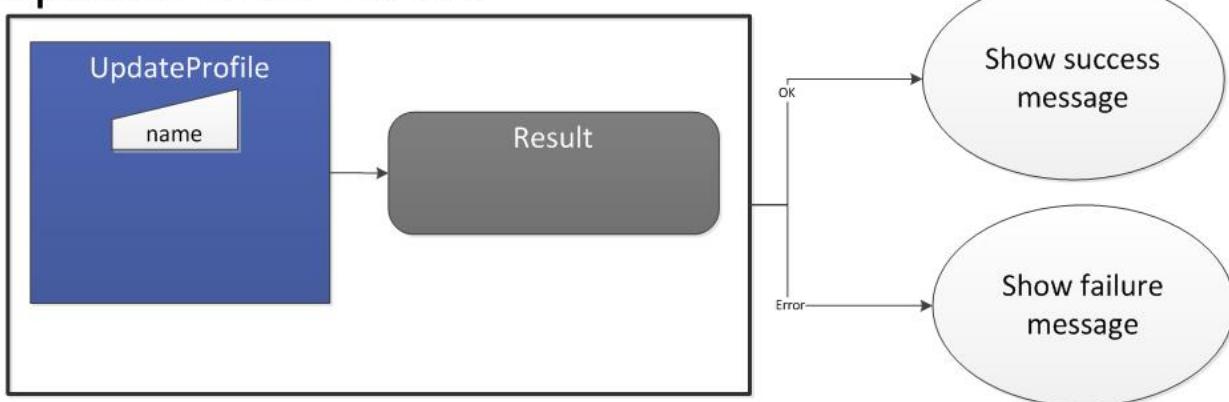
95 Update a User's Name

95.1 Overview

The name given to a user can be changed.

95.2 Requests and Responses

Update User Name



Request: [UpdateProfile \(from COMPASS RT API\)](#)

This request updates the logged in user's details. To change the user's name, only one parameter is required:

Parameter	Description
<i>name</i>	The new name for the user.

See the COMPASS RT API documentation for additional parameters.

Response: [Result \(from COMPASS RT API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

95.3 Related Tutorials

To	Refer To Tutorial
Create or update users in the account	93 Creating and Updating Users (p.257)

95.4 Example

<https://<operator>.tvaas.com/compass/UpdateProfile?name=Jim&client=json>

This request changes the name of the logged-in user to Jim.

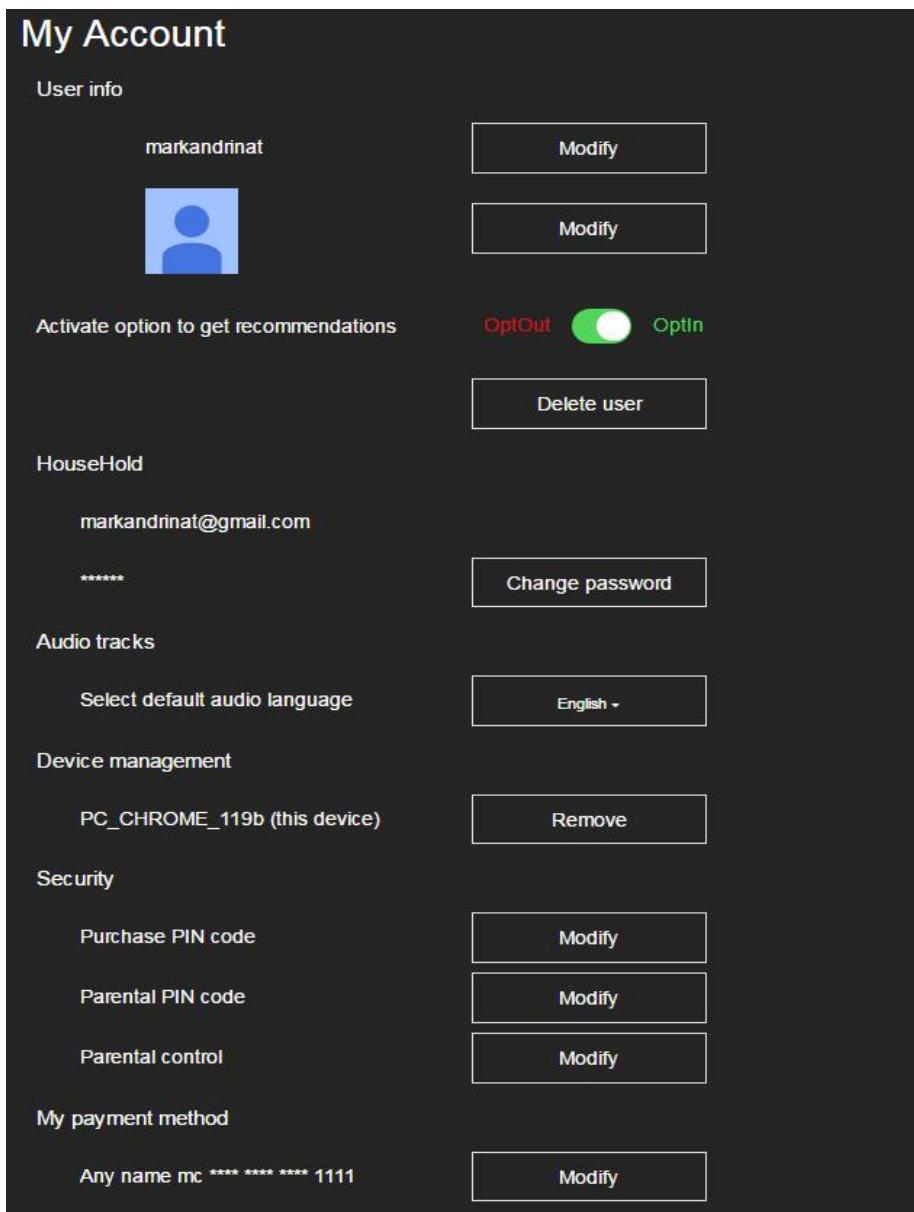
The user's details were updated successfully:

```
[{"metadata": {  
    "request": "UpdateProfile",  
    "timestamp": 1460882854280  
},  
 "response": {  
    "message": "Update profile operation finished successfully",  
    "properties": {},  
    "status": "SUCCESS"  
}}
```

96 Building a My Account Screen

96.1 Overview

A My Account screen should show current account and user information, and allow users to make changes as required. This tutorial gives an overview of what should be on a My Account screen, and provides links to other tutorials for implementation instructions.



96.2 Requests and Responses

These requests obtain much of the information to be displayed on a My Accounts screen:

- *GetHousehold* (from RiGHTv Front-end API) returns a range of information on the logged in account, including the parental PIN and account username.
- *GetMemberList* (from RiGHTv Front-end API) returns information on the account's users, including: first name, last name, purchase PIN, and e-mail address.
- *GetProfileList* (from COMPASS RT API) returns a range of information on the account's users, including the user name, images, and data usage opt-in/out status.

The table below refers you to the relevant tutorials for retrieving and changing the account and user information displayed on the My Account screen. The table indicates whether each item is set at the account or user level.

Section	Feature	Account/User Level	Refer to
Personalization	User Name	User	95 Update a User's Name (p.262)
	User Image		98 Managing User Images (p.269)
	Opt Out/Opt In		103 Managing User Usage Data Opt-In/out Status (p.278)
User	Email	Account	E-mail address of account. Cannot be changed.
	Password	Account	106 Updating and Resetting Account Passwords (p.283)
My Subscriptions		Account	158 My Offers / All Offers Screen (p.429)
Device Management		Account	113 Device Management (p.298)
Security	Purchase PIN	User	109 Purchase PIN Management (p.291)



Section	Feature	Account/User Level	Refer to
Parental Control	Parental Control PIN	Account	185 Parental Control (p.509)
	Parental Control Threshold	Account	192 Set the Parental Control Threshold (p.521)
Purchase Method		Account	120 Managing Payment Methods (p.318)

97 Get Account Information

97.1 Overview

The application will at times have to retrieve account information, for example, to get the account's preferred language or parental PIN. The *GetHousehold* request gets all account information for a specified account.

97.2 Requests and Responses

Request: [GetHousehold \(from RiGHTv Front-end API\)](#)

This request gets a range of information about the logged-in account. No parameters are submitted.

See the RiGHTv Front-end API guide.

Response: [Household \(from RiGHTv Front-end API\)](#)

Includes a range of information about the logged-in account. Examples of properties include: attributes:

Property	Description
<i>languageCode</i>	A two or three digit language code indicating the account's preferred language. See 179 Working with Languages (p.500) .
<i>parentalPIN</i>	The PIN code used to access adult content. See 185 Parental Control (p.509) .

For more information, see the RiGHTv Front-end API guide.

97.3 Example

`https://<operator>.tvaas.com/RTEFacade/GetHousehold?client=json`

This request requests information on the logged-in account.

A sample response of account information is shown below.



```
{  
    "metadata": {  
        "request": "GetHousehold",  
        "timestamp": 1460634563382  
    },  
    "response": {  
        "cluster": {},  
        "timezone": "INSTALLATION_TZ",  
        "billingMethod": "postpaid",  
        "externalId": "HH_Trial",  
        "discount": 0,  
        "groups": [],  
        "languageCode": "es",  
        "transactionsPaymentMode": "normal",  
        "spentTotal": 107,  
        "isHouseholdScope": true,  
        "responseElementType": "Household",  
        "paymentAuthentication": "always",  
        "spendingLimit": -1,  
        "phone": "",  
        "maxTerminals": "",  
        "extrafields": [],  
        "billingDay": 31,  
        "internalPrepaidBalance": 0,  
        "plannedInactivationDate": "",  
        "id": 10300,  
        "parentalPin": "1234",  
        "username": "1"  
    }  
}
```

98 Managing User Images

98.1 Overview

Each user in an account can upload and later change their own profile image. The user's image should be shown on the My Account screen, and on the User Selection screen.

User image files are stored on the COMPASS system. Each image has a unique image_id.

98.2 Related Tutorials

To	Refer to Tutorial
Create a My Account screen	96 Building a My Account Screen (p.264)
Get a user's current image	99 Get a User's Image (p.270)
Upload a user's initial image	100 Create a User Image (p.272)
Replace a user's image	101 Update a User Image (p.274)
Delete a user's image	102 Delete a Profile Image (p.276)



99 Get a User's Image

99.1 Overview

The image_id of a user's image, if one exists, is returned as part of the response to the *GetProfile* request.

99.2 Requests and Responses

Request: [GetProfileList](#) (from COMPASS RT API)

This request gets the list of existing users in the account. No parameters are required.

Response: [Profile](#) (from COMPASS RT API)

Provides information about each user in the account . For displaying the user's image, the most relevant property is:

Property	Description
<i>url</i>	Within the images array, the url of the user's image file. Image is retrieved using GetProfileImage. (See the example below.)

99.3 Related Tutorials

To	Refer To Tutorial
Manage a user's image	98 Managing User Images (p.269)

99.4 Example

<https://<operator>.tvaas.com/compass/GetProfileList?client=json>

This request gets the list of users in the account.

The response for a single user is shown below.

```
        "name": "Colin Coltrain",
        "externalId": "10702",
        "pinCodeExist": 0,
        "id": 10702,
        "profileProperties": [],
        "suspended": 0
    },
    {
        "image": "",
        "images": [
            {
                "name": "ProfileImage",
                "description": "",
                "id": 154,
                "type": "Internal",
                "url": "/GetProfileImage?image_id=154" ←
            }
        ]
    }
}
```

The user's image id is shown to be 154.

The request below downloads that image.

https://<operator>.tvaas.com/compass/GetProfileImage?image_id=154

100 Create a User Image

100.1 Overview

A user should be able to create a user image if one does not exist already.

Image binaries should be provided to COMPASS via HTTP Post. The ‘content-type’ HTTP header should be set to the image type (for example “image/jpeg”)

100.2 Requests and Responses

Request: [CreateProfileImage](#) (from COMPASS RT API)

This request creates an image for the user. The following parameter must be set:

Parameter	Description
<i>image_name</i>	A friendly name for the image. Must be unique per user.
<i>image_url</i>	The url of the image.

Response: [ProfileImage](#) (from COMPASS RT API)

Provides information about the image just created. The following properties are returned:

Property	Description
<i>id</i>	The ID of the image. Used when retrieving images.
<i>name</i>	The name of the image.
<i>url</i>	The reference to the image location.

100.3 Related Tutorials

To	Refer to Tutorial
Manage a user’s image	98 Managing User Images (p.269)

100.4 Example

`https://<operator>.tvaas.com/compass/CreateProfileImage?image_name=WilliamSmith&image_url=/images/WilliamSmith1.jpg`

This creates a new user image called WilliamSmith, using the file previously uploaded as WilliamSmith1.jpg.

A sample response is shown below. Note that the ID assigned to this image is 303.

```
[{"metadata": {  
    "request": "CreateProfileImage",  
    "timestamp": 1460881136060  
},  
 "response": {  
    "name": "WilliamSmith",  
    "description": "",  
    "id": 303,  
    "type": "External",  
    "url": "/images/WilliamSmith.jpg"  
}  
]
```

101 Update a User Image

101.1 Overview

A user should be able to update his user's image with an image that already exists within the system.

Image binaries should be provided to COMPASS via HTTP Post. The 'content-type' HTTP header should be set to the image type (for example "image/jpeg")

101.2 Requests and Responses

Request: [UpdateProfileImage \(from COMPASS RT API\)](#)

This request creates an image for the user. The following parameter must be set:

Parameter	Description
<i>image_id</i>	The ID of the image to be used.

Response: [ProfileImage \(from COMPASS RT API\)](#)

Provides information about the image just created. The following properties are returned:

Property	Description
<i>id</i>	The ID of the image. Used when retrieving images.
<i>name</i>	The name of the image.
<i>url</i>	The reference to the image location.

101.3 Related Tutorials

To	Refer To Tutorial
Manage a user's image	98 Managing User Images (p.269)

101.4 Example

`https://<operator>.tvaas.com/compass/UpdateProfileImage?image_id=301&client=json`

This updates the user's image to the image with ID 301.

A sample response is shown below.

```
[{"metadata": {  
    "request": "CreateProfileImage",  
    "timestamp": 1460881136060  
},  
 "response": {  
    "name": "WilliamSmith",  
    "description": "",  
    "id": 301,  
    "type": "External",  
    "url": "/images/WilliamSmith.jpg"  
}}]
```

102 Delete a Profile Image

102.1 Overview

A user should be able to delete his user's image from the system.

102.2 Requests and Responses

Request: [DeleteProfileImage](#) (from COMPASS RT API)

This request deletes an image from the system. The following parameter must be set:

Parameter	Description
<i>image_id</i>	The ID of the image to be used.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the *DeleteProfileImage* action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

102.3 Related Tutorials

To	Refer To Tutorial
Manage a user's image	98 Managing User Images (p.269)

102.4 Example

`https://<operator>.tvaas.com/compass/DeleteProfileImage?image_id=301&client=json`

This deletes the image with ID 301.

A sample response is shown below. The image was successfully deleted.

```
[{"metadata": {  
    "request": "DeleteProfileImage",  
    "timestamp": 1460881625968  
},  
"response": {  
    "message": "Deleted Profile Image",  
    "properties": {},  
    "status": "SUCCESS"  
}]
```

103 Managing User Usage Data Opt-In/out Status

103.1 Overview

The Service Delivery Platform can monitor individual users' viewing patterns and use that information to improve the content recommendations provided. Users can choose whether to opt-in or out of this service. Usage data is not recorded for a user who has chosen to opt-out.

103.2 Related Tutorials

To	Refer to Tutorial
Get the user's current opt-in/out status	104 Get the Current Opt-in/out Status (p.279)
Change the user's opt-in/out status	105 Opt-in and Opt-out (p.281)

104 Get the Current Opt-in/out Status

104.1 Overview

Users manage their opt-in/out status from a My Account screen. The screen should show their current opt-in/out status.

104.2 Requests and Responses

Request: [GetProfileList](#) (from COMPASS RT API)

This request gets details of existing users in the account. No parameters are required.

Response: [Profile](#) (from COMPASS RT API)

Provides information about each user in the account. For reading the current opt-in/out status, the following properties are relevant:

Property	Description
<i>externalid</i>	The ID number of the user.
<i>suspended</i>	Returns 0 if the user has opted-in, or 1 if the user has opted-out.

104.3 Related Tutorials

To	Refer to Tutorial
Manage a user's opt-in/out status	103 Managing User Usage Data Opt-In/out Status (p.278)

104.4 Example

<https://<operator>.tvaas.com/compass/GetProfileList?client=json>

This request gets the list of users in the account.

The response for a single user is shown below. The *suspended* value is 0, so this user has opted-in.

```
  "name": "Colin Coltrain",
  "externalId": "10702",
  "pinCodeExist": 0,
  "id": 10702,
  "profileProperties": [],
  "suspended": 0 ←
},
{
  "image": "",
  "images": [
    {
      "name": "ProfileImage",
      "description": "",
      "id": 154,
      "type": "Internal",
      "url": "/GetProfileImage?image_id=154"
    }
  ]
}
```

105 Opt-in and Opt-out

105.1 Overview

Users manage their opt-in/out status from a My Account screen. The screen should show them to change their opt-in/out status.

105.2 Requests and Responses

Request: [OptInProfile & OptOutProfile \(from COMPASS RT API\)](#)

These requests opt the logged-in user in our out of the data usage function. No parameters are required.

Response: [Result \(from COMPASS RT API\)](#)

Indicates whether the *OptInProfile* or *OptOutProfile* action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .

105.3 Related Tutorials

To	Refer to Tutorial
Manage a user's opt-in/out status	103 Managing User Usage Data Opt-In/out Status (p.278)

105.4 Example

<https://<operator>.tvaas.com/compass/OptInProfile?client=json>

This request opts the user into the data usage function.

<https://<operator>.tvaas.com/compass/OptOutProfile?client=json>

This request opts the user out of the data usage function.

The response for an opt-in is shown below.



```
[{"  
  "metadata": {  
    "request": "OptInProfile",  
    "timestamp": 1460884297239  
  },  
  "response": {  
    "message": "Opt In profile operation finished successfully",  
    "properties": {},  
    "status": "SUCCESS"  
  }  
}
```



106 Updating and Resetting Account Passwords

106.1 Overview

Users need a password to log in to their accounts.

This password can be changed in two ways:

- Resetting the password
- Updating the password

Note

Password reset and update is only possible for Online accounts, not for operator accounts. See [118 Identify Account Type \(p.308\)](#).

106.2 Related Tutorials

To	Refer To Tutorial
Create a login screen	80 Create a Login Screen (p.227)
Create a My Account screen	96 Building a My Account Screen (p.264)
Reset an account's password	107 Reset an Account's Password (p.284)
Change an account's password	108 Update an Account's Password (p.288)

107 Reset an Account's Password

107.1 Overview

An account's password can be reset. Most typically, a user who has forgotten their password will ask for the password to be reset. The password reset operation sends an email to the account's e-mail address, with a link to a screen where the new password is displayed. The user does not have to be logged in to reset the password.

The user can optionally be subjected to a Captcha challenge.

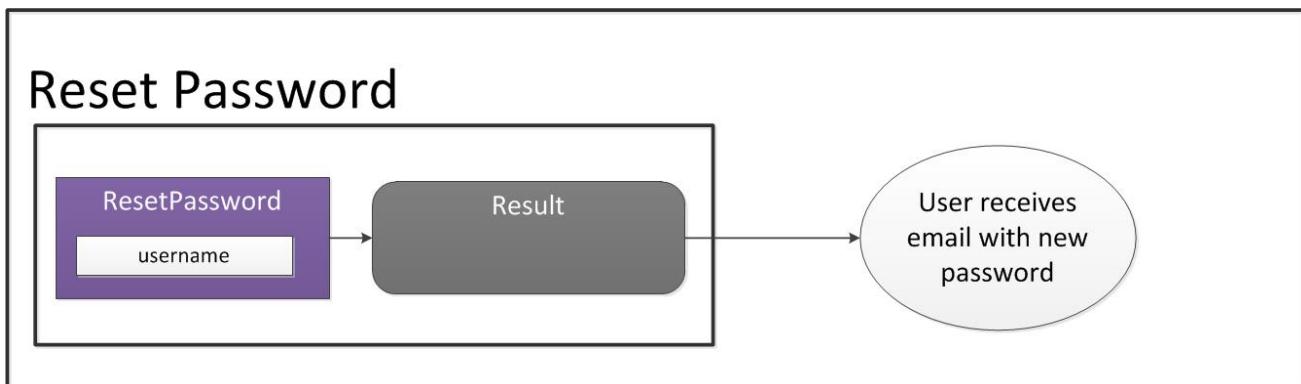
Note

Password reset is only possible for Online accounts, not for operator accounts. See [118 Identify Account Type \(p.308\)](#).

107.2 Requests and Responses

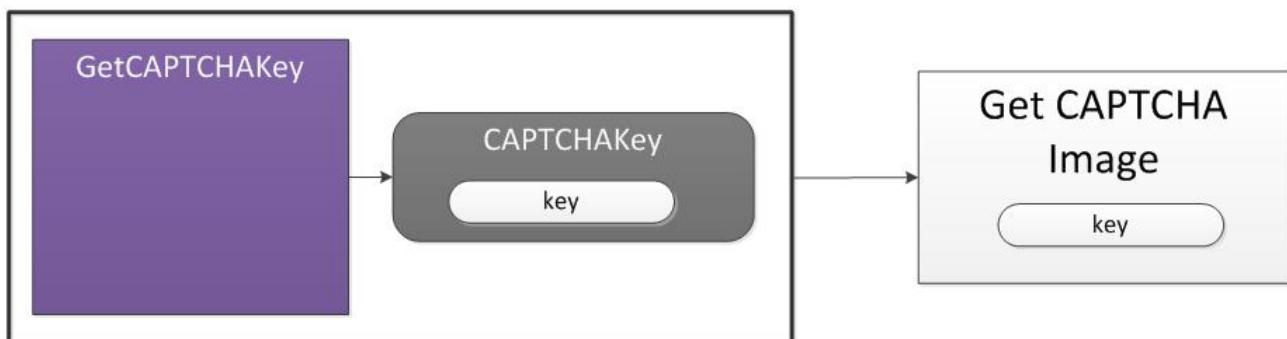
The process of resetting a password can optionally include a Captcha image challenge.

Process Flow without Captcha Challenge

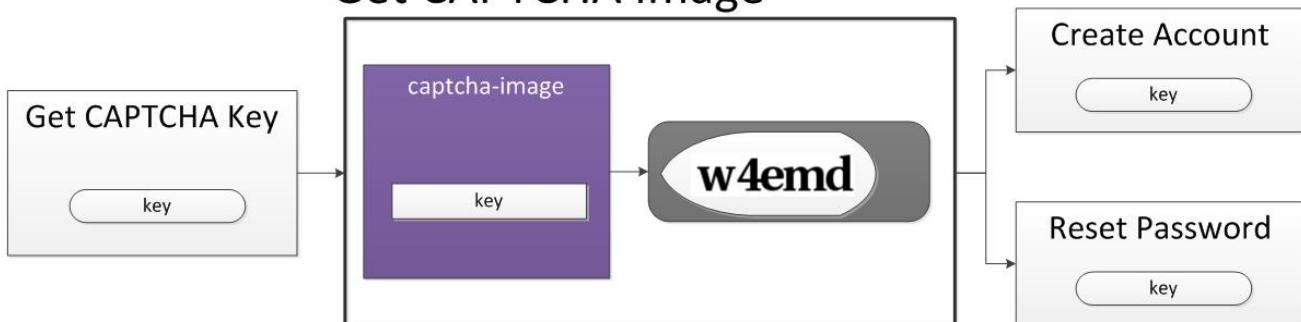


Process Flow with Captcha Challenge

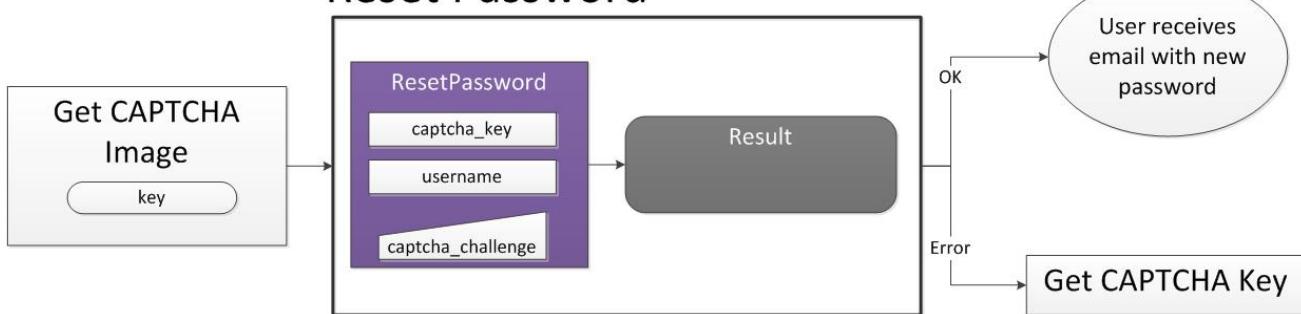
Get CAPTCHA Key



Get CAPTCHA Image



Reset Password



Request: [ResetPassword](#) (from RiGHTv Front-end API)

This request resets the user's password. The action sends an email to the user with a new password. The following parameters are required:

Parameter	Description
<i>username</i>	The account name (e-mail address)
The following parameters are only relevant when implementing a Captcha challenge:	
<i>captcha_key</i>	The Captcha key obtained using the GetCaptchaKey request.
<i>captcha_challenge</i>	The user's answer to the Captcha challenge.

Response: **Result** (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

107.3 Related Tutorials

To	Refer to Tutorial
Update and reset an account's password	80 Create a Login Screen (p.227)
Determine whether the account is an online account or a traditional account	118 Identify Account Type (p.308)
Get a CAPTCHA key	83 Get CAPTCHA Key (p.233)
Get a CAPTCHA image	84 Get CAPTCHA Image (p.235)

107.4 Example

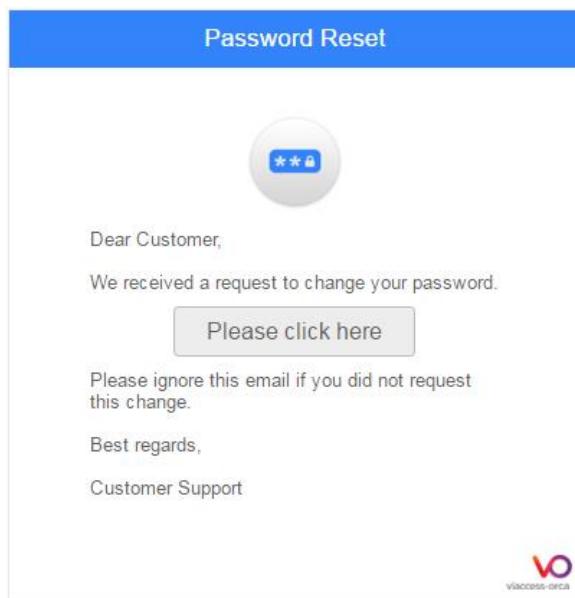
https://<operator>.tvaas.com/RTEFacade/ResetPassword?username=9952349@gmail.com&captcha_key=1460453137534&captcha_challenge=6rpcb&client=json

This request resets the password. It supplies the username and the Captcha challenge information.

The sample below indicates successful password reset.

```
[{"metadata": {
    "request": "ResetPassword",
    "timestamp": 1460453235553
},
"response": {
    "message": "Successfully sent email with reset password instructions",
    "status": "SUCCESS"
}]
```

The user receives an e-mail as below.



108 Update an Account's Password

108.1 Overview

An account's password can be updated. The user must be logged in to the application to update the password. The application should request that the user enter the old password, and then the new password twice for verification.

The screenshot shows a dark-themed 'Change password' interface. It features three text input fields: 'Current password', 'New password', and 'New password again'. Below these fields is a 'Save' button.

Note

Only Online accounts can update their passwords, operator accounts cannot do so. See [118 Identify Account Type \(p.308\)](#).

108.2 Requests and Responses

Request: [UpdatePassword](#) (from RiGHTv Front-end API)

This request updates the account's password with a new one. Significant parameters are:

Parameter	Description
<i>old_password</i>	The current password.
<i>new_password</i>	The new password.
<i>new_password_retyped</i>	The new password, again.

Response: Result (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

108.3 Related Tutorials

To	Refer to Tutorial
Update and reset an account's password	80 Create a Login Screen (p.227)
Determine whether the account is an online account or a traditional account	118 Identify Account Type (p.308)

108.4 Example

https://<operator>.tvaas.com/RTEFacade/UpdatePassword?old_password=Mgk2VQLFyl&new_password=rainbow1&new_password_retyped=rainbow1&client=json

This request updates the password of the logged-in account to rainbow1.

The sample below indicates a successful password update.



```
[{"metadata": {  
    "request": "UpdatePassword",  
    "timestamp": 1460454595698  
},  
"response": {  
    "message": "Password Updated Successfully",  
    "status": "SUCCESS"  
}}
```

109 Purchase PIN Management

109.1 Overview

When logging in to the application, the customer first has to log into their account, using the account's ID (e-mail address) and password. The customer then identifies his or her user within the account. Each user has their own purchase PIN code.

The user is prompted to enter their purchase PIN whenever purchasing content. The PIN must be verified before any purchases are allowed. The operator can reset the user's purchase PIN in the Account Details in the TVE Console.

The client application is responsible for verifying the PIN that the user enters against the PIN stored in the database.

109.2 Related Tutorials

To	Refer to Tutorial
Create a My Account screen	96 Building a My Account Screen (p.264)
Get the user's current PIN	110 Get Purchase PIN (p.292)
Reset the user's PIN	111 Reset a User's PIN (p.294)
Allow the user to choose a new PIN	112 Update a User's PIN (p.296)

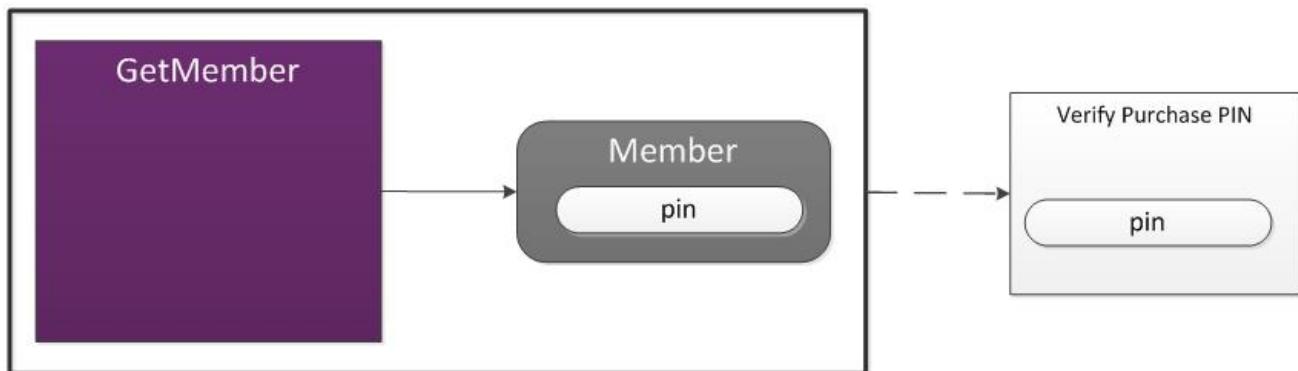
110 Get Purchase PIN

110.1 Overview

The application must request that the user enter his or her PIN when making a purchase. The application itself is responsible for verifying that the correct PIN was entered.

The PIN should be retrieved from the back end once, and then be stored in the applications' local cache. When the user is requested to enter the PIN before purchasing content, the application should validate the PIN against the PIN saved in the cache before sending an order request to the server. If the PIN entered is incorrect, display an error message requesting the user to reenter the PIN.

110.2 Requests and Responses



Request: [GetMember](#) (from RiGHTv Front-end API)

This request gets details of the logged-in user, including the purchase PIN.

No parameters need be specified.

Response: [Member](#) (from RiGHTv Front-end API)

The entity includes the following attributes for all users.

Property	Description
<i>pin</i>	The user's PIN code.

110.3 Related Tutorials

To	Refer to Tutorial
Manage a user's PIN	109 Purchase PIN Management (p.291)
Purchase content	119 The Purchase Process (p.312)

110.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetMember?client=json>

This request gets details of the logged-in user.

The sample below shows the details of one user. The user's PIN is 0000.

```
{
  "metadata": {
    "request": "GetMember",
    "timestamp": 1490850514683
  },
  "response": {
    "responseElementType": "Member",
    "firstName": "tttt",
    "lastName": "",
    "spendingLimit": -1,
    "isMainMember": true,
    "pin": "0000", -----|
    "extrafields": [],
    "externalId": "35701",
    "id": 35701,
    "email": "eran.rodrig@gmail.com",
    "spentTotal": 0
  }
}
```

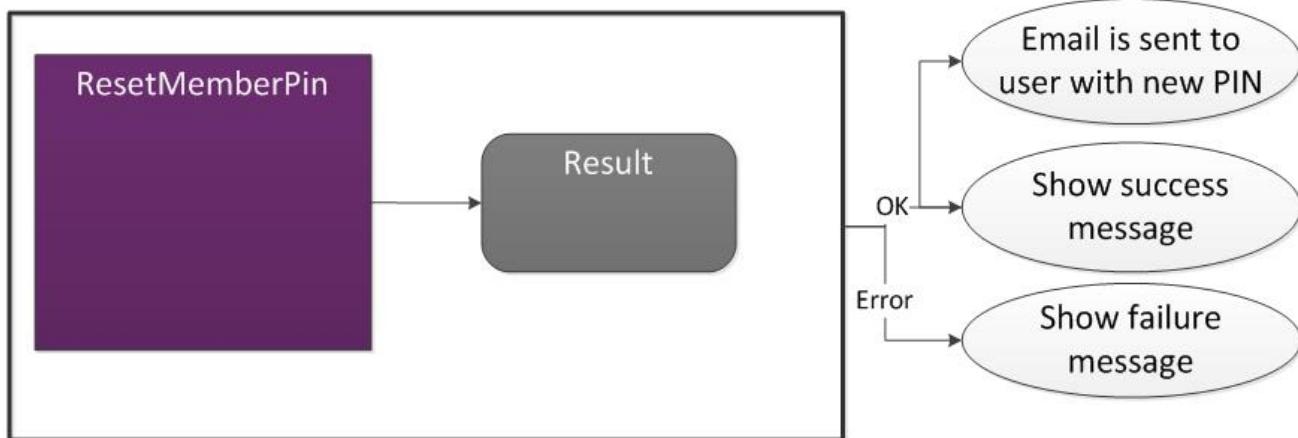
111 Reset a User's PIN

111.1 Overview

A user's PIN can be reset. Most typically, a user who has forgot their PIN will ask for the PIN to be reset. The PIN reset operation generates a new 4-digit PIN that is then emailed to the user.

Note: Resetting an operator user's PIN is only possible if an e-mail address has been defined.

111.2 Requests and Responses



Request: [ResetMemberPin](#) (from RiGHTv Front-end API)

This request resets the user's PIN. The Service Delivery Platform sends an e-mail to the user with a link to a screen that displays a new 4-digit purchase PIN code. No parameters are supplied.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

111.3 Related Tutorials

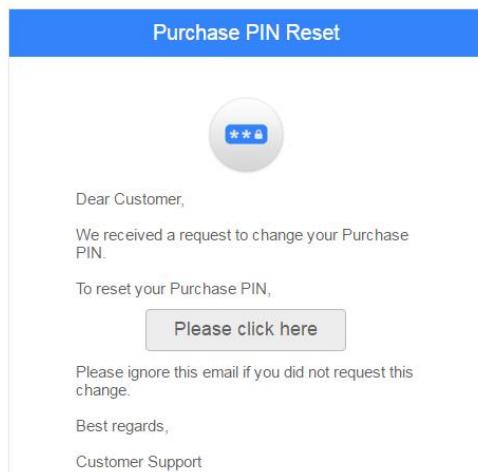
To	Refer to Tutorial
Manage a user's PIN	109 Purchase PIN Management (p.291)
Purchase content	119 The Purchase Process (p.312)

111.4 Example

<https://<operator>.tvaas.com/RTEFacade/ResetMemberPin?client=json>

This request initiates the PIN reset process. The user will receive an e-mail with a link to a screen that displays a new 4-digit PIN.

The user receives an e-mail as shown below.

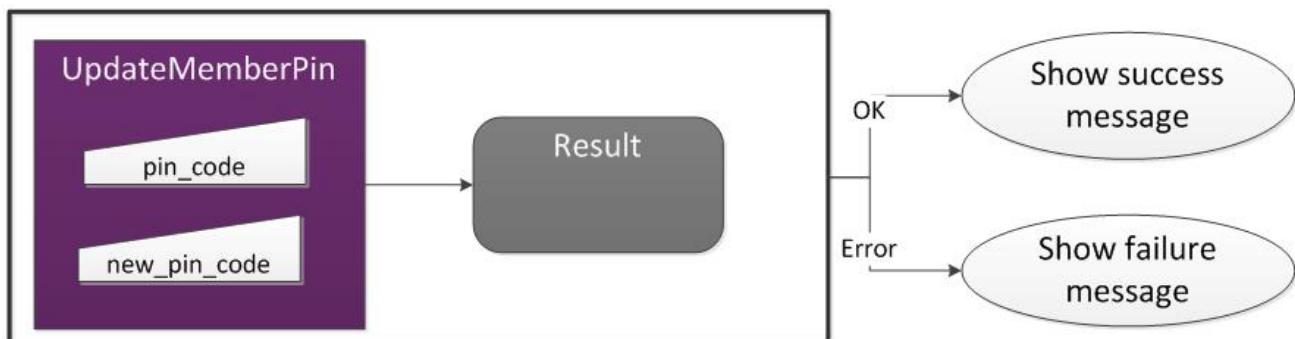


112 Update a User's PIN

112.1 Overview

A user's PIN can be updated. The user is required to enter his or her old PIN, and the new PIN.

112.2 Requests and Responses



Request: [UpdateMemberPin](#) (from RiGHTv Front-end API)

This request updates the user's PIN with a new one. Significant parameters are:

Property	Description
<i>new_pin_code</i>	The new 4-digit PIN.
<i>pin_code</i>	The current PIN code used to identify the user requesting a new PIN. If not specified, the logged-in user is assumed.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

112.3 Related Tutorials

To	Refer To Tutorial
Manage a user's PIN	109 Purchase PIN Management (p.291)
Purchase content	119 The Purchase Process (p.312)

112.4 Example

https://<operator>.tvaas.com/RTEFacade/UpdateMemberPin?new_pin_code=3652&client=json

This request updates the PIN used by the logged-in member to 3652.

The sample below indicates a successful PIN update.

```
[{"metadata": {  
    "request": "UpdateMemberPin",  
    "timestamp": 1460273678886  
},  
"response": {  
    "message": "Updated member PIN successfully",  
    "status": "SUCCESS"  
}}
```



113 Device Management

113.1 Overview

Device registration is required in order to view content. A user's device (also referred to as a terminal) is registered the first time the user logs in using the device. During subsequent logins, the device's ID is retrieved and checked against the list of devices registered to the account. If the device is not in the list, it must be registered.

The operator can set a limit for the number of devices an account can register as well as for the number of device removals that can be performed during a specific time period.

The account holder should also be able to deregister devices (especially if a limit is placed on the number of devices allowed), and to change the name of a registered device.

113.2 Related Tutorials

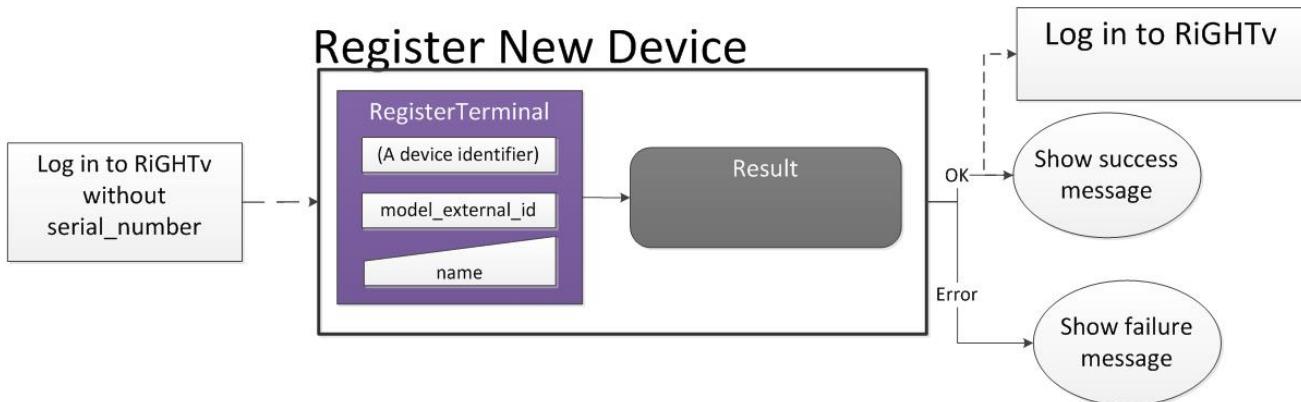
To	Refer To Tutorial
Create a My Account screen	96 Building a My Account Screen (p.264)
Register a new device to the account	114 Register a New Device (p.299)
Get a list of devices registered to the account	115 Get Device List (p.301)
Unregister a device from an account	116 Remove Device (p.304)
Change the name of a registered device	117 Update Device Name (p.306)

114 Register a New Device

A user's device (terminal) is registered the first time the user logs in using the device.

As explained in [87 Log into RiGHTv Account \(p.243\)](#), if the login fails when the device serial number is provided, the application should log the user in without the device serial number, and should then register the device.

114.1 Requests and Responses



Request: [RegisterTerminal \(from RiGHTv Front-end API\)](#)

This request registers the device. The following parameters are required:

Parameter	Description
A unique device identifier	Any of the following can be used as the device identifier: <ul style="list-style-type: none"> <i>serial_number</i> <i>mac_address</i> <i>smart_card_id</i> <i>manufacturer_id</i>
<i>model_external_id</i>	A string that identifies the device model (e.g. iPad) in the application.
<i>name</i>	A unique name for the device (can be a concatenation of the model ID and an index number).

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: **Result** (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following properties:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server.

For more properties, see the RiGHTv Front-end API Developers Guide.

114.2 Related Tutorials

To	Refer to Tutorial
Manage devices	96 Building a My Account Screen (p.264)
Log the account into RiGHTv	87 Log into RiGHTv Account (p.243)

114.3 Example

https://<operator>.tvaas.com/RTEFacade/RegisterTerminal?serial_number=R21F72WF9WF&model_external_id=PC&name=PC1&client=json

This request adds a device specified by the serial number. The device is a PC.

The response below indicates successful registering of the device.

```
[{"metadata": {
    "request": "RegisterTerminal",
    "timestamp": 1458128545522
},
"response": {
    "message": "Terminal Registered Successfully",
    "status": "SUCCESS"
}]
```

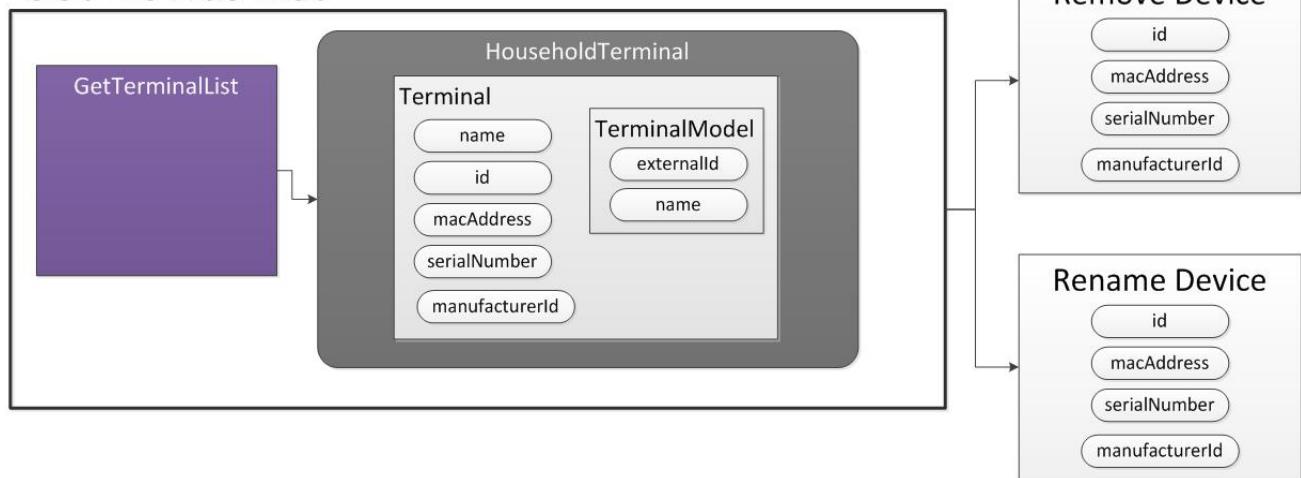
115 Get Device List

115.1 Overview

The application will at times need to get a list of devices registered to the account. For example a list of registered devices should be shown on the My Settings screen so that the user can choose to unregister a device or to update the name of a registered device.

115.2 Requests and Responses

Get Device List



Request: [GetTerminalList](#) (from RiGHTv Front-end API)

This request gets the list of devices registered to the account of the logged-in user. No parameters need be specified.

Response: [HouseholdTerminals](#) (from RiGHTv Front-end API)

A list of the account's devices including:

Property	Description						
<i>terminals</i>	A list of devices, providing for each: <ul style="list-style-type: none"> • id, the device internal ID • macAddress • serialNumber • manufacturerId 						
<i>name</i>	The name given to the device.						
<i>model</i>	The terminal's model: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The terminal model external ID.</td></tr> <tr> <td><i>name</i></td><td>The model name.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The terminal model external ID.	<i>name</i>	The model name.
Property	Description						
<i>externalId</i>	The terminal model external ID.						
<i>name</i>	The model name.						

For more information see the RiGHTv Front-end API Developers Guide.

115.3 Related Tutorials

To	Refer To Tutorial
Manage devices	96 Building a My Account Screen (p.264)
Create a My Account screen	96 Building a My Account Screen (p.264)

115.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetTerminalList?&client=json`

This request gets the device the list of devices registered to the account of the logged-in user.

In the response below the logged-in account has two devices with the names “MyGalaxy” and “iPad Air”.



```
{
  "metadata": {
    "request": "GetTerminalList",
    "timestamp": 1459839953535
  },
  "response": {
    "householdId": 23504,
    "responseElementType": "HouseholdTerminals",
    "terminals": [
      {
        "serialNumber": "5555",
        "terminalModelId": "",
        "ip": "",
        "manufacturerId": "",
        "responseElementType": "Terminal",
        "macAddress": "",
        "name": "MyGalaxy", 
        "registrationDate": 1459839946339,
        "model": {
          "responseElementType": "TerminalModel",
          "vodCasNames": [
            "Viaccess-Purple-DRM"
          ],
          "liveVsName": "RiGHTvLive",
          "vodVsName": "amazon-s3",
          "categoryExternalId": "Smartphone",
          "highestDefinitionForVideo": "",
          "name": "AndroidSmartphone",
          "externalId": "AndroidSmartphone",
          "vodCasName": "Viaccess-Purple-DRM",
          "liveCasName": "Viaccess-Purple-DRM",
          "vodVsNames": [
            "amazon-s3"
          ]
        }
      },
      {
        "id": 37301, 
        "smartCardId": "",
        "operatorId": "",
        "extraFields": []
      }
    ]
  }
}
```

The JSON response shows two terminal objects. The first terminal has a name of "MyGalaxy". The second terminal has an id of 37301. Red arrows point to these specific fields to highlight them.

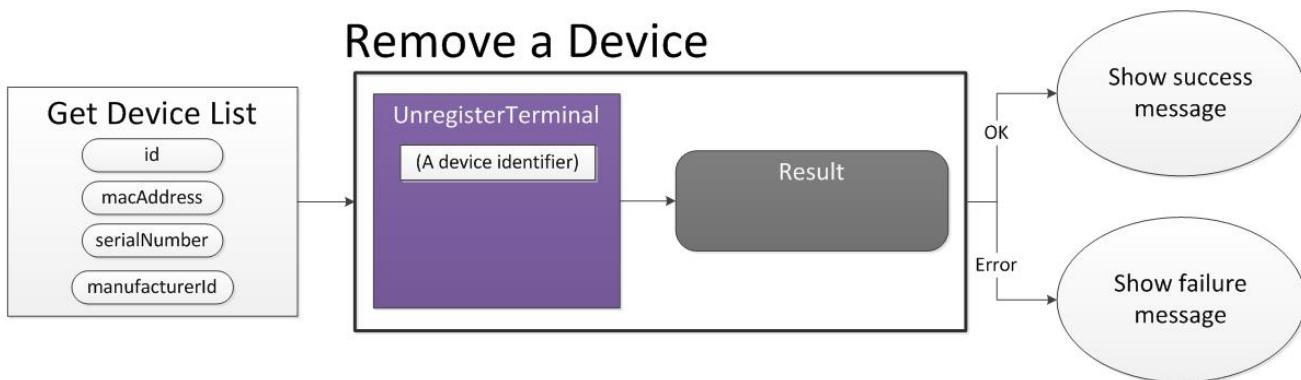


116 Remove Device

116.1 Overview

A device can be removed from the system (deregistered). This may be done, for example, when a user wants to register a different device and has a limit on the amount of devices that can be registered. The user can log in from any device belonging to the account.

116.2 Requests and Responses



Request: [UnregisterTerminal](#) (from RiGHTv Front-end API)

This request removes the device. The following parameters are required:

Parameter	Description
A unique device identifier	Any of the following can be used as the device identifier: <i>serial_number</i> <i>mac_address</i> <i>smart_card_id</i> <i>manufacturer_id</i>

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

116.3 Related Tutorials

To	Refer to Tutorial
Manage devices	96 Building a My Account Screen (p.264)
Create a My Account screen	96 Building a My Account Screen (p.264)

116.4 Example

https://<operator>.tvaas.com/RTEFacade/UnregisterTerminal?serial_number=5555&client=json

This request removes the MyGalaxy device specified by the serial number 5555.

```
{  
  "metadata": {  
    "request": "UnregisterTerminal",  
    "timestamp": 1459840224199  
  },  
  "response": {  
    "message": "unregistered terminal successfully",  
    "status": "SUCCESS"  
  }  
}
```

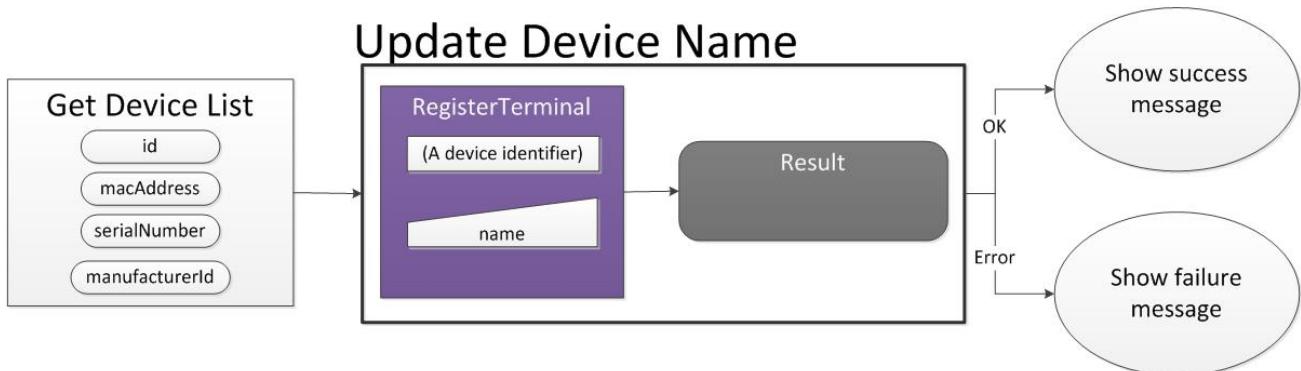


117 Update Device Name

117.1 Overview

A user can change the name used to identify the device used in the system.

117.2 Requests and Responses



Request: [UpdateTerminal](#) (from RiGHTv Front-end API)

Updates the name of the specified device. The following parameters are required:

Parameter	Description
A unique device identifier	Any of the following can be used as the device identifier: <ul style="list-style-type: none"> <i>serial_number</i> <i>mac_address</i> <i>smart_card_id</i> <i>manufacturer_id</i>
<i>name</i>	The new name to be used to identify the device in the system. For example, <i>Kate's tablet</i> .

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .
<i>message</i>	A short message from the server

117.3 Related Tutorials

To	Refer To Tutorial
Manage devices	96 Building a My Account Screen (p.264)
Create a My Account screen	96 Building a My Account Screen (p.264)

117.4 Example

`https://<operator>.tvaas.com/RTEFacade/UpdateTerminal?serial_number=R21F72WF9WF&name=MyAndroidPhone&client=json`

This actions updates the name of the specified device to “MyAndroidPhone”.

```
[{"metadata": {  
    "request": "UpdateTerminal",  
    "timestamp": 1458128751178  
},  
 "response": {  
    "message": "Updated terminal successfully",  
    "status": "SUCCESS"  
}}
```

118 Identify Account Type

118.1 Overview

Two types of accounts should be supported by the client application:

- Operator accounts

These subscribers' accounts are debited using the operator's billing systems. They either pay in advance, or are debited at the end of a billing cycle. Operator accounts are limited in the extent to which they can manage their account details and purchased content.

- Online accounts

These users access the service over the Internet, and are debited by credit card, through the Service Delivery Platform, as they purchase content or services. Online accounts can be given a high level of control over their account details and purchased content.

This Guide indicates where functionality is unavailable to operator accounts.

This tutorial explains how to determine whether the logged-in user is an online or operator subscriber.

118.2 Requests and Responses

Request: [GetHousehold](#) (from RiGHTv Front-end API)

This request gets a range of information about the logged-in account. No parameters are submitted.

See the RiGHTv Front-end API guide.

Response: [Household](#) (from RiGHTv Front-end API)

Includes a range of information about the logged-in account. To determine the type of customer:

Property	Description
<i>billingMethod</i>	If returns online payment method, the account is an Online account. Otherwise, the account is an operator account.

118.3 Related Tutorials

There are multiple situations in which the client application will have to determine whether the logged-in account is an online or traditional account. These are listed below.

To	Refer to Tutorial
Create a login screen	80 Create a Login Screen (p.227)
Update or reset an account's password	106 Updating and Resetting Account Passwords (p.283)
Purchase content or subscribe to services	119 The Purchase Process (p.312)
Register an online payment method	125 Register a New Payment Method via the Agnostic Payment Service (p.329)
Unsubscribe from a Service Plan	132 Unsubscribing from a Service Plan (p.351)
Unsubscribe from an SVOD service	137 Unsubscribing from an SVOD Service (p.362)
Unsubscribe from a Channel Package	141 Unsubscribe from a Channel Package (p.371)
Unsubscribe from a Catch-Up service	169 Unsubscribing from a Catch-Up Service (p.474)

118.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHousehold?client=json>

This request requests information on the logged-in account.

A sample response of account information is shown below. *billingMethod* is set to `post paid` (and not to `online billing` method.) This account is an operator account.



```
[{"metadata": {  
    "request": "GetHousehold",  
    "timestamp": 1461128670972  
},  
"response": {  
    "cluster": {},  
    "timezone": "INSTALLATION_TZ",  
    "billingMethod": "postpaid",  
    "externalId": "HH_Trial",  
    "discount": 0,  
    "groups": [],  
    "languageCode": "es",  
    "transactionsPaymentMode": "normal",  
    "spentTotal": 107,  
    "isHouseholdScope": true,  
    "responseElementType": "Household",  
    "paymentAuthentication": "always",  
    "spendingLimit": -1,  
    "phone": "",  
    "maxTerminals": "",  
    "extrafields": [],  
    "billingDay": 31,  
    "internalPrepaidBalance": 0,  
    "plannedInactivationDate": "",  
    "id": 10300,  
    "parentalPin": "1234",  
    "username": "1"  
}}
```

The response for a different account shows *billingMethod* set to online payment method, indicating that this is an Online account.

```
[{"metadata": {  
    "request": "GetHousehold",  
    "timestamp": 1461128751947  
},  
"response": {  
    "cluster": {},  
    "timezone": "INSTALLATION_TZ",  
    "billingMethod": "online payment method", ←  
    "externalId": "61901",  
    "discount": 0,  
    "groups": [],  
    "languageCode": "en",  
    "transactionsPaymentMode": "normal",  
    "spentTotal": 0,  
    "isHouseholdScope": true,  
    "responseElementType": "Household",  
    "paymentAuthentication": "never",  
    "spendingLimit": -1,  
    "phone": "",  
    "maxTerminals": "",  
    "extrafields": [],  
    "billingDay": 31,  
    "internalPrepaidBalance": 0,  
    "plannedInactivationDate": "",  
    "id": 61901,  
    "parentalPin": "0000",  
    "username": "9952349@gmail.com"  
}  
}
```



119 The Purchase Process

119.1 Overview

Account holders can purchase content (TVOD and Movie Packages) and subscribe to services (e.g. SVOD, Channel Packages, Service Plan, Catch-Up) through the PC portal.

Online account holders pay by credit card at the time of purchase, whereas traditional account holders are charged by the operator through external systems.

Note

Both Online and Operator accounts can also purchase content and subscribe to services from their mobile devices through the app stores. For more information, see [In-App Purchasing](#)

The purchase process thus differs by account type. This tutorial provides outlines of the:

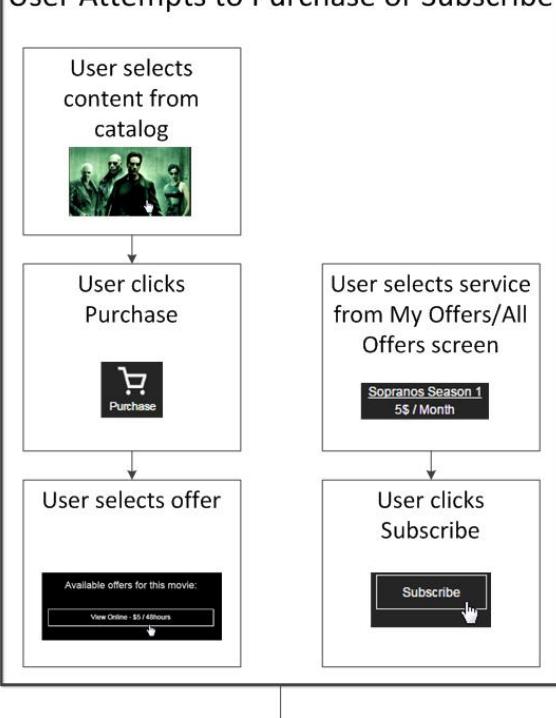
- [119.2 Online Account Purchase Process \(p.312\)](#)
- [119.3 Traditional Account Purchase Process \(p.315\)](#)

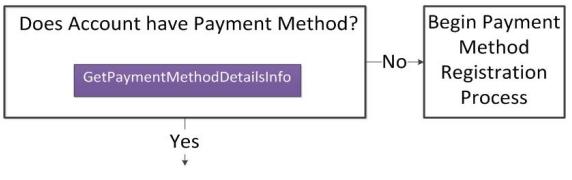
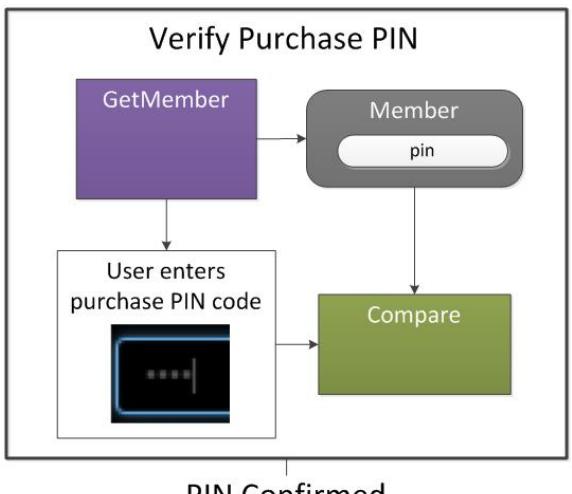
To determine whether the logged-in account is an online or traditional account, see [118 Identify Account Type \(p.308\)](#).

119.2 Online Account Purchase Process

The process to be followed when an online account attempts to purchase content or subscribe to a service is outlined below.



Steps	References	
	To	Refer To Tutorial
<p>User Attempts to Purchase or Subscribe</p>  <pre> graph TD A[User selects content from catalog] --> B[User clicks Purchase] B --> C[User selects offer] C --> D[User clicks Subscribe] </pre>	List offers for VOD Content	36 List Offers (p.115)
	List Channel Packages through which a channel can be purchased	72 List Offers for a Channel (p.211)
	List all available subscription offers on the My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)

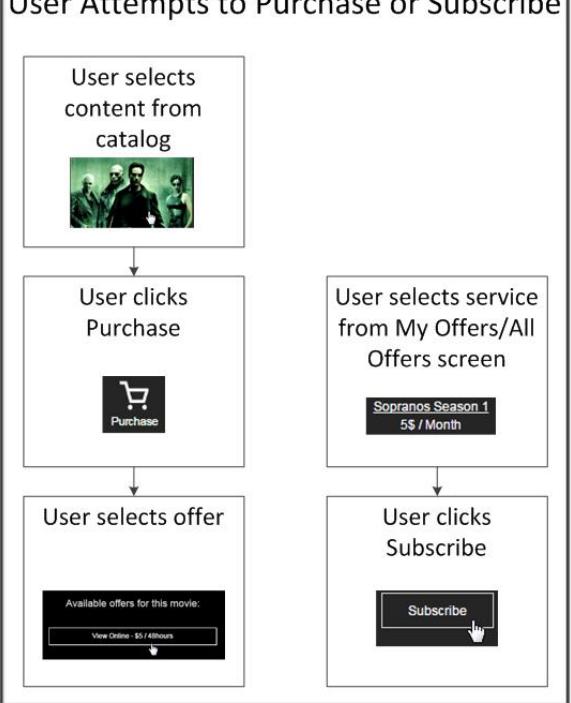
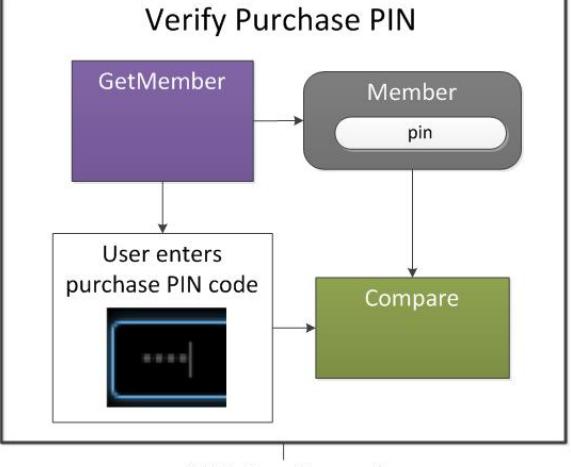
Steps	References	
	To	Refer To Tutorial
 <pre> graph TD A[Does Account have Payment Method? GetPaymentMethodDetailsInfo] -- No --> B[Begin Payment Method Registration Process] A -- Yes --> C[Next Step] </pre>	Determine whether the account has a payment method registered	126 Get an Online Account's Payment Method (p.331)
	Begin payment registration process	121 Register a New Payment Method via Adyen (p.320)
	Or	125 Register a New Payment Method via the Agnostic Payment Service (p.329)
Verify Purchase PIN		
	To	Refer To Tutorial
 <pre> graph TD A[GetMember] --> B[Member pin] B --> C[User enters purchase PIN code] C --> D[Compare] D --> E["PIN Confirmed"] </pre>	Get the user's purchase PIN from the back-end	110 Get Purchase PIN (p.292)

Steps	References														
<p>Perform Order/Subscription</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> User clicks image of card to confirm  </div> <div style="background-color: #800080; color: white; padding: 5px; margin-top: 10px;"> OrderVideo OrderVideoPackage AddServicePlan AddServiceSubscription AddPackageSubscription </div>	<table border="1" data-bbox="779 339 1430 788"> <thead> <tr> <th data-bbox="779 339 1176 440">To</th><th data-bbox="1176 339 1430 440">Refer To Tutorial</th></tr> </thead> <tbody> <tr> <td data-bbox="779 440 1176 653">Get last four digits and expiry to display on credit card image</td><td data-bbox="1176 440 1430 653">126 Get an Online Account's Payment Method (p.331)</td></tr> <tr> <td data-bbox="779 653 1176 837">Place an order for TVOD</td><td data-bbox="1176 653 1430 837">128 Ordering TVOD Content (p.336)</td></tr> <tr> <td data-bbox="779 837 1176 1028">Place an order for a Movie Package</td><td data-bbox="1176 837 1430 1028">129 Ordering a VOD Movie Package (p.341)</td></tr> <tr> <td data-bbox="779 1028 1176 1219">Subscribe to an SVOD Service</td><td data-bbox="1176 1028 1430 1219">130 Subscribing to an SVOD Service (p.345)</td></tr> <tr> <td data-bbox="779 1219 1176 1365">Subscribe to a Channel Package</td><td data-bbox="1176 1219 1430 1365">140 Subscribe to a Channel Package (p.368)</td></tr> <tr> <td data-bbox="779 1365 1176 1558">Subscribe to a Service Plan</td><td data-bbox="1176 1365 1430 1558">131 Subscribe to a Service Plan (p.348)</td></tr> </tbody> </table>	To	Refer To Tutorial	Get last four digits and expiry to display on credit card image	126 Get an Online Account's Payment Method (p.331)	Place an order for TVOD	128 Ordering TVOD Content (p.336)	Place an order for a Movie Package	129 Ordering a VOD Movie Package (p.341)	Subscribe to an SVOD Service	130 Subscribing to an SVOD Service (p.345)	Subscribe to a Channel Package	140 Subscribe to a Channel Package (p.368)	Subscribe to a Service Plan	131 Subscribe to a Service Plan (p.348)
To	Refer To Tutorial														
Get last four digits and expiry to display on credit card image	126 Get an Online Account's Payment Method (p.331)														
Place an order for TVOD	128 Ordering TVOD Content (p.336)														
Place an order for a Movie Package	129 Ordering a VOD Movie Package (p.341)														
Subscribe to an SVOD Service	130 Subscribing to an SVOD Service (p.345)														
Subscribe to a Channel Package	140 Subscribe to a Channel Package (p.368)														
Subscribe to a Service Plan	131 Subscribe to a Service Plan (p.348)														

119.3 Traditional Account Purchase Process

The process to be followed when a traditional account attempts to purchase content or subscribe to a service is outlined below.



Steps	References								
<p>User Attempts to Purchase or Subscribe</p>  <pre> graph TD A[User selects content from catalog] --> B[User clicks Purchase] B --> C[User selects offer] C --> D[User selects service from My Offers/All Offers screen] D --> E[User clicks Subscribe] </pre>	<table border="1"> <thead> <tr> <th data-bbox="798 332 1203 406">To</th><th data-bbox="1203 332 1432 406">Refer To Tutorial</th></tr> </thead> <tbody> <tr> <td data-bbox="798 406 1203 676">List offers for VOD Content</td><td data-bbox="1203 406 1432 676">35 List Compatible Devices (p.113)</td></tr> <tr> <td data-bbox="798 676 1203 900">List Channel Packages through which a channel can be purchased</td><td data-bbox="1203 676 1432 900">72 List Offers for a Channel (p.211)</td></tr> <tr> <td data-bbox="798 900 1203 1073">List all available subscription offers on the My Offers/All Offers screen</td><td data-bbox="1203 900 1432 1073">158 My Offers / All Offers Screen (p.429)</td></tr> </tbody> </table>	To	Refer To Tutorial	List offers for VOD Content	35 List Compatible Devices (p.113)	List Channel Packages through which a channel can be purchased	72 List Offers for a Channel (p.211)	List all available subscription offers on the My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
To	Refer To Tutorial								
List offers for VOD Content	35 List Compatible Devices (p.113)								
List Channel Packages through which a channel can be purchased	72 List Offers for a Channel (p.211)								
List all available subscription offers on the My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)								
<p>Verify Purchase PIN</p>  <pre> graph TD A[GetMember] --> B[Member pin] B --> C[User enters purchase PIN code] C --> D[Compare] D --> E["PIN Confirmed"] </pre>	<table border="1"> <thead> <tr> <th data-bbox="798 1230 1203 1304">To</th><th data-bbox="1203 1230 1432 1304">Refer To Tutorial</th></tr> </thead> <tbody> <tr> <td data-bbox="798 1304 1203 1484">Get the member's purchase PIN from the back-end</td><td data-bbox="1203 1304 1432 1484">110 Get Purchase PIN (p.292)</td></tr> </tbody> </table>	To	Refer To Tutorial	Get the member's purchase PIN from the back-end	110 Get Purchase PIN (p.292)				
To	Refer To Tutorial								
Get the member's purchase PIN from the back-end	110 Get Purchase PIN (p.292)								

Steps	References	
	To	Refer To Tutorial
Perform Order/Subscription <div style="background-color: #800080; color: white; padding: 5px; margin-top: 5px;"> OrderVideo OrderVideoPackage AddServicePlan AddServiceSubscription AddPackageSubscription </div>	Place an order for TVOD	128 Ordering TVOD Content (p.336)
	Place an order for a Movie Package	129 Ordering a VOD Movie Package (p.341)
	Subscribe to a Service	130 Subscribing to an SVOD Service (p.345)
	Subscribe to a Channel Package	140 Subscribe to a Channel Package (p.368)
	Subscribe to a Service Plan	131 Subscribe to a Service Plan (p.348)



120 Managing Payment Methods

120.1 Overview

An account must have a payment method defined to be able to order movies or services through a client application. Once a payment method has been defined, customers can order movies and services through the application. The Service Delivery Platform will handle the charges against the credit card company (for Online accounts) or through the operator's billing system (for Operator accounts).

This version supports Adyen as a payment partner. The Adyen payment service URL and the secretKey parameter should be configurable in the application.

Future versions will support the Agnostic Payment Service for registering payment methods with any payment partner the operator selects.

Each account can only have one payment method defined.

120.2 Related Tutorials

To	Refer to Tutorial
Create a My Account screen	96 Building a My Account Screen (p.264)
Determine whether an account has a payment method registered	126 Get an Online Account's Payment Method (p.331)
Get an online account's payment method details for display	126 Get an Online Account's Payment Method (p.331)
Register a new payment method via Adyen	121 Register a New Payment Method via Adyen (p.320)
Register a new payment method using the Agnostic Payment Service	125 Register a New Payment Method via the Agnostic Payment Service (p.329)



Delete a payment method	127 Delete Payment Method (p.334)
-------------------------	---

121 Register a New Payment Method via Adyen

121.1 Overview

The process of registering a payment method via Adyen consists of three steps:

1. Get the payment registration properties from the Service Delivery Platform.
2. Display the Partner's payment registration screen.
3. Validate and complete registration.

121.2 Related Tutorials

To	Refer to Tutorial
Learn about managing payment methods	120 Managing Payment Methods (p.318)
Get the payment registration properties from the Service Delivery Platform.	122 Get Payment Registration Properties (p.321)
Display the Partner's payment registration screen.	123 Register Payment Method (p.324)
Validate and complete registration.	124 Complete Registration (p.327)

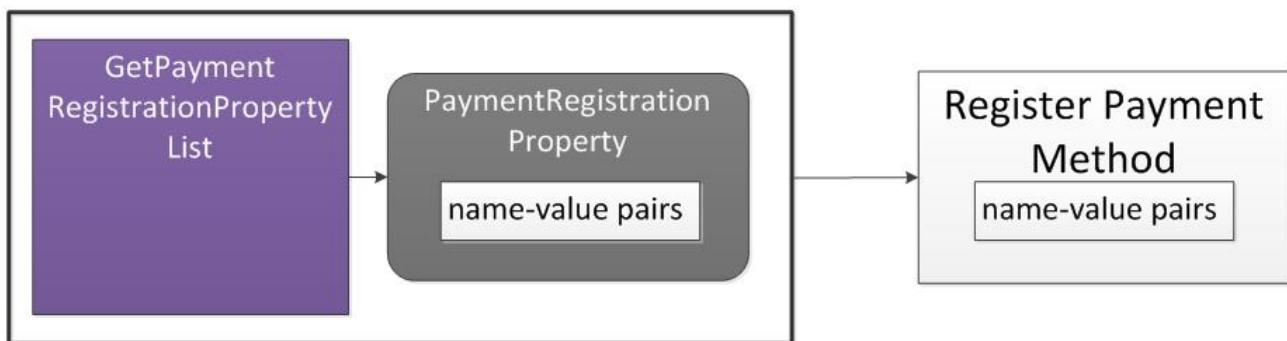
122 Get Payment Registration Properties

122.1 Overview

In this step, the client application retrieves relevant customer and system data from the SDP.

122.2 Requests and Responses

Get Payment Registration Properties



Request: [GetPaymentRegistrationPropertyList](#) (from RiGHTv Front-end API)

This request gets the required registration information to send to the payment service. No parameters are specified.

Response: [PaymentRegistrationProperty](#) (from RiGHTv Front-end API)

Returns the payment registration properties. A number of attributes such as currency, customer e-mail and more are returned. The following properties are returned for each attribute:

Property	Description
name	The payment property name.
value	The payment property value.

122.3 Related Tutorials

To	Refer to Tutorial
Register a new payment method via Adyen	121 Register a New Payment Method via Adyen (p.320)
Display the Partner's payment registration screen	123 Register Payment Method (p.324)
Validate and complete registration.	124 Complete Registration (p.327)

122.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetPaymentRegistrationPropertyList?client=json`

This request gets the payment registration details.

The response below shows the payment method details for the logged in user.

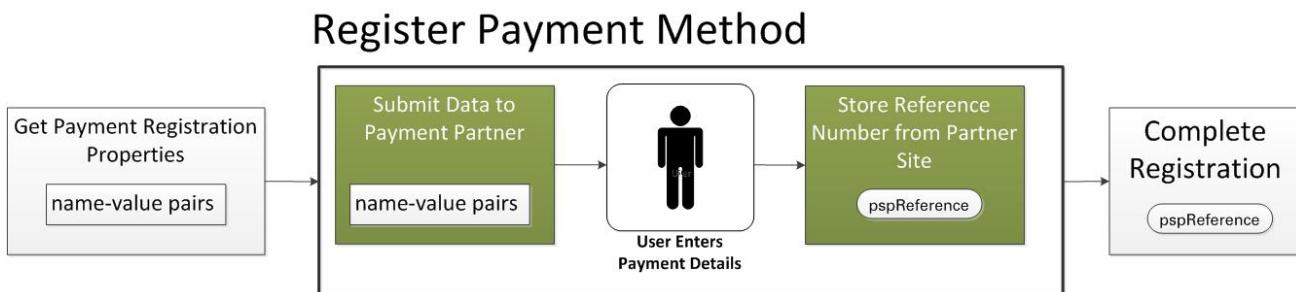
```
[{"metadata": {  
    "request": "GetPaymentRegistrationPropertyList",  
    "fullLength": 0,  
    "timestamp": 1454867885168  
},  
"response": [  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "paymentAmount",  
        "value": "2"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "currencyCode",  
        "value": "USD"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "shipBeforeDate",  
        "value": "2016-02-07"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "merchantReference",  
        "value": "REGISTRATION_14100"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "skinCode",  
        "value": "0bTvZVGQ"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "shopperEmail",  
        "value": "amir.zubary@viaccess-orca.com"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "sessionValidity",  
        "value": "2016-02-08T17:58:05Z"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "merchantAccount",  
        "value": "ViaccessCOM"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "shopperReference",  
        "value": "14100"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "recurringContract",  
        "value": "RECURRING"  
    },  
    {  
        "responseElementType": "PaymentRegistrationProperty",  
        "name": "merchantSig",  
        "value": "+7/EQBT04XMiyBn6KvPJ+kGufLU="  
    }  
]
```



123 Register Payment Method

123.1 Overview

In this step, the client application submits the data retrieved in the previous step to the payment partner as part of an HTTPS URL. The customer is then shown a web form to complete. After submitting the web form, the customer is shown a success screen. The client application must read the URL of that screen to get the transaction ID.



This process is explained through the example below.

123.2 Example

Use the response from *GetPaymentRegistrationPropertyList* to send the registration information to Ayden, as in the example below.

```
https://test.adyen.com/hpp/pay.shtml?paymentAmount=2&currencyCode=USD&shipBeforeDate=2016-03-03&merchantReference=REGISTRATION_14100&skinCode=0bTvZVGQ&shopperEmail=amir.zubary%40viaccess-orca.com&sessionValidity=2016-03-04T16%3A11%3A04Z&merchantAccount=ViaccessCOM&shopperReference=14100&recurringContract=RECURRING&secretKey=Kah942*%247sdp0%29&merchantSig=VPr6rlF3C152PUclXrWy3b6VMEE%3D
```

This URL directs the user to the payment partner's screen. The user is asked to register a payment method.

Please enter your payment method details

Credit Card

Card Number

Card Holder Name

Card Expiry Date

CVC/CVV/CID

What is CVC/CVV/CID?

Submit

Next Step: Enter your Payment Details

previous

Once the payment details have been submitted, a success screen is displayed.



The URL of the success screen includes a *pspReference* parameter. The client application must record this value.

```
https://test.adyen.com/hpp/result.shtml?merchantReference=REGISTRATION_
65300&skinCode=0bTvZVGQ&shopperLocale=en_
GB&paymentMethod=visa&authResult=AUTHORISED&pspReference=
8814605317852701&merchantSig=aw2mrAtJ35vCYboMWy0Gx%2Br9jdE%3D
```

123.3 Related Tutorials

To	Refer to Tutorial
Register a new payment method via Adyen	121 Register a New Payment Method via Adyen (p.320)
Get the payment registration properties from the Service Delivery Platform.	122 Get Payment Registration Properties (p.321)

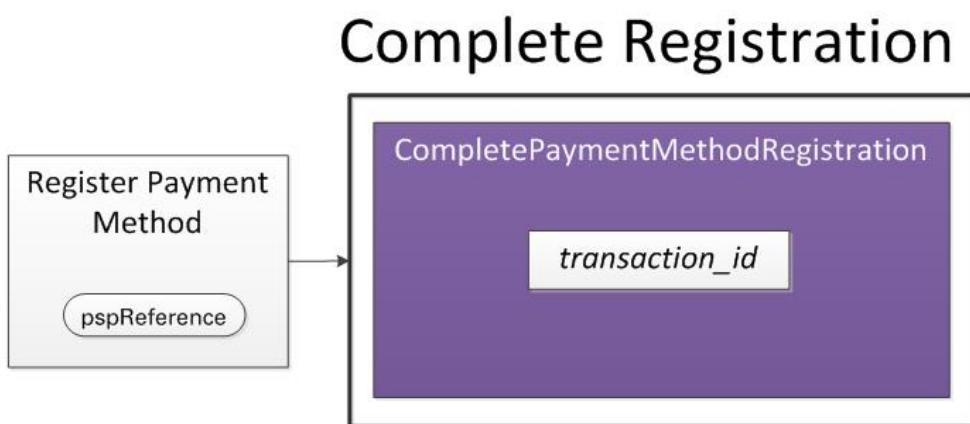
Validate and complete registration.	124 Complete Registration (p.327)
-------------------------------------	---

124 Complete Registration

124.1 Overview

In this step, the client application forwards the transaction ID (*pspReference*) received in the previous step to the Service Delivery Platform.

124.2 Requests and Responses



Request: [CompletePaymentMethodRegistration](#) (from RiGHTv Front-end API)

This request is used to validate and complete the user payment method registration. Only one parameter is required:

Property	Description
transaction_id	The transaction ID from Adyen's response (pspReference).

124.3 Related Tutorials

To	Refer To Tutorial
Register a new payment method via Adyen	121 Register a New Payment Method via Adyen (p.320)
Get the payment registration properties from the Service Delivery Platform.	122 Get Payment Registration Properties (p.321)
Display the Partner's payment registration screen.	123 Register Payment Method (p.324)

124.4 Example

https://<operator>.tvaas.com/RTEFacade/CompletePaymentMethodRegistration?transaction_id=8614548684156144&client=json

This request completes the payment method details for the logged in user.

The response below shows the successful validation of the payment method.

```
{  
  "metadata": {  
    "request": "CompletePaymentMethodRegistration",  
    "timestamp": 1459415073511  
  },  
  "response": {  
    "message": "'Validate Payment Registration' operation finished successfully",  
    "status": "SUCCESS"  
  }  
}
```

125 Register a New Payment Method via the Agnostic Payment Service

125.1 Overview

The Agnostic Payment Service is a VO microservice offered to simplify the process of registering payment methods.

The Agnostic Payment Service provides RiGHTv-facing client applications the ability to register payment methods (credit cards) with the payment company selected by the Operator, without having to be configured specifically for the selected payment company.

The Agnostic Payment Service should be called whenever an online account-holder attempts to make a purchase or requests to register a payment method. For a recommended purchase workflow, see [119 The Purchase Process \(p.312\)](#).

The Agnostic Payment Service displays a registration form customized for the payment partner the operator has selected, for example:

The screenshot shows a dark-themed user interface for adding a payment method. At the top, it says "Add a payment method". Below that are logos for VISA, MasterCard, American Express, and Discover, followed by the text "(and all others major cards)". The form contains several input fields: "Card Number", "Card Holder Name", "Card Expiry Date" (with a placeholder "mm / yyyy"), "CVC/CV/CID", and "Zip Code". Each input field has a question mark icon in its bottom right corner. At the bottom is a "Save" button.

125.2 Requests and Responses

The client application should call the Agnostic Payment Service whenever payment method registration is needed, as follows:

1. The client application must call the Agnostic Payment Service, by calling a URL of the following format:

`http://<server ip or host name: application port>/ PaymentRegistration`
For example:

`http://localhost:9090/PaymentRegistration`

The Agnostic Payment service then presents the user with a form to complete.

2. The client application should return to the screen from which the Agnostic Payment Service was called, for example:

- The screen showing the list of offers through which a VOD movie or live channel can be purchased.
- The My Offers/All Offers screen from which the user clicked Subscribe.

3. The user will click again to make the purchase or subscribe to the service. The purchase process will continue as normal for any online account holder, as described in [119 The Purchase Process \(p.312\)](#).

125.3 Related Tutorials

To	Refer to Tutorial
Learn about managing payment methods	120 Managing Payment Methods (p.318)
Create a My Account screen	96 Building a My Account Screen (p.264)
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)

126 Get an Online Account's Payment Method

126.1 Overview

This tutorial explains how to retrieve an online account's payment method. This is required in the following situations:

- The user should be able to view the details of the registered payment method. Only limited information may be displayed.
- When a user within an online account attempts to purchase content or subscribe to a service, the client application should determine whether the online account has a payment method registered or not.

126.2 Requests and Responses

Request: [GetPaymentMethodDetailsInfo](#) (from RiGHTv Front-end API)

This request gets the payment method details for the logged-in user. No parameters are specified.

Response: [PaymentMethodDetailsInfo](#) (from RiGHTv Front-end API)

Returns the payment information details. The entity includes the following attributes:

Property	Description
<i>ccExpiryMonth, ccExpiryYear</i>	The credit card expiry date.
<i>ccType</i>	The credit card type (e.g. Visa or Mastercard).
<i>ccLastFourDigits</i>	The last 4 digits of the credit card.
<i>ccHolderName</i>	The credit card holder first and last name.
<i>paymentSystemReferenceId</i>	The reference ID assigned to the payment method by the payment system.

If no payment method is registered for the account, an error will be returned as follows:

Property	Description
<i>code</i>	PAYMENT_METHOD_NOT_FOUND
<i>message</i>	Could not retrieve payment method for household <household ID> not found.
<i>status</i>	FAILURE

126.3 Related Tutorials

To	Refer To Tutorial
Learn about managing payment methods	120 Managing Payment Methods (p.318)
Create a My Account screen	96 Building a My Account Screen (p.264)
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)

126.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetPaymentMethodDetailsInfo?client=json>

This request gets the payment method details for the logged in user.

The response below shows the payment method for the logged in user.

```
{
  "metadata": {
    "request": "GetPaymentMethodDetailsInfo",
    "timestamp": 1459415948549
  },
  "response": {
    "ccExpiryMonth": 6,
    "ccType": "mc",
    "responseElementType": "PaymentMethodDetailsInfo",
    "ccExpiryYear": 2016,
    "ccLastFourDigits": "2909",
    "paymentSystemReferenceId": "8314594146183360",
    "ccHolderName": "Consumer"
  }
}
```



If the account did not have a payment method registered, the response would be as follows:

```
[{"metadata": {"request": "GetPaymentMethodDetailsInfo", "timestamp": 1490850752167}, "response": {"code": "PAYMENT_METHOD_NOT_FOUND", "message": "Could not retrieve payment method for household 35700 not found", "status": "FAILURE"}]
```



127 Delete Payment Method

127.1 Overview

The user must be able to delete a payment method, typically from a My Account screen. The user must be able to delete a payment method without having to enter a new one immediately. No purchases can be made while no payment method is defined.

127.2 Requests and Responses

Request: [DeletePaymentMethod](#) (from RiGHTv Front-end API)

This request deletes the payment method details for the logged-in user. A user can be left without any payment method. No parameters are specified.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

127.3 Related Tutorials

To	Refer To Tutorial
Learn about managing payment methods	120 Managing Payment Methods (p.318)
Create a My Account screen	96 Building a My Account Screen (p.264)

127.4 Example

<https://<operator>.tvaas.com/RTEFacade/DeletePaymentMethod?client=json>

This request deletes the payment method details for the logged in user.

The response below shows a successful deletion of the user's payment method.



```
{  
  "metadata": {  
    "request": "DeletePaymentMethod",  
    "timestamp": 1459416161193  
  },  
  "response": {  
    "message": "Delete Payment Method operation finished successfully",  
    "status": "SUCCESS"  
  }  
}
```

128 Ordering TVOD Content

128.1 Overview

To buy the right to view content, a user has to purchase an offer that includes the content they want.

This tutorial explains how to order TVOD movies.

The client application is responsible for verifying the user's purchase PIN before submitting an ordering request.

The back end handles the transaction through the credit card payment system on its own, without client application involvement.

Only logged-in users can make purchases. If an anonymous user presses a Purchase/Subscribe button, the log-in screen should be displayed.

The ordering process is illustrated below.

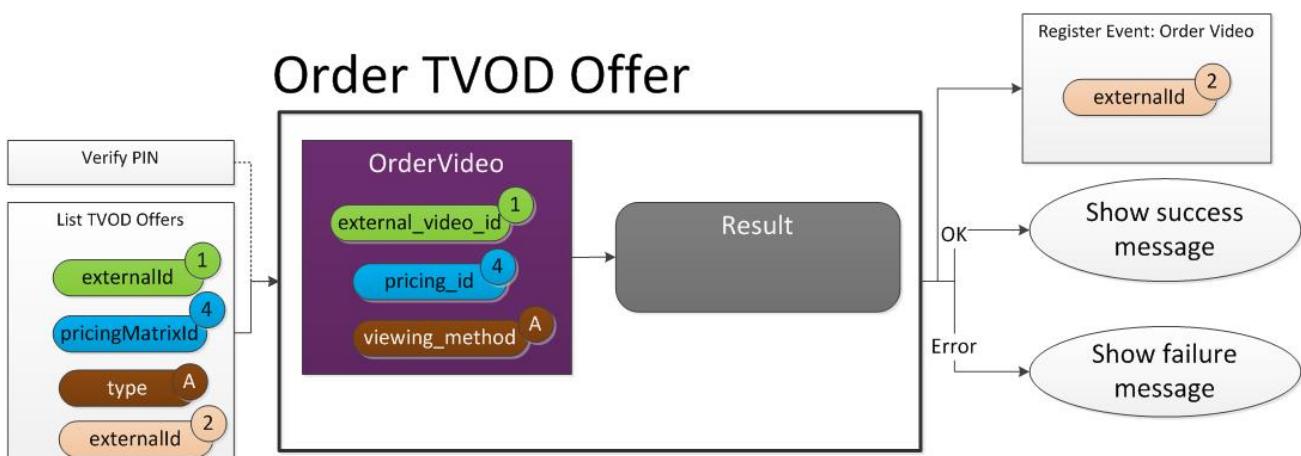


1. The user selects a movie or movie package to order.
2. The application requests the user to enter their purchase PIN, and then verifies that against the PIN stored in the Service Delivery Platform database.
3. Once the PIN is successfully verified, the client application sends the order to the Service Delivery Platform.

For a more detailed purchase process (including verification of a registered payment method for online accounts), see [119 The Purchase Process \(p.312\)](#).

Note on Analytics: The Analytics Platform extracts data sent when ordering TVOD content for use in many of its charts and KPIs.

128.2 Requests and Responses



Request: [OrderVideo](#) (from RiGHTv Front-end API)

This request orders the specified video for the account. Significant parameters include:

Parameter	Description
<i>external_video_id</i>	The external ID of the VOD content in the catalog.
<i>pricing_id</i>	The <i>ID of the pricing matrix of the offer selected</i> .
<i>viewing_method</i>	The viewing method that matches the pricing option.
For In-App Purchasing Only:	
<i>store_id</i>	The ID of the app store through which the video is purchased.
<i>purchase_reference</i>	The reference to a purchase made for the specified video at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the video purchased at a particular app store. Relevant for the Google App store only.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

Note

For additional guidelines relevant for ordering VOD offers for download, see [Ordering Content for Download](#).



128.3 Related Tutorials

To	Refer to Tutorial
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)
List available TVOD offers for a movie	37 List TVOD Offers (p.117)
Register an Order Video event	196 Register Event: Order Video (p.529)
Get the user's purchase PIN for verification	110 Get Purchase PIN (p.292)
Learn about ordering VOD offers for download	Ordering Content for Download

128.4 Example

As explained in [37 List TVOD Offers \(p.117\)](#), the *GetAggregatedVideo* request gets a list of TVOD movie offers and SVOD subscriptions that include a selected movie. A TVOD movie offer can be identified in the returned *AggregatedVideo* response by the false value in the *isSecured* property.

The details on a TVOD offer for the movie Cinderella is shown below. This list is obtained using the request below.

```
https://<operator>.tvaas.com/RTEFacade/GetAggregatedVideo?external_asset_id=Cinderella&language_code=es&client=json
```



```
{
  "isSecured": false, ←
  "attachments": [],
  "encodings": [
    {
      "responseElementType": "Encoding",
      "name": "HLS"
    }
  ],
  "externalId": "Cinderella-test", ←
  "responseElementType": "UniqueVideoInfo",
  "simultaneousViewsLimit": 1,
  "removalDate": "",
  "extrafields": [
    {
      "responseElementType": "Extrafield",
      "name": "search",
      "value": "1"
    }
  ],
  "pricingMatrixId": 4024, ←
  "definition": "",
  "securityGroups": [],
  "discountId": "",
  "status": 2
}
```

To obtain the list of viewing methods available in the pricing matrix, use the *GetPricingMatrix* request.

https://<operator>.tvaas.com/RTEFacade/GetPricingMatrix?pricing_matrix_id=4024&client=json

This request obtains details on pricing matrix 4024. The response shows that only a streaming option is available, as specified in the *type* property. The internal *id* of the pricing matrix is shown to be 4025.

```
RESPONSE = L
{
  "responseElementType": "PricingOption",
  "allowedViews": "",
  "downloadPeriod": "",
  "price": 5, ←
  "rentalPeriod": 48,
  "downloadPeriodUnit": "",
  "name": "Streaming",
  "id": 4025, ←
  "type": "streaming", ←
  "rentalPeriodUnit": 3
}
```

To order the movie, use the OrderVideo request.

https://<operator>.tvaas.com/RTEFacade/OrderVideo?external_video_id=Cinderella&pricing_id=4025&viewing_method=streaming&client=json



This request orders the movie `Cinderella` with the pricing ID `4025` and the viewing method `streaming`.

The response below indicates successful ordering of the video.

```
[{"metadata": {  
    "request": "OrderVideo",  
    "timestamp": 1460277106544  
},  
 "response": {  
    "message": "Order VOD operation finished successfully",  
    "status": "SUCCESS"  
}}
```



129 Ordering a VOD Movie Package

129.1 Overview

To buy the right to view content, a user has to purchase an offer that includes the content they want.

This tutorial explains how to order VOD movie packages.

The client application is responsible for verifying the user's purchase PIN before submitting an ordering request.

The back end handles the transaction through the credit card payment system on its own, without client application involvement.

Only logged-in users can make purchases. If an anonymous user presses a Purchase/Subscribe button, the log-in screen should be displayed.

The ordering process is illustrated below.



1. The user selects a movie package to order.
2. The application requests the user to enter their purchase PIN, and then verifies that against the PIN stored in the Service Delivery Platform database.
3. Once the PIN is successfully verified, the client application sends the order to the Service Delivery Platform.

For a more detailed purchase process (including verification of a registered payment method for online accounts), see [119 The Purchase Process \(p.312\)](#).

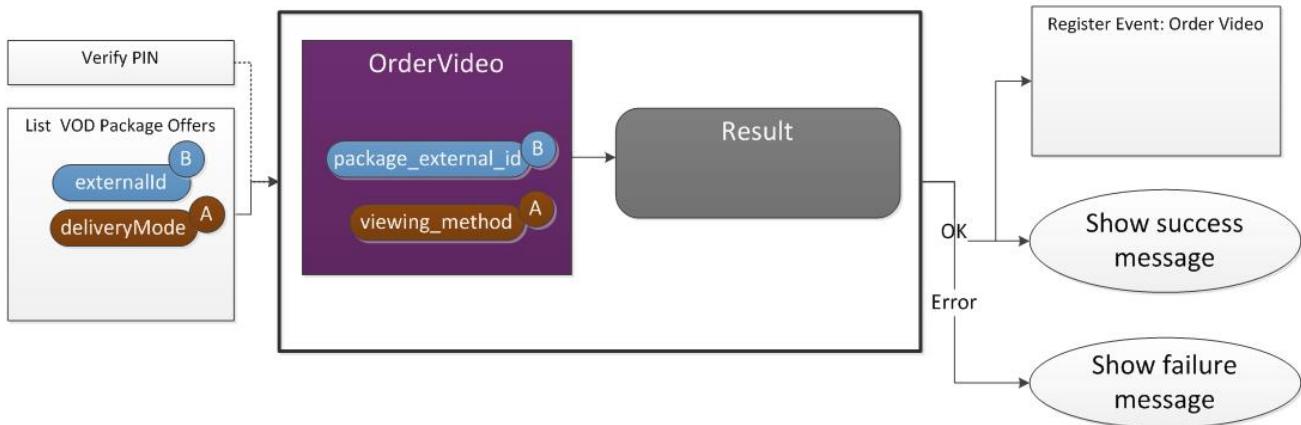
Note on Analytics: The Analytics Platform extracts data sent when ordering VOD Movie Packages for use in many of its charts and KPIs.

Note

After ordering a VOD Movie Package, the ticket for each movie in the package must be activated. See [52 Activate VOD Package Movie Ticket \(p.157\)](#).

129.2 Requests and Responses

Order Movie Package



Request: [OrderVideoPackage](#) (from RiGHTv Front-end API)

This request orders the specified VOD Movie Package.

Parameter	Description
<i>package_external_id</i>	The external ID of the VOD movie package in the catalog.
<i>viewing_method</i>	Either streaming, download or converged.
For In-App Purchasing Only:	
<i>store_id</i>	The ID of the app store through which the movie package is purchased.
<i>purchase_reference</i>	The reference to a purchase made for the specified movie package at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the movie package purchased at a particular app store. Relevant for the Google App store only.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:



Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

Note

For additional guidelines relevant for ordering VOD offers for download, see [Ordering Content for Download](#).

129.3 Related Tutorials

To	Refer to Tutorial
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)
Activate the tickets for movies in the VOD Movie Package.	52 Activate VOD Package Movie Ticket (p.157)
List available VOD Movie Package offers for a movie	40 List VOD Movie Package Offers (p.129)
List available SVOD offers for a movie	38 List SVOD Offers for VOD Movie (p.122)
List available Service Plan offers for a movie	39 List Service Plan Offers for a VOD Movie (p.125)
Register an Order Video event	196 Register Event: Order Video (p.529)
Get the user's purchase PIN for verification	110 Get Purchase PIN (p.292)



Register the order of a VOD Movie Package for analytics purposes	205.3.2 Ordering a VOD Package (p.552)
Learn about ordering VOD offers for download	Ordering Content for Download

129.4 Example

`https://<operator>.tvaas.com/RTEFacade/OrderVideoPackage?package_external_id=Favorites&viewing_method=streaming&client=json`

This request orders the VOD package with external id=Favorite and the viewing method streaming.

The response below indicates successful ordering of the video.

```
[{"metadata": {  
    "request": "OrderVideoPackage",  
    "timestamp": 1460280196673  
},  
 "response": {  
    "message": "Order-vod-package operation finished successfully",  
    "status": "SUCCESS"  
}}]
```



130 Subscribing to an SVOD Service

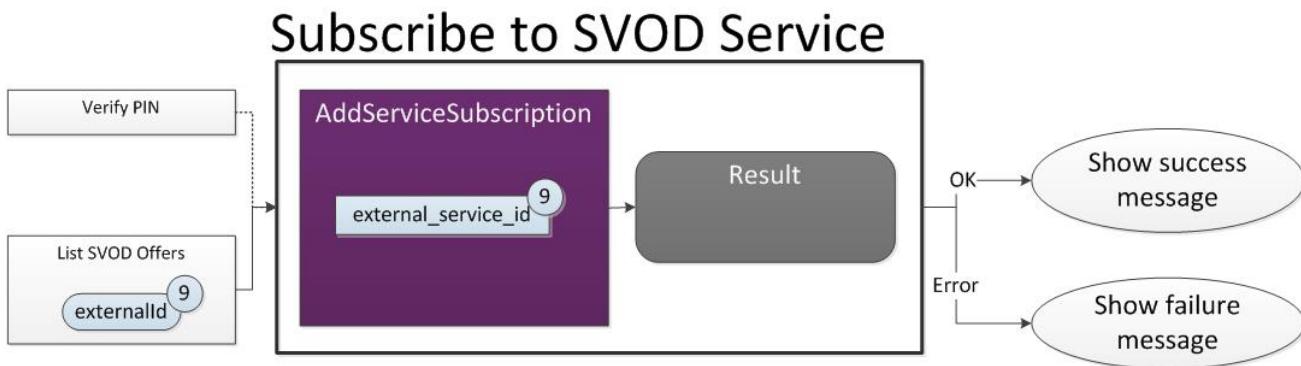
130.1 Overview

Customers can subscribe to an SVOD service via the PC portal. The customer is billed periodically through the online payment system. If the recurring payment fails, the online payment system will notify the Service Delivery Platform and the customer is removed from the service.

SVOD subscription can be performed from PC apps only.

For a detailed purchase process (including verification of a registered payment method for online accounts), see [119 The Purchase Process \(p.312\)](#).

130.2 Requests and Responses



Request: [AddServiceSubscription](#) (from RiGHTv Front-end API)

This request subscribes the account to the specified SVOD service. The following parameters are required:

Parameter	Description
<code>external_service_id</code>	The SVOD external ID.
For In-App Purchasing Only:	
<code>store_id</code>	The ID of the app store through which the SVOD service is being subscribed.

Parameter	Description
<i>purchase_reference</i>	The reference to a purchase made for the specified SVOD service at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the SVOD service purchased at a particular app store. Relevant for the Google App store only.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .
<i>message</i>	A short message from the server

Note

For additional guidelines relevant for ordering VOD offers for download, see [Ordering Content for Download](#).



130.3 Related Tutorials

To	Refer To Tutorial
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
List available SVOD offers for a movie	38 List SVOD Offers for VOD Movie (p.122)
Get the user's purchase PIN for verification	110 Get Purchase PIN (p.292)
Order a movie from within a subscribed SVOD offer	135 Opening a Ticket for an SVOD Movie (p.357)
List SVOD offers in the My Offers/All Offers screen	159 List All Available SVOD Offers (p.431)
Learn about ordering VOD offers for download	Ordering Content for Download

130.4 Example

`https://<operator>.tvaas.com/RTEFacade/AddServiceSubscription?external_service_id=Disney&client=json`

This request subscribes the account to the Disney SVOD service.

The response below indicates successful subscription to the SVOD service.

```
[{"metadata": {
    "request": "AddServiceSubscription",
    "timestamp": 1458210578295
},
"response": {
    "message": "'Add Service Subscription' operation finished successfully",
    "status": "SUCCESS"
}]
```



131 Subscribe to a Service Plan

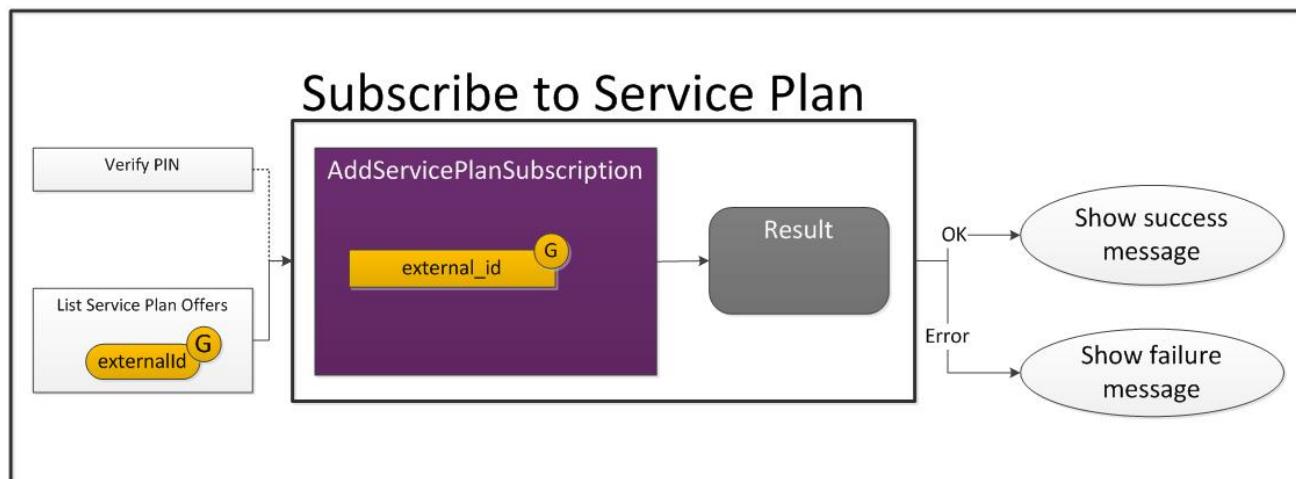
131.1 Overview

Customers can subscribe to a Service Plan via the PC portal.

The customer is billed periodically through the online payment system. If the recurring payment fails, the online payment system will notify the Service Delivery Platform and the customer is removed from the Service Plan.

For a detailed purchase process (including verification of a registered payment method for online accounts), see [119 The Purchase Process \(p.312\)](#).

131.2 Requests and Responses



Request: [AddServicePlanSubscription](#) (from RiGHTv Front-end API)

This request subscribes the account to the specified Service Plan. The following parameters are required:

Parameter	Description
<i>external_id</i>	The Service Plan external ID.
For In-App Purchasing Only:	
<i>store_id</i>	The ID of the app store through which the service plan SVOD service is being subscribed.

Parameter	Description
<i>purchase_reference</i>	The reference to a purchase made for the specified service plan SVOD service at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the service plan SVOD service purchased at a particular app store. Relevant for the Google App store only.

Response: Result (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

131.3 Related Tutorials

To	Refer To Tutorial
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)
List available Service Plan offers for a movie	39 List Service Plan Offers for a VOD Movie (p.125)
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Get the user's purchase PIN for verification	110 Get Purchase PIN (p.292)



To	Refer To Tutorial
List all Service Plans in a My Offers/All Offers screen	161 List All Available Service Plan Offers (p.444)

131.4 Example

https://<operator>.tvaas.com/RTEFacade/AddServicePlanSubscription?external_id=Childrens_Place&client=json

This request subscribes the account to the Children's place Service Plan.

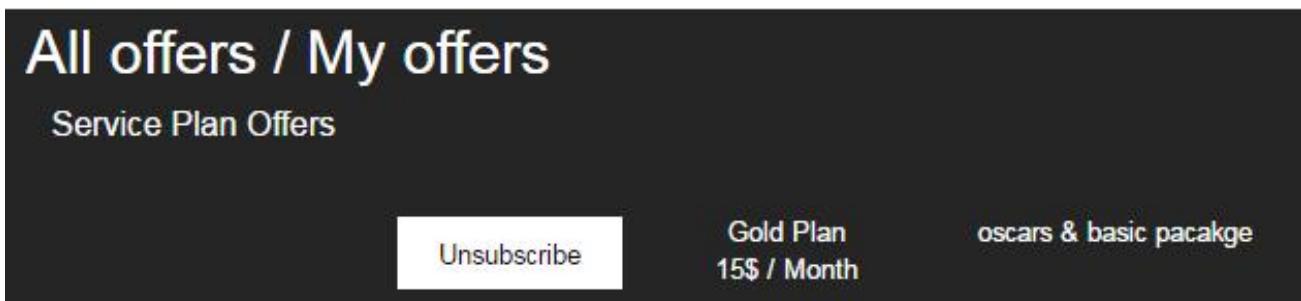
The response below indicates successful subscription to the Service Plan.

```
[{"metadata": { "request": "AddServicePlanSubscription", "timestamp": 1488106482075 }, "response": { "message": "'Add Service Plan Subscription' operation finished successfully", "status": "SUCCESS" }}
```



132 Unsubscribing from a Service Plan

A user should be able to unsubscribe from a Service Plan, typically from a My Offers/All Offers screen.



Two APIs exist for unsubscribing accounts from a Service Plan:

- [133 Finish Service Plan Subscription \(p.353\)](#)
For Online accounts. The customer will continue to have access to the Service Plan until the next planned billing date, at which time the Service Plan will terminate and no charge will be levied.
- [134 Remove Service Plan Subscription \(p.355\)](#)
For operator accounts. The customer loses access to the Service Plan immediately, and no further charges are levied. Any prepayment is forfeited.

For identifying the accounts type, see [118 Identify Account Type \(p.308\)](#).

132.1 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Terminate a Service Plan subscription for an online account	133 Finish Service Plan Subscription (p.353)
Terminate a Service Plan subscription for a traditional account	134 Remove Service Plan Subscription (p.355)

To	Refer To Tutorial
Identify whether an account is an online or traditional account	118 Identify Account Type (p.308).
List all Service Plans in a My Offers/All Offers screen	161 List All Available Service Plan Offers (p.444)

133 Finish Service Plan Subscription

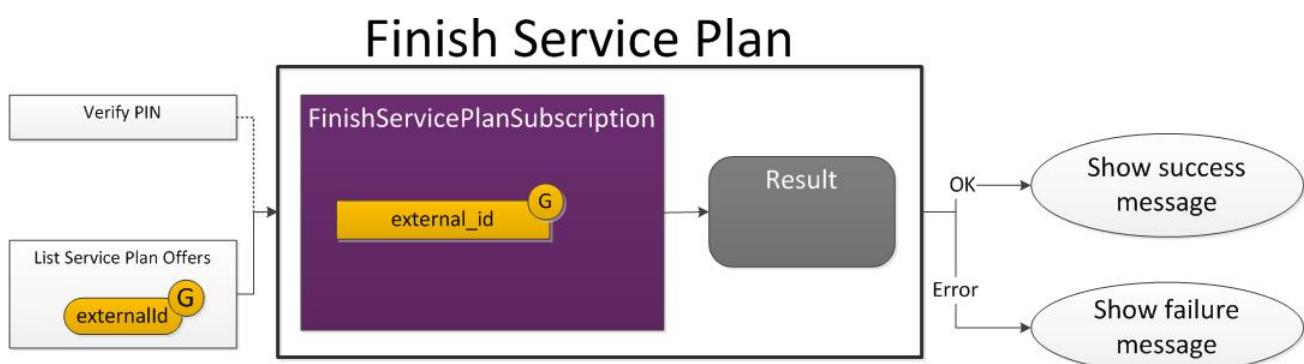
133.1 Overview

Online accounts can unsubscribe from a Service Plan. Unsubscribing using *FinishServicePlanSubscription* sets the end date of the Service Plan to the next planned debit date. The account will have access to the Service Plan until that date.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

133.2 Requests and Responses



Request: [FinishServicePlanSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified Service Plan. The following parameters are required:

Parameter	Description
<i>external_id</i>	The Service Plan external ID.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

133.3 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Unsubscribe from a Service Plan	132 Unsubscribing from a Service Plan (p.351)

133.4 Example

`https://<operator>.tvaas.com/RTEFacade/FinieshServicePlanSubscription?external_service_id=Gold_Plan&client=json`

This request ends the account's subscription to the Gold Plan Service Plan.

The response below indicates successful ending of subscription to the Service Plan.

```
{
  "metadata": {
    "request": "FinishServicePlanSubscription",
    "timestamp": 1488107618409
  },
  "response": {
    "message": "'Finish Service Plan Subscription' operation finished successfully",
    "status": "SUCCESS"
  }
}
```

134 Remove Service Plan Subscription

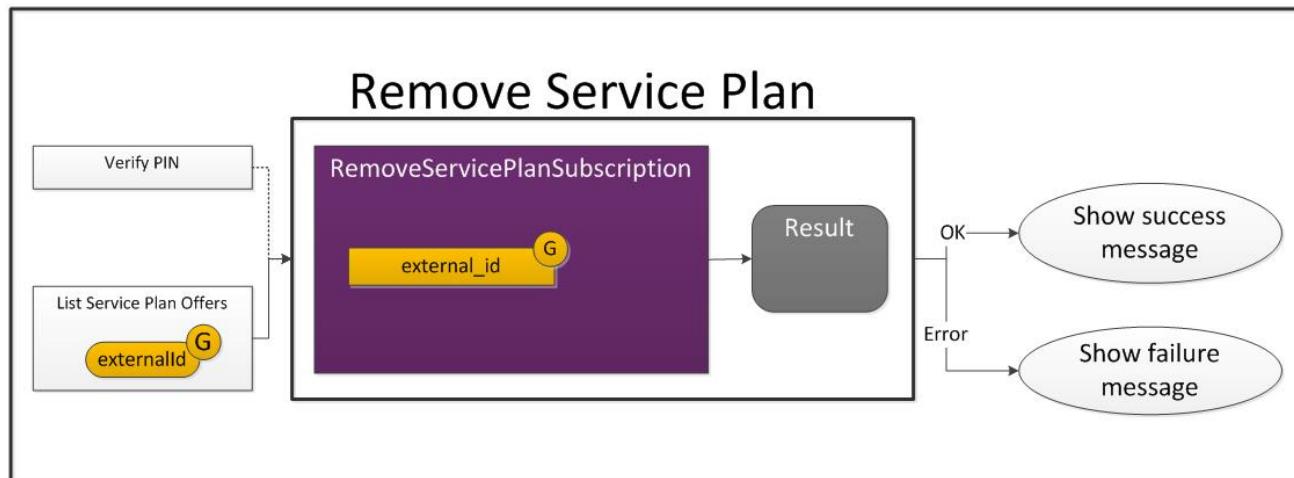
134.1 Overview

Removing Service Plans is intended for use with operator accounts. Unsubscribing using *RemoveServicePlanSubscription* ends the Service Plan immediately.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

134.2 Requests and Responses



Request: [RemoveServicePlanSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified Service Plan. The following parameters are required:

Parameter	Description
<i>external_id</i>	The Service Plan external ID

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

134.3 Related Tutorials

To	Refer to Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Unsubscribe from a Service Plan	132 Unsubscribing from a Service Plan (p.351)

134.4 Example

https://<operator>.tvaas.com/RTEFacade/RemoveServicePlanSubscription?external_id=Gold_Plan001&client=json

This request ends the account's subscription to the Gold Plan Service Plan.

The response below indicates successful ending of subscription to the Service Plan.

```

{
  "metadata": {
    "request": "RemoveServicePlanSubscription",
    "timestamp": 1488111294907
  },
  "response": {
    "message": "'Remove Service Plan Subscription' operation finished successfully",
    "status": "SUCCESS"
  }
}

```



135 Opening a Ticket for an SVOD Movie

135.1 Overview

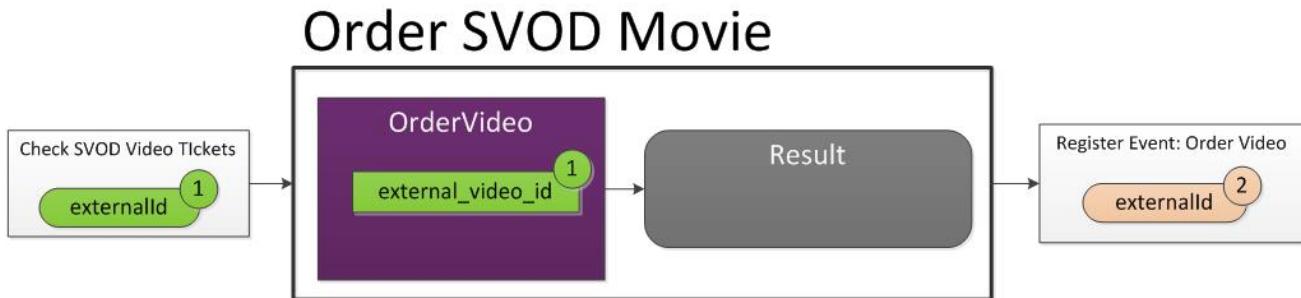
Once an account has been subscribed to an SVOD service, the client application has to order each video from the service before it can be viewed.

The video can be ordered, for example, when the user attempts to play the movie.

Note for Analytics: For data collection purposes (for use in the Analytics Platform, see [205 Apps and Analytics \(p.550\)](#)), it is important that each SVOD video be ordered only once. If a ticket for the video already exists, the video should not be ordered again (see [136 Check SVOD Video Tickets \(p.359\)](#)).

Note: The procedure shown here applies whether the customer subscribed to the SVOD service directly, or through a Service Plan.

135.2 Requests and Responses



Request: [OrderVideo](#) (from RiGHTv Front-end API)

This request orders the specified video.

Parameter	Description
<i>external_video_id</i>	The external ID of the VOD content in the catalog.
<i>viewing_method</i>	Set to <code>streaming</code> .
<i>subscription</i>	Set to <code>true</code> , to indicate that the video is part of one of the account's subscriptions.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

Note

For additional guidelines relevant for ordering VOD offers for download, see [Ordering Content for Download](#).

135.3 Related Tutorials

To	Refer To Tutorial
Play a VOD movie	45 Playing a Movie (p.141)
Check if a ticket has already been opened for an SVOD movie, and to get the external ID of the movie page	136 Check SVOD Video Tickets (p.359)
Learn about ordering VOD offers for download	Ordering Content for Download

135.4 Example

This request orders the SVOD movie Avatar and the viewing method streaming.

`https://<operator>.tvaas.com/RTEFacade/OrderVideo?external_video_id=Avatar&subscription=true&viewing_method=streaming&client=json`

The response below indicates successful ordering of the video.

```
{
  "metadata": {
    "request": "OrderVideo",
    "timestamp": 1459921188495
  },
  "response": {
    "message": "Order VOD operation finished successfully",
    "status": "SUCCESS"
  }
}
```



136 Check SVOD Video Tickets

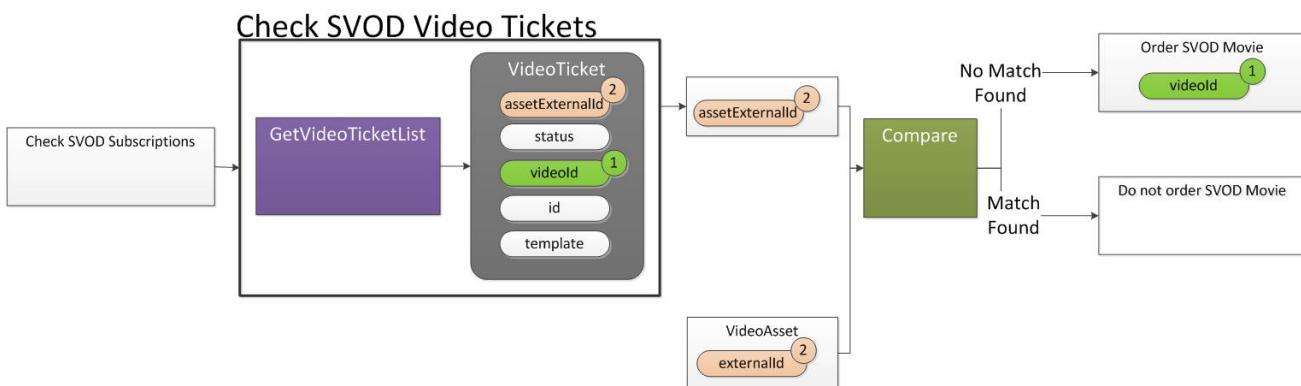
136.1 Overview

Once an SVOD subscription has been ordered, a ticket for each individual video included in the service must be ordered individually before the video can be played. Because the videos in the SVOD service can change from time to time, each video should be ordered when the customer first attempts to play the video. (See [135 Opening a Ticket for an SVOD Movie \(p.357\)](#) for instructions on how to order SVOD movies.)

Because the action used to order videos, *OrderVideo*, is monitored for data collection for the Analytics Platform, it is important that *OrderVideo* be called only once per SVOD movie.

The application should check whether a ticket for the video has already been opened. If a ticket has been opened, the video should not be ordered again.

136.2 Requests and Responses



GetVideoTicketList should be called to get a list of all a customer's active tickets. The list of assets for which tickets exist should be compared to the asset that the customer wants to view. If a match is found, that movie should not be ordered again, but if no match is found, the movie should be ordered.

Request: [GetVideoTicketList](#) (from RiGHTv Front-end API)

Returns a list of the account's video tickets. No parameters are required.

Request: [VideoTicket](#) (from RiGHTv Front-end API)

This entity lists movies and movie packages purchased by the account. It includes the following properties.



Property	Description
<i>template</i>	Indicates the template (business model) through which the video was purchased. Displays svod service if purchased through an SVOD Service.
<i>assetExternalId</i>	The external ID of the asset in the Content Library.
<i>status</i>	Indicates if active or expired .
<i>videoId</i>	The external ID of the movie page in the VOD Catalog.
<i>id</i>	The ID of the ticket opened for the movie.

Check the external ID of the movie to be played against the values of *assetExternalId* in the listed SVOD movie tickets.

If a match is found, a ticket already exists and the video must not be ordered again.

136.3 Related Tutorials

To	Refer To Tutorial
Open a ticket for a movie ordered through an SVOD offer	135 Opening a Ticket for an SVOD Movie (p.357)
Check to see whether the account subscribes to an SVOD service that includes the movie	48 Check SVOD Subscriptions (p.148)

136.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetVideoTicketList?client=json`

This call requests all video tickets opened for the logged-in account.

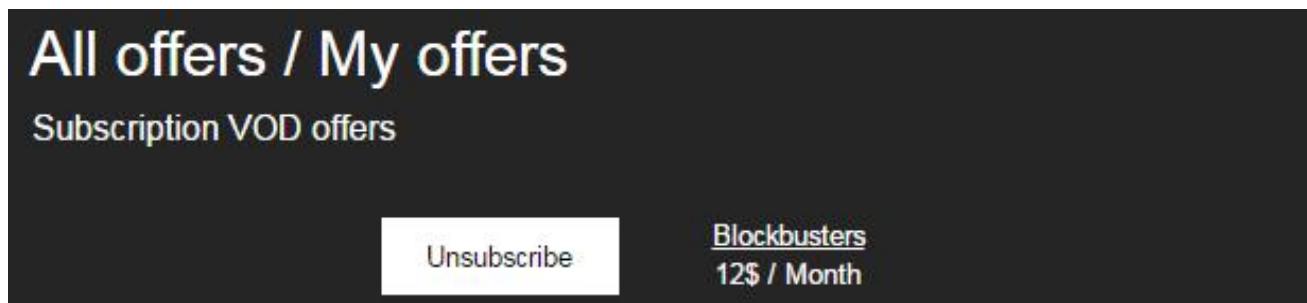
The sample response below shows that the account has an active ticket to movie The Avengers, ordered through an SVOD service.



```
"template": "svod_service",  
"advisories": "",  
"prLevel": 0,  
"allowedEstDownloadingDevices": "",  
"videoId": 6502,  
"type": "subscription",  
"assetExternalId": "The Avengers",  
"responseElementType": "VideoTicket",  
"attachment": {},  
"downloadPeriod": "",  
"deliveryMode": "streaming",  
"assetId": 412063,  
"id": 875052,  
"windowEnd": 1482830113941,  
"videoPackageName": "",  
"rentalPeriodUnit": 3,  
"allowedViews": "",  
"windowStart": 1482822913941,  
"videoPackageExternalId": "",  
"rentalPeriod": 2,  
"prName": "NR",  
"videoExternalId": "The Avengers",  
"downloadWindowEnd": "",  
"downloadWindowStart": "",  
"downloadPeriodUnit": 0,  
"name": "subtitles_program_2",  
"currentEstDownloadingDevices": "",  
"status": "active"
```

137 Unsubscribing from an SVOD Service

A user should be able to unsubscribe from an SVOD service, typically from a My Offers/All Offers screen.



Two APIs exist for unsubscribing users from an SVOD service:

- [138 Finish Service Subscription \(p.364\)](#)
Available only for Online accounts. The customer will continue to have access to the service until the next planned billing date, at which time the service will terminate and no charge will be levied.
- [139 Remove Service Subscription \(p.366\)](#)
Available for online and operator accounts. The customer loses access to the service immediately, and no further charges are levied. Any prepayment is forfeited.

For identifying the accounts type, see [118 Identify Account Type \(p.308\)](#).

137.1 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Terminate an SVOD subscription for an online account	138 Finish Service Subscription (p.364)
Terminate an SVOD subscription for a traditional account	139 Remove Service Subscription (p.366)

To	Refer To Tutorial
Identify whether an account is an online or traditional account	118 Identify Account Type (p.308).
List SVOD offers in the My Offers/All Offers screen	159 List All Available SVOD Offers (p.431)

138 Finish Service Subscription

138.1 Overview

Online accounts can unsubscribe from an SVOD service. Unsubscribing using *FinishServiceSubscription* sets the end date of the subscription to the next planned debit date. The account will have access to the service until that date.

Note

Subscription termination for services subscribed to through a mobile app store is described in .

138.2 Requests and Responses

Request: [FinishServiceSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified SVOD service. The following parameters are required:

Parameter	Description
<i>external_service_id</i>	The SVOD external ID

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

138.3 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	
Unsubscribe from an SVOD service	

138.4 Example

`https://<operator>.tvaas.com/RTEFacade/
FinishServiceSubscription?external_service_id=Disney&client=json`

This request ends the account's subscription to the Disney SVOD service.

The response below indicates successful ending of subscription to the SVOD service.

```
{  
  "metadata": {  
    "request": "FinishServiceSubscription",  
    "timestamp": 1459417187726  
  },  
  "response": {  
    "message": "'Finish Service Subscription' operation finished successfully",  
    "status": "SUCCESS"  
  }  
}
```

139 Remove Service Subscription

139.1 Overview

Online and operator accounts can unsubscribe from an SVOD service. Unsubscribing using *RemoveServiceSubscription* ends the service immediately.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

139.2 Requests and Responses

Request: [RemoveServiceSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified SVOD service. The following parameters are required:

Parameter	Description
<i>external_service_id</i>	The SVOD external ID

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

139.3 Related Tutorials

To	Refer to Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Unsubscribe from an SVOD service	137 Unsubscribing from an SVOD Service (p.362)



139.4 Example

https://<operator>.tvaas.com/RTEFacade/RemoveServiceSubscription?external_service_id=Disney&client=json

This request ends the account's subscription to the Disney SVOD service.

The response below indicates successful ending of subscription to the SVOD service.

```
[{"metadata": {  
    "request": "RemoveServiceSubscription",  
    "timestamp": 1460460819520  
},  
 "response": {  
    "message": "'Remove Service Subscription' operation finished successfully",  
    "status": "SUCCESS"  
}}]
```

140 Subscribe to a Channel Package

140.1 Overview

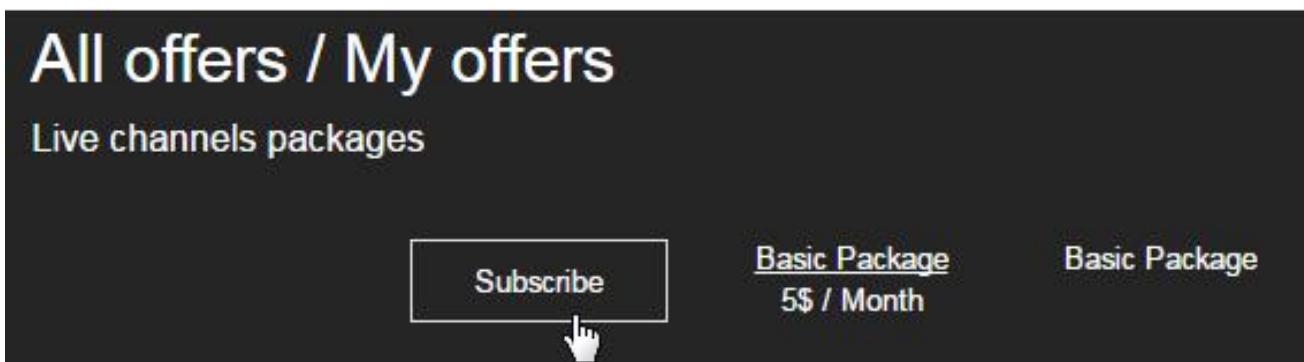
A customer can subscribe to a channel package service via the PC portal. The customer is billed periodically through the online payment system.

If the recurring payment fails, the online payment system will notify RiGHTv and the customer is removed from the service.

Channel subscription can be done from PC apps only.

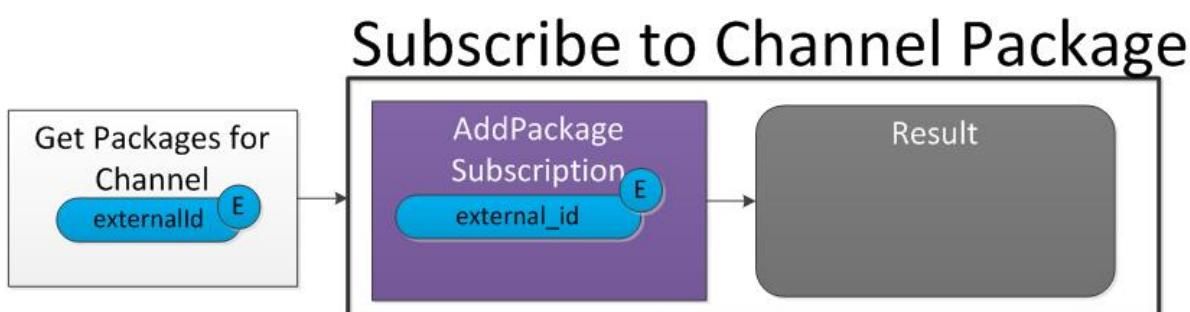
Customers can also unsubscribe from a channel package service, but only from a PC app.

Users subscribe to a Channel Package either from a My Offers/All Offers screen, or after being presented with a list of offers for purchasing access to a particular channel.



For a detailed purchase process (including verification of a registered payment method for online accounts), see [119 The Purchase Process \(p.312\)](#).

140.2 Requests and Responses



Request: [AddpackageSubscription](#) (from RiGHTv Front-end API)

This request subscribes the account to the specified channel package. The following parameters are required:

Parameter	Description
<i>external_id</i>	The external ID of the channel package
For In-App Purchasing Only:	
<i>store_id</i>	The ID of the app store through which the channel packageSVOD service is being subscribed.
<i>purchase_reference</i>	The reference to a purchase made for the specified channel packageSVOD service at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the channel packageSVOD service purchased at a particular app store. Relevant for the Google App store only.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server



140.3 Related Tutorials

To	Refer To Tutorial
View a recommended purchase/subscribe workflow	119 The Purchase Process (p.312)
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
List Channel Packages through which access to a channel can be purchased	72 List Offers for a Channel (p.211)
List Channel Packages in a My Offers/All Offers screen	160 List Channel Package Offers (p.438)

140.4 Example

https://<operator>.tvaas.com/RTEFacade/AddPackageSubscription?external_id=Trial%20Package%202&client=json

This request subscribes the account to the channel package with external ID Trial Package 2.

The response below indicates successful subscription to the channel package.

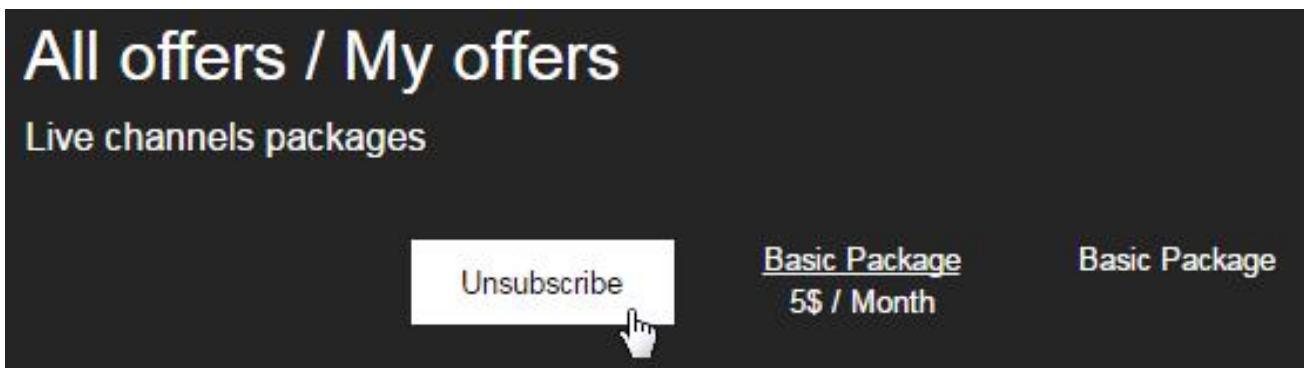
```

{
  "metadata": {
    "request": "AddPackageSubscription",
    "timestamp": 1460461967144
  },
  "response": {
    "message": "'Add Package Subscription' operation finished successfully",
    "status": "SUCCESS"
  }
}
  
```

141 Unsubscribe from a Channel Package

141.1 Overview

A user should be able to unsubscribe from an SVOD service, typically from a My Offers/All Offers screen.



Two APIs exist for unsubscribing users from a channel package:

- [142 Finish Channel Package Subscription \(p.373\)](#)
Available only for Online accounts. The customer will continue to have access to the channel package until the next planned billing date, at which time access will terminate and no charge will be levied.
- [143 Remove Channel Package Subscription \(p.375\)](#)
Available for online and operator accounts. The customer loses access to the channel package immediately, and no further charges are levied. Any prepayment is forfeited.

To learn how to identify account types, see [118 Identify Account Type \(p.308\)](#).

141.2 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Terminate a Channel Package subscription for an online account	142 Finish Channel Package Subscription (p.373)
Terminate an Channel Package subscription for a traditional account	143 Remove Channel Package Subscription (p.375)
Identify whether an account is an online or traditional account	118 Identify Account Type (p.308).
List Channel Packages in a My Offers/All Offers screen	160 List Channel Package Offers (p.438)



142 Finish Channel Package Subscription

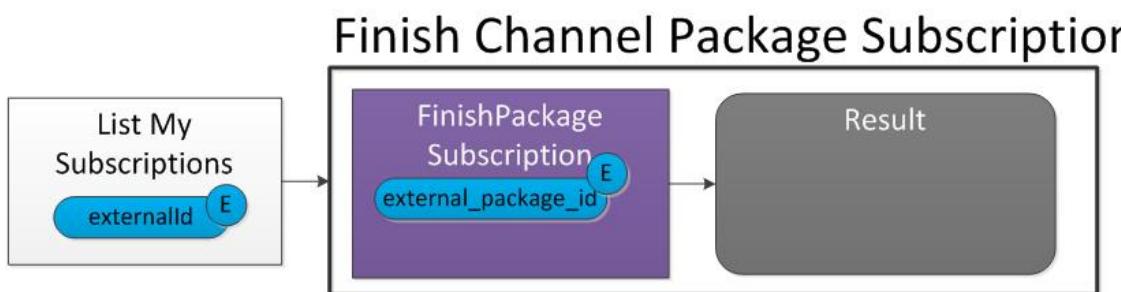
142.1 Overview

Online accounts can finish a channel package. Finishing a channel package subscription sets the end date of the subscription to the next planned debit date. The account will have access to the channel package until that date.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

142.2 Requests and Responses



Request: [FinishPackageSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified channel package. The following parameters are required:

Parameter	Description
<i>external_package_id</i>	The channel package external ID.

Response: [Result \(from RiGHTv Front-end API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server

142.3 Related Tutorials

To	Refer To Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Unsubscribe from a Channel Package	141 Unsubscribe from a Channel Package (p.371)

142.4 Example

`https://<operator>.tvaas.com/RTEFacade/FinishPackageSubscription?external_package_id=Trial%20Package%202&client=json`

This request finishes the account's subscription to the Trial Package 2 channel package.

The response below indicates successful finishing of subscription to the channel package.

```
{
  "metadata": {
    "request": "FinishServiceSubscription",
    "timestamp": 1459417187726
  },
  "response": {
    "message": "'Finish Service Subscription' operation finished successfully",
    "status": "SUCCESS"
  }
}
```



143 Remove Channel Package Subscription

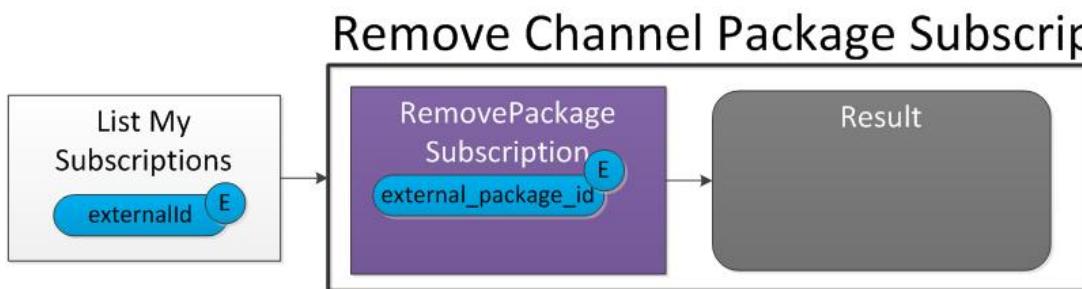
143.1 Overview

Online and operator accounts can remove channel package subscriptions. Removing a package subscription ends access immediately.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

143.2 Requests and Responses



Request: [RemovePackageSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the channel package. The following parameters are required:

Parameter	Description
<i>external_package_id</i>	The channel package external ID.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

143.3 Related Tutorials

To	Refer to Tutorial
Create a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Unsubscribe from a Channel Package	141 Unsubscribe from a Channel Package (p.371)

143.4 Example

`https://<operator>.tvaas.com/RTEFacade/RemovePackageSubscription?external_id=Trial%20Package%20&client=json`

This request removes the account's subscription to the Trial Package 2 channel package.

The response below indicates successful removal of subscription to the channel package.

```
[{"metadata": {  
    "request": "RemovePackageSubscription",  
    "timestamp": 1460462417438  
},  
 "response": {  
    "message": "'Remove Package Subscription' operation finished successfully",  
    "status": "SUCCESS"  
}}
```

144 Performing a Search

144.1 Overview

A Search facility provides a fast and effective means for content discovery. Searches can be performed on a range of content types, for example: movies, series, episodes, live programs, actors, directors or writers.

Search results can be filtered by: genres, parental control level, compatible device. Search results can be sorted by relevance or alphabetically.

A number of different search types can be performed.



Note

Recommended practice is to perform separate searches for VOD movies, VOD TV shows, and live programs, instead of performing a general search across all content types.

144.2 Related Tutorials

To	Refer to Tutorial
Perform a general search	145 Perform a General Search (p.378)
Search for live programs	146 Performing a Live Programs Search (p.384)
Search for videos (movies and TV Shows)	147 Performing a Video Search (p.390)
Add auto-suggestion to search	148 Adding Auto-suggestion to Search (p.397)

145 Perform a General Search

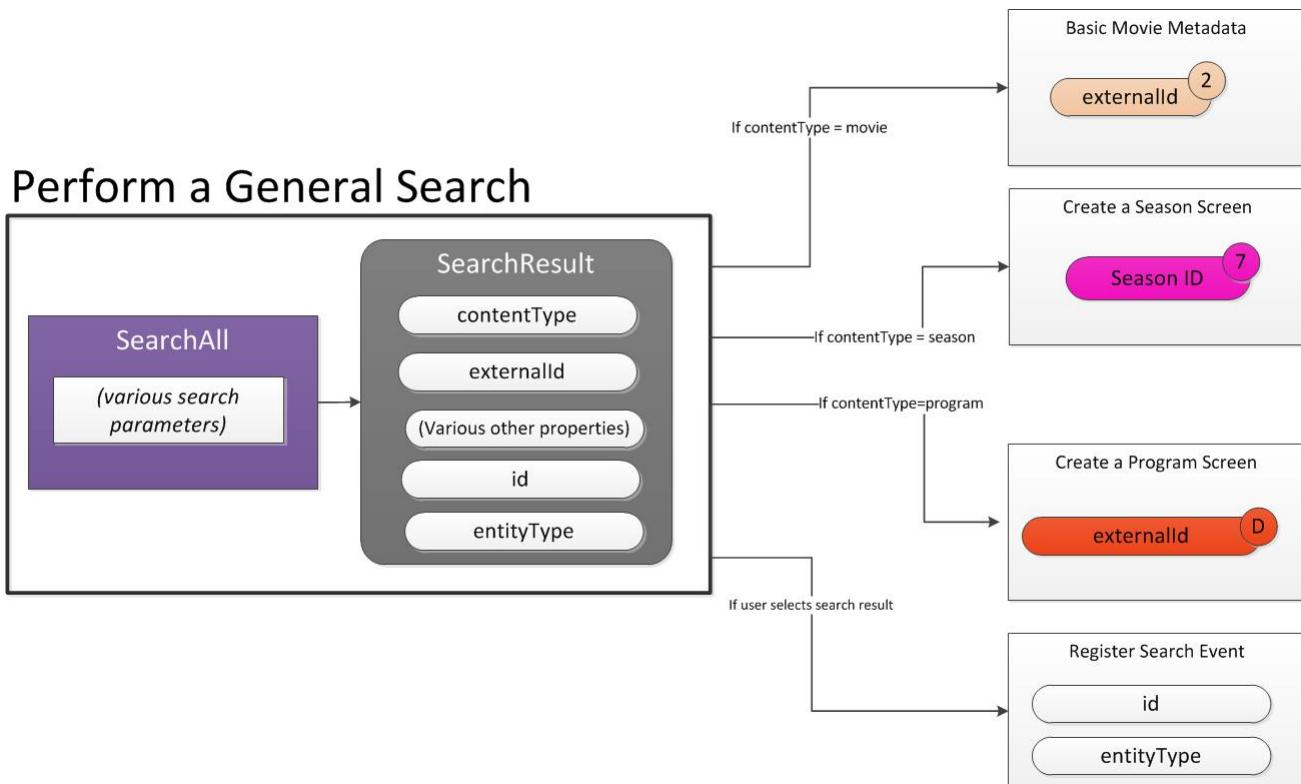
145.1 Overview

A general search searches across all videos and live programs. The search returns any content whose name or title includes the search term.

Note

Recommended practice is to instead perform separate searches for VOD movies, VOD TV shows, and live programs.

145.2 Requests and Responses



Request: [SearchAll](#) (from COMPASS Search RT API)

Searches across all content types, searching within the name and title. All available parameters are listed below.

Parameter	Description
<i>text</i>	The search string entered. Can include multiple terms separated by white space. This is the only mandatory parameter.
<i>language_code</i>	Specifies the language in which to search. If not specified, will use the customer's default language if specified or the primary language defined within the system.
<i>max_results</i>	The maximum number of search results to return. The default value is 20.
<i>image_name</i>	The name of the image type to return. Typically <code>HorizontalImage</code> or <code>VerticalImage</code> . If not specified, the search results will not include images.
<i>fuzzy</i>	Turns fuzzy search on (<code>true</code>) and off (<code>false</code>). If set to <code>false</code> (default), search results include only results that include the search term as entered. If set to <code>true</code> , search results also include results that include words similar to the search term, for example, alternative or corrected spellings.
<i>group_entities</i>	Groups results by type together if set to <code>true</code> . Results will be grouped in the following order: Video movies, Video series (episodes & seasons), Live programs. If set to <code>false</code> (default), results are not grouped by type.
<i>series_content_type</i>	Lists the type of contents to be returned in the search results, and can only be used if <i>group_entities</i> is set to <code>true</code> . List the content types to be returned separated by commas. Valid values include <code>series</code> , <code>episode</code> .
<i>from</i>	The index number of the first result to display. Useful for displaying search results over a number of screens.
<i>count</i>	The number of results to return, starting from the <i>from</i> index.
<i>availability_type</i>	The availability type of live content to be found. Valid values are: "all", "live". Default value: "live".

For more information, see the COMPASS Search RT API Developers Guide.

Response: [SearchResult](#) (from COMPASS Search RT API)

This entity presents the search results. This entity is returned by all searching actions. All of the properties are listed below.

Parameter	Description
<i>id</i>	The ID of the search result.
<i>entityType</i>	The type of returned entity, e.g. content.
<i>contentType</i>	The type of content (relevant in case <i>entityType</i> is content). For VOD contents possible values are movie, season, episode. For live programs the only possible value is program. <div style="border: 1px solid black; padding: 5px;"> Note If a result representing a season is clicked (i.e. <i>contentType</i> = <i>season</i>), open the relevant Season Page, using the <i>externalId</i> parameter. </div>
<i>externalId</i>	The external id of the entity.
<i>name</i>	The name of the entity. <div style="border: 1px solid black; padding: 5px;"> Note For a result representing a season (i.e. <i>contentType</i> = <i>season</i>), the name displayed should be constructed as follows: <i>seriesName</i> Season: <i>contentSeq</i> For example: Weeds: Season 7 </div>
<i>imageUrl</i>	The URL of the image of the entity (whose name was given in the 'image_name' parameter).
<i>year</i>	The year of the content (relevant in case <i>entityType</i> is 'content').
<i>rating</i>	The average user rating of the content (relevant in case <i>entityType</i> is 'content').



Parameter	Description
<i>ratersCount</i>	The number of raters for the content (relevant in case entityType is 'content').
<i>externalRating</i>	The external rating of the content (relevant in case entityType is 'content').
<i>prLevel</i>	The level of the Parental Rating for the content (relevant in case entityType is 'content'). Returned as a number from 0 (low) to 999 (high).
<i>genres</i>	The array of genres associated with the content (relevant in case entityType is 'content'). This array includes only one property, <i>name</i> , the name of the genre.
<i>contentSeq</i>	The sequential number for the content item in the season (relevant in case contentType is 'episode' or 'program').
<i>seasonName</i>	The name of the season (relevant in case contentType is 'episode' or 'program').
<i>seriesName</i>	The name of the series (relevant in case contentType is 'episode', 'season' or 'program').
<i>sourceChannelId</i>	The external ID of the live channel in which the content was originally broadcast (relevant for VOD content).
<i>broadcastStartTime</i>	The start time of a live program broadcast (relevant in case contentType is 'program').
<i>broadcastEndTime</i>	The end time of a live program broadcast (relevant in case contentType is 'program').
<i>channelName</i>	The name of the channel in which a live program is broadcasted (relevant in case contentType is 'program').
<i>channelNumber</i>	The number of the channel in which a live program is broadcast (relevant in case contentType is 'program').



Parameter	Description
<i>availabilities</i>	The array of availabilities associated with the content (relevant in case entityType is ‘content’). This array includes only one property: <i>externalId</i> : the ID of the video (for VOD content) or program (for Live content).

For more information, see the COMPASS Search RT API Developers Guide.

145.3 Related Tutorials

To	Refer to Tutorial
Learn about searching in general	144 Performing a Search (p.377)
Search for live programs	146 Performing a Live Programs Search (p.384)
Search for videos (movies and TV Shows)	147 Performing a Video Search (p.390)
Get metadata on a movie to display with the search result	34 Get Movie Metadata (p.108)
Get metadata on a live program to display with the search result	70 Get Program Metadata (p.205)
Get metadata on a TV Show season to display with the search result	61 Create a VOD Season Screen (p.179)
Register a search event, to improve future searches	202 Register Search Event (p.542)



145.4 Example

<https://<operator>.tvaas.com/search/compass/SearchAll?text=Glee&client=json>

This request gets a list of all contents whose name or title includes the search term Glee.

The response extract below shows one element in the search response.

```
{  
    "contentSeq": "1",  
    "prLevel": "0",  
    "year": "2010",  
    "entityType": "content",  
    "seriesName": "Glee S2",  
    "externalRating": "8.3",  
    "availabilities": [  
        {  
            "externalId": "Glee S2 Ep1"  
        }  
    ],  
    "externalId": "Glee S2 Ep1",  
    "genres": [  
        {  
            "name": "Drama"  
        },  
        {  
            "name": "Comedy"  
        },  
        {  
            "name": "Music"  
        }  
    ],  
    "seasonName": "2",  
    "name": "Glee S2 Ep1",  
    "id": "22323",  
    "contentType": "episode",  
    "isCatchup": 0  
},  
r
```



146 Performing a Live Programs Search

146.1 Overview

A live programs search searches for the entered search string within the names of all live programs.

To learn how to indicate which programs are catch-up enabled, see [173 Identify Catch-Up Enabled Programs \(p.483\)](#).

The screenshot shows a search interface with a search bar at the top containing the text "The". A red box highlights this search term. Below the search bar, there are three main sections: "Movies", "TV Shows", and "Live".

Movies

- Meet the Parents
2000, Family, Comedy
- Star Wars: The Force Awakens
2015, Fantasy, Action, Adventure
- The Amazing Spider-Man
2012, Fantasy, Action, Adventure
- The Avengers
2012, Sci-Fi, Action, Adventure

TV Shows

- The Simpsons 1
1989, Animation, Comedy
- The Sopranos 1
1999, Drama, Crime
- The Larry Sanders Show 1
1992, Family, Comedy
- The Walking Dead 1
2011, Drama, Horror

Live

- The Thing
LUXE.TV at Today 12:00, 1982, Horror
- The Game
MSS ENC at Today 14:00
- The Game
MSS ENC at Today 14:30
- The Sopranos
MSS CLR at Today 17:00

146.2 Requests and Responses

Request: [SearchLivePrograms](#) (from COMPASS Search RT API)

The following parameters can be specified.



Parameter	Description
<i>text</i>	The search string entered. Can include multiple terms separated by white space. This is the only mandatory parameter.
<i>language_code</i>	Specifies the language in which to search. If not specified, will use the customer's default language if specified or the primary language defined within the system.
<i>max_results</i>	The maximum number of search results to return. The default value is 20.
<i>image_name</i>	The name of the image type to return. Typically <code>HorizontalImage</code> or <code>VerticalImage</code> . If not specified, the search results will not include images.
<i>fuzzy</i>	Turns fuzzy search on (<code>true</code>) and off (<code>false</code>). If set to <code>false</code> (default), search results include only results that include the search term as entered. If set to <code>true</code> , search results also include results that include words similar to the search term, for example, alternative or corrected spellings.
<i>genres</i>	A comma-delimited list of genres, identified by their external IDs. Only live programs that match at least one of the genres listed will be included in the search results.
<i>device_models</i>	A comma-delimited list of device models, identified by their device model external IDs. Only programs that are licensed to be viewed on at least one of the models listed will be included in the search results.
<i>channels</i>	A comma-delimited list of channels, identified by their channel external IDs. Only programs to be broadcast on at least one of the channels listed will be included in the search results.
<i>max_pr_level</i>	The maximum level of the parental rating allowed in the returned content items, as a number from 0 (low) to 999 (high).
<i>order_by</i>	Defines whether to sort the list by <code>name</code> , <code>year</code> , <code>broadcastStartTime</code> , <code>relevance</code> (default).

Parameter	Description
<i>from</i>	The index number of the first result to display. Useful for displaying search results over a number of screens.
<i>count</i>	The number of results to return, starting from the <i>from</i> index.
<i>availability_type</i>	The availability type of live content to be found. Valid values are: "all", "live". Default value: "live".

For more information, see the COMPASS Search RT API Developers Guide.

Response: [SearchResult](#) (from COMPASS Search RT API)

This entity presents the search results. All of the properties are listed below.

Parameter	Description
<i>id</i>	The ID of the search result.
<i>entityType</i>	The type of returned entity, e.g. content.
<i>contentType</i>	For live programs the only possible value is program.
<i>externalId</i>	The external id of the entity.
<i>name</i>	The name of the entity.
<i>imageUrl</i>	The URL of the image of the entity (whose name was given in the 'image_name' parameter).
<i>year</i>	The year of the content (relevant in case entityType is 'content').
<i>rating</i>	The average user rating of the content (relevant in case entityType is 'content').
<i>ratersCount</i>	The number of raters for the content (relevant in case entityType is 'content').
<i>externalRating</i>	The external rating of the content (relevant in case entityType is 'content').



Parameter	Description
<i>prLevel</i>	The level of the Parental Rating for the content (relevant in case entityType is 'content'). Returned as a number from 0 (low) to 999 (high).
<i>genres</i>	The array of genres associated with the content (relevant in case entityType is 'content'). This array includes only one property, <i>name</i> , the name of the genre.
<i>contentSeq</i>	The sequential number for the content item in the season (relevant in case contentType is 'episode' or 'program').
<i>seasonName</i>	The name of the season (relevant in case contentType is 'episode' or 'program').
<i>seriesName</i>	The name of the series (relevant in case contentType is 'episode', 'season' or 'program').
<i>sourceChannelId</i>	The external ID of the live channel in which the content was originally broadcast (relevant for VOD content).
<i>broadcastStartTime</i>	The start time of a live program broadcast (relevant in case contentType is 'program').
<i>broadcastEndTime</i>	The end time of a live program broadcast (relevant in case contentType is 'program').
<i>channelName</i>	The name of the channel in which a live program is broadcasted (relevant in case contentType is 'program').
<i>channelNumber</i>	The number of the channel in which a live program is broadcast (relevant in case contentType is 'program').
<i>availabilities</i>	The array of availabilities associated with the content (relevant in case entityType is 'content'). This array includes only one property: <i>externalId</i> : the ID of the video (for VOD content) or program (for Live content).



146.3 Related Tutorials

To	Refer to Tutorial
Learn about searching in general	144 Performing a Search (p.377)
Get metadata on a live program to display with the search result	70 Get Program Metadata (p.205)
Register a search event, to improve future searches	202 Register Search Event (p.542)

146.4 Example

`https://<operator>.tvaas.com/search/compass/SearchLivePrograms?image_name=HorizontalImage&language_code=en&text=the&order_by=broadcastStartTime&client=json`

This request gets a list of all live programs that include the search term `The` in their English titles. The search results will be returned with horizontal images and will be ordered by broadcast start time.

The response extract below shows one of the search results.

```
{  
    "prLevel": "0",  
    "year": "1982",  
    "entityType": "content",  
    "channelNumber": "3",  
    "availabilities": [  
        {  
            "externalId": "eoae2lsr5o6e4aln18jjte62ns"  
        }  
    ],  
    "broadcastStartTime": "1490004000000",  
    "broadcastEndTime": "1490013000000",  
    "genres": [  
        {  
            "name": "Horror"  
        },  
        {  
            "name": "Sci-Fi"  
        }  
    ],  
    "imageUrl": "/eoae2lsr5o6e4aln18jjte62ns_HorizontalImage.jpg",  
    "name": "The Thing",  
    "channelName": "LUXE.TV",  
    "id": "25150",  
    "contentType": "program",  
    "isCatchup": 0  
},
```

147 Performing a Video Search

147.1 Overview

A video search searches for the entered search string within the names of all videos (movies and TV Shows). Separate video searches should be performed for movies and for TV shows. Results should be separated by type.

To learn how to indicate which programs are catch-up enabled, see [173 Identify Catch-Up Enabled Programs \(p.483\)](#).

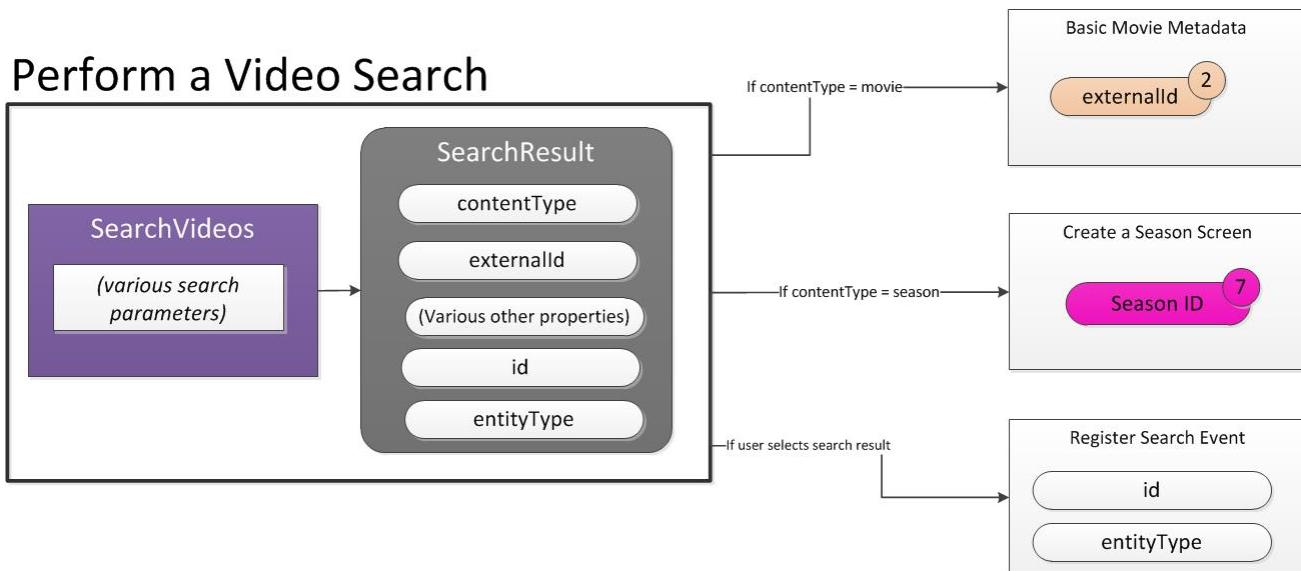
Note for Analytics: The Analytics Platform extracts data sent when searching for videos for use in many of its charts and KPIs.

The screenshot shows a search interface with a search bar containing 'The'. Below the search bar, there are three main sections: 'Movies', 'TV Shows', and 'Live'. Each section contains four items, each with a thumbnail, the title, and a brief description. The 'Movies' section includes 'Meet the Parents' (2000, Family, Comedy), 'Star Wars: The Force Awakens' (2015, Fantasy, Action, Adventure), 'The Amazing Spider-Man' (2012, Fantasy, Action, Adventure), and 'The Avengers' (2012, Sci-Fi, Action, Adventure). The 'TV Shows' section includes 'The Simpsons 1' (1989, Animation, Comedy), 'The Sopranos 1' (1999, Drama, Crime), 'The Larry Sanders Show 1' (1992, Family, Comedy), and 'The Walking Dead 1' (2011, Drama, Horror). The 'Live' section includes 'The Thing' (LUXE.TV at Today 12:00, 1982, Horror), 'The Game' (MSS ENC at Today 14:00), 'The Game' (MSS ENC at Today 14:30), and 'The Sopranos' (MSS CLR at Today 17:00). Each item has a small 'viaccess-orca' logo below it. Navigation arrows are visible on the right side of each section.

Movies	TV Shows	Live
Meet the Parents 2000, Family, Comedy	Star Wars: The Force Awakens 2015, Fantasy, Action, Adventure	The Amazing Spider-Man 2012, Fantasy, Action, Adventure
The Avengers 2012, Sci-Fi, Action, Adventure	The Simpsons 1 1989, Animation, Comedy	The Sopranos 1 1999, Drama, Crime
The Larry Sanders Show 1 1992, Family, Comedy	The Walking Dead 1 2011, Drama, Horror	The Thing LUXE.TV at Today 12:00, 1982, Horror
The Game MSS ENC at Today 14:00	The Game MSS ENC at Today 14:30	The Sopranos MSS CLR at Today 17:00

147.2 Requests and Responses

Perform a Video Search



Request: [SearchVideos](#) (from COMPASS Search RT API)

The following parameters can be specified.

Property	Description
<i>text</i>	The search text, searched for in movie and series names.
<i>language_code</i>	Specifies the language in which to search. If not specified, will use the customer's default language if specified or the primary language defined within the system.
<i>max_results</i>	The maximum number of search results to return. The default value is 20.
<i>image_name</i>	The name of the image type to return. Typically <code>HorizontalImage</code> or <code>VerticalImage</code> . If not specified, the search results will not include images.
<i>fuzzy</i>	Turns fuzzy search on (<code>true</code>) and off (<code>false</code>). If set to <code>false</code> (default), search results include only results that include the search term as entered. If set to <code>true</code> , search results also include results that include words similar to the search term, for example, alternative or corrected spellings.

Property	Description
<i>genres</i>	A comma-delimited list of genres, identified by their external IDs. Only videos that match at least one of the genres listed will be included in the search results.
<i>services</i>	A comma-delimited list of service external IDs identified by their external IDs. Only videos that match at least one of the services listed will be included in the search results.
<i>device_models</i>	A comma-delimited list of device models, identified by their device model external IDs. Only videos that are licensed to be viewed on at least one of the models listed will be included in the search results. If specified, overrides the <i>delivery_mode</i> parameter.
<i>source_channel_ids</i>	A comma-delimited list of source channels, identified by their channel external IDs. Only programs originally broadcast on at least one of the channels listed will be included in the search results.
<i>delivery_mode</i>	A comma-delimited list of delivery modes, with possible values download and streaming. Only videos that can be delivered by at least one of the listed modes will be included in the search results. This parameter is overridden if the <i>device_models</i> parameter is specified.
<i>quality</i>	A comma-delimited list of video qualities, such as HD, SD and 3D. Only videos that can be delivered by at least one of the listed qualities will be included in the search results.
<i>content_type</i>	A comma-delimited list of content types to be included in the search results. Valid values include movie, season and episode.
<i>max_pr_level</i>	The maximum level of the parental rating allowed in the returned content items, as a number from 0 (low) to 999 (high).
<i>order_by</i>	Defines whether to sort the list by name, year and relevance (default).
<i>from</i>	The index number of the first result to display. Useful for displaying search results over a number of screens.



Property	Description
<i>count</i>	The number of results to return, starting from the <i>from</i> index.

For more information, see the COMPASS Search RT API Developers Guide.

Response: [SearchResult](#) (from COMPASS Search RT API)

This entity presents the search results. All of the properties are listed below.

Parameter	Description
<i>id</i>	The ID of the search result.
<i>entityType</i>	The type of returned entity, e.g. content.
<i>contentType</i>	The type of content (relevant in case entityType is content). For VOD contents possible values are movie, season, episode. For live programs the only possible value is program.
<i>externalId</i>	The external id of the entity.
<i>name</i>	The name of the entity.
<i>imageUrl</i>	The URL of the image of the entity (whose name was given in the 'image_name' parameter).
<i>year</i>	The year of the content (relevant in case entityType is 'content').
<i>rating</i>	The average user rating of the content (relevant in case entityType is 'content').
<i>ratersCount</i>	The number of raters for the content (relevant in case entityType is 'content').
<i>externalRating</i>	The external rating of the content (relevant in case entityType is 'content').
<i>prLevel</i>	The level of the Parental Rating for the content (relevant in case entityType is 'content'). Returned as a number from 0 (low) to 999 (high).



Parameter	Description
<i>genres</i>	The array of genres associated with the content (relevant in case entityType is 'content'). This array includes only one property, <i>name</i> , the name of the genre.
<i>contentSeq</i>	The sequential number for the content item in the season (relevant in case contentType is 'episode' or 'program').
<i>seasonName</i>	The name of the season (relevant in case contentType is 'episode' or 'program').
<i>seriesName</i>	The name of the series (relevant in case contentType is 'episode', 'season' or 'program').
<i>sourceChannelId</i>	The external ID of the live channel in which the content was originally broadcast (relevant for VOD content).
<i>broadcastStartTime</i>	The start time of a live program broadcast (relevant in case contentType is 'program').
<i>broadcastEndTime</i>	The end time of a live program broadcast (relevant in case contentType is 'program').
<i>channelName</i>	The name of the channel in which a live program is broadcasted (relevant in case contentType is 'program').
<i>channelNumber</i>	The number of the channel in which a live program is broadcast (relevant in case contentType is 'program').
<i>availabilities</i>	The array of availabilities associated with the content (relevant in case entityType is 'content'). This array includes only one property: <i>externalId</i> : the ID of the video (for VOD content) or program (for Live content).



147.3 Related Tutorials

To	Refer To Tutorial
Learn about searching in general	144 Performing a Search (p.377)
Get metadata on a movie to display with the search result	34 Get Movie Metadata (p.108)
Register a search event, to improve future searches	202 Register Search Event (p.542)

147.4 Example

`https://<operator>.tvaas.com/search/compass/SearchVideos?image_name=HorizontalImage&language_code=en&content_type=movie&text=the&order_by=name&client=json`

This request gets all movie videos that include the search term `the` in their English titles. The search results will be returned with horizontal images and will be ordered by name.

`https://<operator>.tvaas.com/search/compass/SearchVideos?image_name=HorizontalImage&language_code=en&content_type=season&text=the&order_by=name&client=json`

This request gets all TV Shows that include the search term `the` in their English titles. The search results will be returned with horizontal images and will be ordered by name.

A sample response is shown below.

```
{  
    "contentSeq": "1",  
    "ratersCount": "15",  
    "prLevel": "0",  
    "year": "1989",  
    "entityType": "content",  
    "seriesName": "The Simpsons",  
    "availabilities": [  
        {  
            "externalId": "1~!The Simpsons~!2"  
        }  
    ],  
    "rating": "3.26",  
    "externalId": "1~!The Simpsons",  
    "genres": [  
        {  
            "name": "Animation"  
        },  
        {  
            "name": "Comedy"  
        }  
    ],  
    "imageUrl": "/attachments/TheSimpsons_OutHorizontalImage.jpg",  
    "name": "1",  
    "id": "22290",  
    "contentType": "season"  
},
```

148 Adding Auto-suggestion to Search

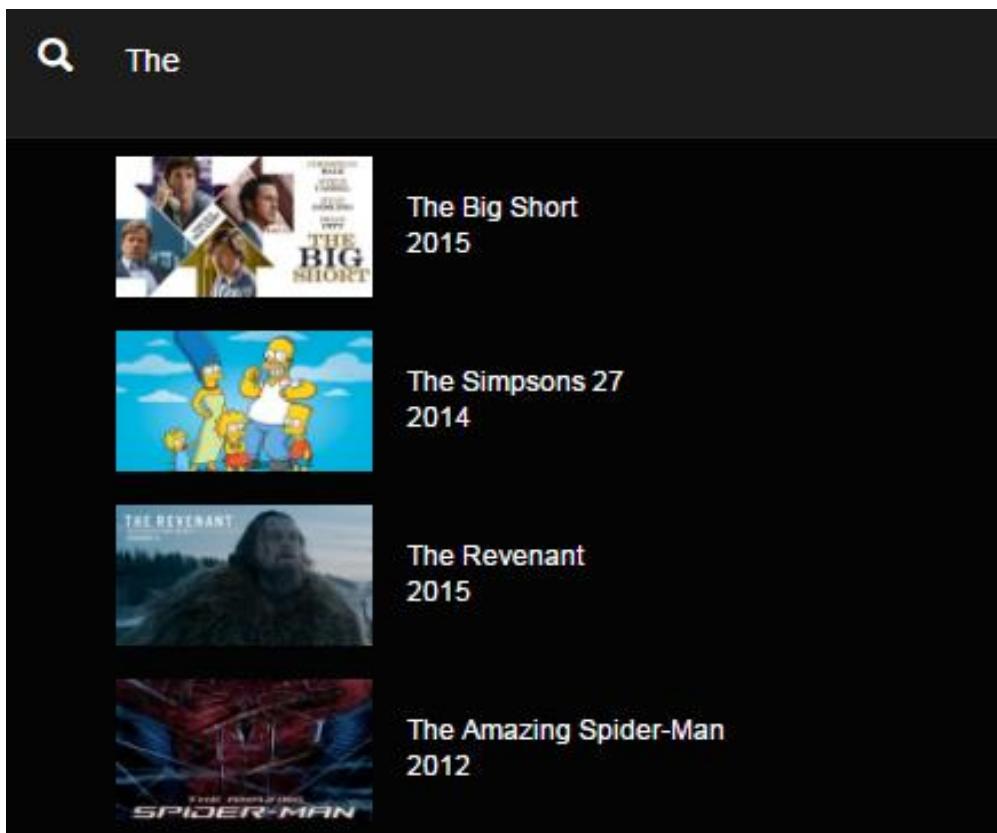
148.1 Overview

By enabling auto-suggestion, the user will be presented with search results immediately after having entered a few characters in the search field, without having to press the search button. The user can then choose a result from the search results, or continue typing into the search field. The auto-suggestions are updated and refined after each additional character is entered.

RiGHTv provides the most likely results based on the few characters entered.

The auto-suggestion search can be configured to search within the titles of videos and live programs.

Note for Analytics: The Analytics Platform extracts data sent when conducting a search.



148.2 Requests and Responses

Use the *SearchSuggest* request to provide auto-suggestion to the search facility.

Request: SearchSuggest (from COMPASS Search RT API)

Searches within the titles of videos and live programs. The following parameters can be specified:

Parameter	Description
<i>text</i>	The search string entered. Can include multiple terms separated by white space. This is the only mandatory parameter.
<i>language_code</i>	Specifies the language in which to search. If not specified, will use the customer's default language if specified, or the primary language defined within the system.
<i>max_results</i>	The maximum number of search results to return. The default value is 20.
<i>image_name</i>	The name of the image type to return. Typically HorizontalImage or VerticalImage. If not specified, the search results will not include images.
<i>entity_type</i>	A comma-delimited list of entity types within which to search, for example content.
<i>content_type</i>	A comma-delimited list of content types to include in search results. Valid values include movie, season, episode, program. This parameter can only be specified if <i>entity_type</i> is set to content or not used.
<i>from</i>	The index number of the first result to display. Useful for displaying search results over a number of screens.
<i>count</i>	The number of results to return, starting from the <i>from</i> index.

For more information, see the COMPASS Search RT API Developers Guide.

Response: [SuggestResult](#) (from COMPASS Search RT API)

Provides search results.

Property	Description
<i>id</i>	The ID of the search result.
<i>entityType</i>	The type of returned entity, for example content.
<i>contentType</i>	The type of content (relevant in case entityType is content). For VOD contents possible values are movie, season, episode. For live programs the only possible value is program.



Property	Description
<i>externalId</i>	The external id of the entity.
<i>name</i>	The name of the entity.
<i>year</i>	The year of the content (relevant in case <i>entityType</i> is <code>content</code>).
<i>contentSeq</i>	The sequential number for the content item in the season (relevant if <i>contentType</i> is <code>episode</code> or <code>program</code>).
<i>seasonName</i>	The name of the season (relevant if <i>contentType</i> is <code>episode</code> or <code>program</code>).
<i>seriesName</i>	The name of the series (relevant if <i>contentType</i> is <code>season</code> , <code>episode</code> or <code>program</code>).
<i>broadcastStartTime</i>	The start time of a live program broadcast (relevant in case <i>contentType</i> is <code>program</code>).
<i>broadcastEndTime</i>	The end time of a live program broadcast (relevant in case <i>contentType</i> is <code>program</code>).
<i>channelName</i>	The name of the channel in which a live program is broadcasted (relevant in case <i>contentType</i> is <code>program</code>).
<i>channelNumber</i>	The number of the channel in which a live program is broadcasted (relevant in case <i>contentType</i> is <code>program</code>).
<i>imageUrl</i>	The URL of the image of the entity (whose name was given in the <code>image_name</code> parameter).
<i>availabilities</i>	The array of availabilities associated with the content (relevant in case <i>entityType</i> is <code>content</code>). This array includes only one property: <code>externalId</code> : the ID of the video (for VOD content) or program (for Live content).

For more information, see the COMPASS Search RT API Developers Guide.

148.3 Additional Information

Apart from showing the content item image and its name, additional information can be displayed for each result. The table below lists examples of additional information that can be displayed, and provides instructions for implementation



Information	Implementation
An indication of whether the content item is VOD or Live.	Refer to the table above – use the <i>contentType</i> parameter.
For Live content, the time and date of the program's first airing.	See 70 Get Program Metadata (p.205)
For Live TV show episodes, the season and episode number.	See 70 Get Program Metadata (p.205)

148.4 Related Tutorials

To	Refer To Tutorial
Learn about searching in general	144 Performing a Search (p.377)
Get metadata on a movie to display with the search result	34 Get Movie Metadata (p.108)
Get metadata on a live program to display with the search result	70 Get Program Metadata (p.205)
Register a search event, to improve future searches	202 Register Search Event (p.542)

148.5 Example

`https://<operator>.tvaas.com/search/compass/SearchSuggest?image_name=HorizontalImage&language_code=en&entity_type=content&content_type=movie%2Cseason%2Cprogram&text=The&client=json`

This request gets all contents (movies, seasons and live programs) that include the search term the in their English titles. The search results will be returned with horizontal images.

The sample response below shows one search result.

```
  "year": "2015",
  "entityType": "content",
  "availabilities": [
    {
      "externalId": "The Revenant_TVOD"
    },
    {
      "externalId": "67ec2294-ec2a-4566-8ac7-678842b30cbd"
    },
    {
      "externalId": "The Revenant_SVOD"
    }
  ],
  "imageUrl": "/attachments/theRevenant_horizontal1.jpg",
  "name": "The Revenant",
  "externalId": "The Revenant",
  "id": "23544",
  "contentType": "movie"
},
```



149 Display Images

149.1 Overview

The system stores images for many entity types. For example, a VideoAsset entity can store horizontal and vertical versions of the movie poster, that can be used as thumbnails in a catalog screen.

This tutorial explains how to retrieve those images for display within an application.

The table below lists entities that include image information. For each, the table indicates which API Request retrieves the entity, and within which API the Request is located.

Entity (Response)	Relevant Request	API	
		RiGHTv Front-end API	Compass RT API
AggregatedVideo	GetUnifiedList	✓	
VideoAsset	GetVideoAsset	✓	
VideoPackage	GetVideoPackage	✓	
AssetChapter	GetVideoAsset	✓	
Channel	GetChannelList	✓	
Program	<i>GetProgramList</i>	✓	
Service	<i>GetServiceList</i>	✓	
Season	<i>GetSeriesSeasonList</i>		✓
VideoRecommendation	GetVideoRecommendationList		✓
LiveRecommendation	GetLiveRecommendationList		✓



149.2 Requests and Responses

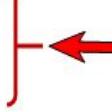
Many responses to API requests include image information. The name of the array in which this information is presented depends on whether the API is part of the RiGHTv Front-end API or the COMPASS RT API:

API	Array Name
RiGHTv Front-end API	Attachment
COMPASS RT API	Image

For both Attachment and Image entities, the following properties are returned:

Property	Description
<i>name</i>	The name of the image and is typically standardized across the system, such as <code>HorizontalImage</code> or <code>VerticalImage</code> .
<i>value</i>	The path and file name of the image file.

Example responses are shown below:

Attachments (RiGHTv Front-end API)	Images (COMPASS RT API)
<pre>"response": { "advisories": "", "studio": "", "prLevel": 3, "attachments": [{ "responseElementType": "Attachment", "assetId": "", "name": "VerticalImage", "assetName": "", "value": "/attachments/argo_vertical_after_edit.jpg" }, { "responseElementType": "Attachment", "assetId": 80006, "name": "Trailer", "assetName": "Test Trailer (es)", "value": "true" }, { "responseElementType": "Attachment", "assetId": "", "name": "HorizontalImage", "assetName": "", "value": "/attachments/argo-horizontal-After Edit.jpg" }] }</pre>	<pre>"images": [{ "name": "HorizontalImage", "value": "/attachments/hateful-eight_horizontal.png" }, { "name": "VerticalImage", "value": "/attachments/the-hateful-8-vertical.jpg" }]</pre> 

Note that the *Attachments* array may include video attachments as well (usually named `Trailer`). The use of video attachments is explained in [43 Playing a Trailer \(p.137\)](#).

149.3 Retrieving Images

The image is retrieved using the *value* parameter in the response, as follows:

`https://[site_prefix]/RTEFacade/images/attachments/[filename including extension]`

For example, if the *value* property is `attachments/jurassic_world_ver.jpg`, the following call will obtain that image:

`https://<operator>.tvaas.com/RTEFacade/images/attachments/jurassic_world_ver.jpg`

149.4 On the Fly Resizing

The image can be retrieved at a customized size by specifying *width* and *height* parameters, in terms of pixels. For example, the call below will retrieve the same poster as the one above, but resized to 111 pixels across by 152 pixels high:

`https://<operator>.tvaas.com/RTEFacade/images/attachments/jurassic_world_ver.jpg?width=111&height=152`

149.5 Example 1: From RiGHTv

`https://<operator>.tvaas.com/RTEFacade/GetVideoAsset?external_id=Jurassic%20World&language_code=es&client=json`

This call requests information on the movie Jurassic World.

The Attachments array from the response is shown below.

```
"attachments": [
  {
    "responseElementType": "Attachment",
    "assetId": "",
    "name": "VerticalImage",
    "assetName": "",
    "value": "/attachments/jurassic_world_ver.jpg"
  },
  {
    "responseElementType": "Attachment",
    "assetId": 80006,
    "name": "Trailer",
    "assetName": "Test Trailer (es)",
    "value": "true"
  },
  {
    "responseElementType": "Attachment",
    "assetId": "",
    "name": "HorizontalImage",
    "assetName": "",
    "value": "/attachments/jurassic_world_hor.jpg"
  }
],
```

To obtain the horizontal image, use the *value* corresponding to the attachment *name* HorizontalImage. Set the *width* and *height* values as required.



https://<operator>.tvaas.com/RTEFacade/images/attachments/jurassic_world_ver.jpg?width=111&height=152

149.6 Example 2: From COMPASS

https://<operator>.tvaas.com/compass/GetSeriesSeasonList?series_id=The%20Larry%20Sanders%20Show&client=json

This request calls for a list of all seasons in the Larry Sanders Show.

The images array in the response is shown below.

```
"images": [
  {
    "name": "HorizontalImage",
    "value": "/attachments/TheLarrySandersShow_OutHorizontalImage.jpg"
  },
  {
    "name": "VerticalImage",
    "value": "/attachments/TheLarrySandersShow_VerticalImage.jpg"
  }
]
```

The vertical image can be retrieved as below. The image is retrieved with a width of 111 pixels and height of 152 pixels.

https://<operator>.tvaas.com/RTEFacade/images/attachments/TheLarrySandersShow_VerticalImage.jpg?width=111&height=152

149.7 Example 3: Live Channel Image

The file name of the image to display for the channel is listed within the *Attachments* array within the *Channel* entity.

```
{
  "responseElementType": "Attachment",
  "assetId": "",
  "name": "HorizontalImage",
  "assetName": "",
  "value": "/attachments/usa-network-horizontal.jpg"
},
```

Select which file to display based on the name, and retrieve the attachment, for example:

https://<operator>.tvaas.com/RTEFacade/images/attachments/USA_networks_vertical.jpg

150 Get List of Recommendations

150.1 Overview

Many screens display lists of recommended content. Those lists are generated by the COMPASS content discovery system.

The Content Discovery component (COMPASS) provides lists of recommended content to display to the user. The recommendations can be customized, for example, to include:

- Only content the operator wants to promote
- Only recommended movies
- Only recommended TV shows

Recommendations can be provided for both VOD and Live content.

150.2 Requests and Responses – VOD Content

Request: [GetVideoRecommendationList](#) (from COMPASS RT API)

The recommendation lists provided are customized based on parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>blend</i>	Specify the name of a recommendations blend created by the operator. A blend is an operator-defined mix of multiple recommendation engines used to create a recommendations list. Blends are defined by the operator in the back end.
<i>categories</i>	Choose to include movies from specified VOD catalog categories only by providing a comma-delimited list of category external IDs. For example, set categories to Drama to limit the list of recommended movies to movies from the Drama category.
<i>genres</i>	Limit the recommendations to movies from one or more genres.
<i>max_pr_level</i>	Specify the maximum parental rating level of movies to be listed. For information on parental control, see 193 Exclude Adult Content from Returned Lists (p.522) .

Parameter	Description
content_type	Filter the list of recommendations to include only specified content types. Valid values include Movie, Season, or Program.
max_results	Specify how many results to retrieve.
random	Choose whether the results should be listed in the order generated by the recommendation engine, or in a random order. Set random to false to maintain the original order.
from, count	Specify which results to retrieve by specifying the first index from which to retrieve, and the count of how many elements to return. Use these when displaying recommendations across multiple screens. These parameters are useful only when random is set to false, so that the full list of recommendations from which the subset of recommendations to be returned will remain constant.
device_models.	Limit the results to content that can be viewed on specific device models.
subscription	Choose whether or not to filter results to services to which the account is subscribed. Setting subscription to false will include unsubscribed content in the recommendations list.

For more parameters, see the Compass RT Developers Guide.

Response: [VideoRecommendation \(from COMPASS RT API\)](#)

The returned entity includes various properties. For displaying a catalog, the most relevant are:

Property	Description
<i>externalContentID</i>	The external ID of the content item. This property corresponds to the external ID of a VideoAsset entity, or the external ID of a season.
<i>name</i>	The name of the content item.
<i>Images</i>	An array containing information about image attachments See 149 Display Images (p.402) to learn how to retrieve the image files.



Property	Description
The parameters listed below are relevant for TV Show seasons and series:	

150.3 Example

```
https://<operator>.tvaas.com/compass/
GetVideoRecommendationList?blend=movies&categories=Movies&
subscription=false&max_results=100&max_pr_
level=15&from=0&count=15&device_models=IPad&random=false&client=json
```

This request calls for a list of recommendations for movies as follows:

Parameter	Value	Description
<i>blend</i>	movies	The operator-defined blend called movies will be used to determine which movies to recommend.
<i>categories</i>	Movies	Only movies from the Movies category will be recommended.
<i>subscription</i>	false	The list of recommendations will include movies to which the user is subscribed, and those to which the user is not subscribed.
<i>max_results</i>	100	A total of 100 recommendations will be generated.
<i>max_pr_level</i>	15	Movies with a parental rating level over 15 will not be recommended.
<i>from, count</i>	0, 15	The first 15 recommendations generated will be returned.
<i>random</i>	false	The recommendations will be returned in the order in which they were generated.

The sample below shows the relevant properties of one movie in the recommendation list received. The name, externalContentId and attachment details are indicated.



```
"images": [  
    {  
        "name": "HorizontalImage",  
        "value": "/attachments/hateful-eight_horizontal.png"  
    },  
    {  
        "name": "VerticalImage",  
        "value": "/attachments/the-hateful-8-vertical.jpg"  
    }  
,  
    {"name": "The Hateful Eight",  
     "externalContentId": "The Hateful Eight",  
     "id": 25540,  
     "contentType": "Movie"  
},  
,
```

150.4 Recommendations for Live Content

Request: [GetLiveRecommendationList](#) (from COMPASS RT API)

The recommendation lists provided are customized based on parameters in the action call. The table below presents the most commonly used parameters:

Parameter	Description
<i>blend</i>	Specify the name of a recommendations blend created by the operator. A blend is an operator-defined mix of multiple recommendation engines used to create a recommendations list. Blends are defined by the operator in the back end.
<i>params=program_id</i>	If the blend chosen requires that the current program be specified, as is the case for recommendations for related content, specify the <i>program_id</i> .
<i>genres</i>	Limit the recommendations to programs from one or more genres.
<i>max_pr_level</i>	Specify the maximum parental rating level of programs to be listed. For information on parental control, see 185 Parental Control (p.509) .
<i>content_type</i>	Filter the list of recommendations to include only specified content types. Valid values include Movie, Season, or Program.
<i>max_results</i>	Specify how many results to retrieve.
<i>random</i>	Choose whether the results should be listed in the order generated by the recommendation engine, or in a random order. Set <i>random</i> to false to maintain the original order.
<i>from, count</i>	Specify which results to retrieve by specifying the first index from which to retrieve, and the count of how many elements to return. Use these when displaying recommendations across multiple screens. These parameters are useful only when <i>random</i> is set to false, so that the full list of recommendations from which the subset of recommendations to be returned will remain constant.
<i>device_models.</i>	Limit the results to content that can be viewed on specific device models.
<i>subscription</i>	Choose whether or not to filter results to channels to which the account is subscribed. Setting <i>subscription</i> to false will include unsubscribed content in the recommendations list.



For more parameters, see the Compass RT Developers Guide.

Response: [LiveRecommendation \(from COMPASS RT API\)](#)

The returned entity includes various properties. The most relevant are:

Property	Description
<i>programId</i>	The external ID of the program.
<i>name</i>	The name of the program.
<i>channelId</i>	The external ID of the channel.
<i>images</i>	An array of images associated with the item. See 149 Display Images (p.402) to learn how to retrieve the image files.

150.5 Example

`https://<operator>.tvaas.com/compass/GetLiveRecommendationList?blend=live&max_results=10&max_pr_level=5&device_models=PC,IPad&client=json`

This request calls for a list of recommendations for live content as follows:

Parameter	Value	Description
<i>blend</i>	live	The operator-defined blend called live will be used to determine which programs to recommend.
<i>max_results</i>	10	A total of 10 recommendations will be generated.
<i>max_pr_level</i>	5	Programs with a parental rating level over 5 will not be recommended.
<i>device_models</i>	PC, IPad	Only programs that can be viewed on PC and IPad devices must be recommended.

The response extract below shows the details of one recommended program.

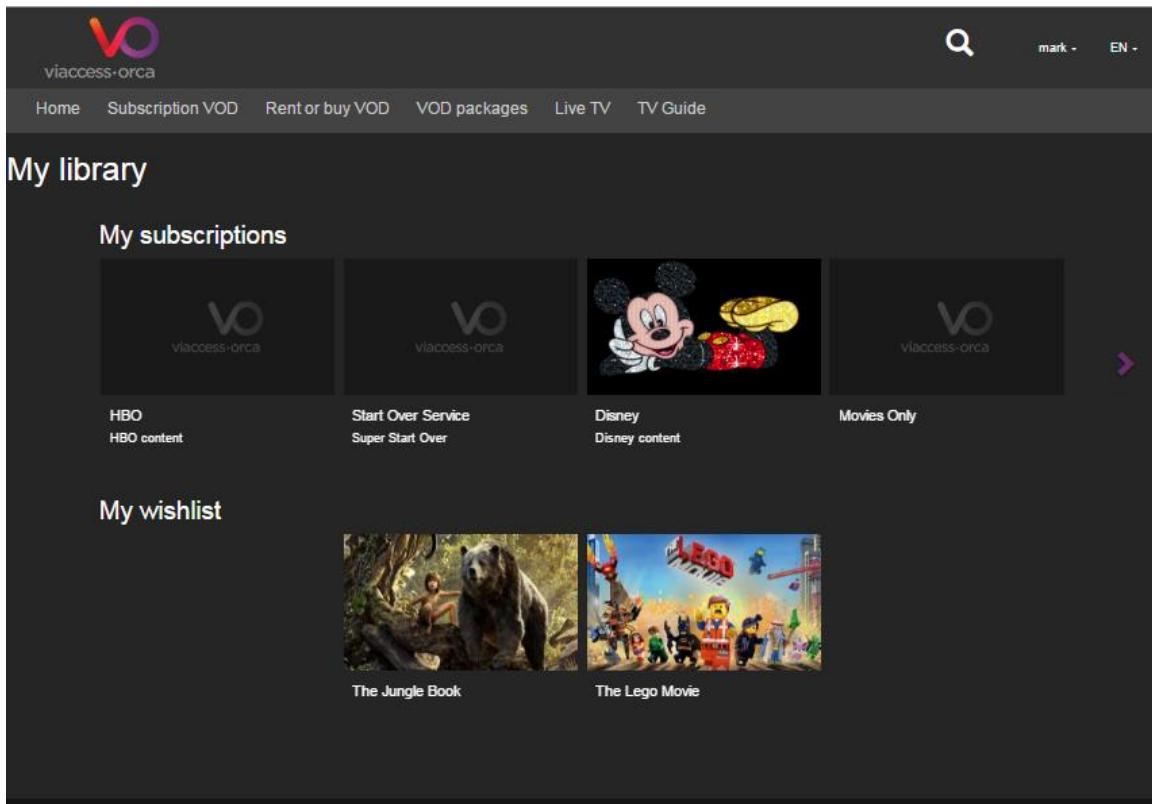


```
{  
  "images": [  
    {  
      "name": "HorizontalImage", ←  
      "value": "/s7n0a31llcfesn3lcipbvb30q_HorizontalImage.jpg"  
    },  
    {  
      "name": "VerticalImage",  
      "value": "/s7n0a31llcfesn3lcipbvb30q_VerticalImage.jpg"  
    }  
  ],  
  "prLevel": 0,  
  "AvailabilityProperties": [],  
  "endDate": 1466343000000,  
  "prName": "NR",  
  "flags": 0,  
  "type": "promotions",  
  "ContentProperties": [],  
  "recommendationReasons": [  
    {  
      "engineLabel": "Label",  
      "reasonTemplateCode": 3000,  
      "engineType": "promotions",  
      "type": "engine"  
    }  
  ],  
  "genres": [  
    {  
      "name": "Romance",  
      "externalId": "Romance",  
      "id": "Romance"  
    },  
    {  
      "name": "Comedy",  
      "externalId": "Comedy",  
      "id": "Comedy"  
    }  
  ],  
  "name": "Friends", ←  
  "id": 25027,  
  "contentType": "Program",  
  "programId": "s7n0a31llcfesn3lcipbvb30q", ←  
  "channelId": "France24Br", ←  
  "startDate": 1466341200000  
},
```

151 Creating a My Library Screen

151.1 Overview

The My Library screen should enable users to see the content to which they have access, divided by offer types. It should also give them quick access to their wishlists and reminders.



151.2 Related Tutorials

To	Refer To Tutorial
List movies purchased through TVOD offers	List My Movies-VOD
List movies purchased through Movie Packages	153 List My VOD Package Movies (p.418)
List SVOD services to which the account is subscribed	154 List My SVOD Subscriptions (p.420)
List Service Plans to which the account is subscribed	155 List My Service Plans (p.423)

To	Refer To Tutorial
List Catch-Up services to which the account is subscribed	172 List My Catch-Up Services (p.479)
List the movies in the user's wish list	156 Show My Wishlist (p.426)
List the reminders the user has set for live programs	157 Show My Reminders (p.428)

152 List My Movies

152.1 Overview

The My Movies section of the My Library screen enable users to see what on-demand movies and TV shows they have purchased.



152.2 Requests and Responses

Request: [GetVideoTicketList](#) (from RiGHTv Front-end API)

Lists all the tickets for videos that can be viewed by the logged in customer. No parameters are required.

Response: [VideoTicket](#) (from RiGHTv Front-end API)

The VideoTicket entity provides various information about each ticket.

Property	Description
<i>template</i>	If set to <code>vod_service</code> , indicates a ticket for a VOD movie.
<i>assetExternalId</i>	The external id of the asset.
<i>name</i>	The name of the move.

For more information, see the RiGHTv Front-end API Developers Guide.

To obtain the image for the movie, use `GetVideo`, as described in [34 Get Movie Metadata \(p.108\)](#).

152.3 Related Tutorials

To	Refer to Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)

152.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetVideoTicketList?client=json>

This request calls for a list of all video tickets ordered by the customer.

The sample response below shows that the account has active access to the movie Goodfellas, ordered as a VOD movie.

```
"template": "vod_service",  
"advisories": "",  
"prLevel": 4,  
"allowedEstDownloadingDevices": -1,  
"videoId": 308,  
"type": "transaction",  
"assetExternalId": "Goodfellas",  
"responseElementType": "VideoTicket",  
"attachment": {},  
"downloadPeriod": 7,  
"deliveryMode": "converged",  
"assetId": 5247,  
"id": 316023,  
"windowEnd": "",  
"videoPackageName": "",  
"rentalPeriodUnit": 3,  
"allowedViews": "",  
"windowStart": "",  
"videoPackageExternalId": "",  
"rentalPeriod": 12,  
"prName": "R",  
"videoExternalId": "Goodfellas",  
"downloadWindowEnd": 1460636490113,  
"downloadWindowStart": 1460031714846,  
"downloadPeriodUnit": 4,  
"name": "Goodfellas (es)",  
"currentEstDownloadingDevices": "",  
"status": "active"
```

https://<operator>.tvaas.com/RTEFacade/GetVideoAsset?external_id=Goodfellas&language_code=es&client=json

This call obtains additional information on the video, including image information.



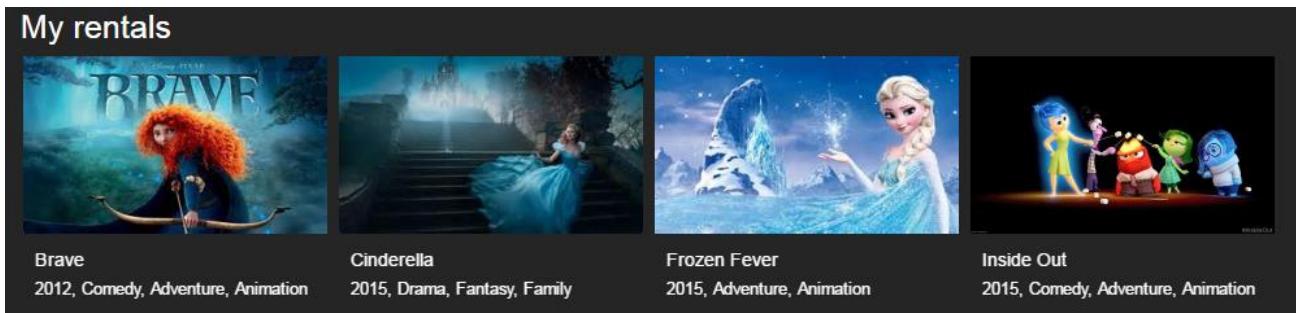
```
"metadata": {  
    "request": "GetVideoAsset",  
    "timestamp": 1466314152562  
},  
"response": {  
    "advisories": "",  
    "studio": "",  
    "prLevel": 4,  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "VerticalImage",  
            "assetName": "",  
            "value": "/attachments/goodfellas_ver.jpg" ←  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": 80006,  
            "name": "Trailer",  
            "assetName": "Test Trailer (es)",  
            "value": "true"  
        },  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "HorizontalImage",  
            "assetName": "",  
            "value": "/attachments/Goodfellas_hor.jpg" ←  
        }  
    ]  
}
```

153 List My VOD Package Movies

153.1 Overview

The movies to which the account has access through a purchased VOD Movie Package should also be listed on the My Library screen.

The movies purchased through VOD movie packages can be listed together with Rent or Buy options (as explained in [List My Movies-VOD](#)).



153.2 Requests and Responses

Request: [GetVideoTicketList](#) (from RiGHTv Front-end API)

Returns a list of movies purchased as VOD movies and as part of VOD packages. No parameters are required.

Response: [VideoTicket](#) (from RiGHTv Front-end API)

The VideoTicket entity provides various information about each ticket.

Property	Description
<i>template</i>	If set to <code>vod package movie</code> , indicates a ticket for a VOD package movie.
<i>assetExternalId</i>	The external id of the movie.
<i>name</i>	The name of the movie.
<i>status</i>	Indicates if <code>active</code> or <code>expired</code> .

To obtain the image for the movie, use `GetVideo`, as described in [34 Get Movie Metadata \(p.108\)](#).

For more information, see the RiGHTv Front-end API Developers Guide.

153.3 Related Tutorials

To	Refer to Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)

153.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetVideoTicketList?client=json>

This call requests all VOD movie and VOD movie packages to which the user has access.

An extract of a sample response is shown below. The account has access to the VOD movie package My Movies. This includes the movie Argo.

```

"template": "vod package movie",
"advisories": "",
"prLevel": 4,
"allowedEstDownloadingDevices": "",
"videoId": 9201,
"type": "transaction",
"assetExternalId": "Argo", ←
"responseElementType": "VideoTicket",
"attachment": {},
"downloadPeriod": "",
"deliveryMode": "streaming",
"assetId": 5274,
"id": 271009,
>windowEnd": 1463810472434,
"videoPackageName": "MyMovies",
"rentalPeriodUnit": 4,
"allowedViews": "",
>windowStart": 1459922472434,
"videoPackageExternalId": "3456345634",
"rentalPeriod": 45,
"prName": "R",
"videoExternalId": "f8dc1ae7-13dc-43d9-8df2-3587a082046b",
"downloadWindowEnd": "",
"downloadWindowStart": "",
"downloadPeriodUnit": 0,
"name": "Argo (es)", ←
"currentEstDownloadingDevices": "", | ←
"status": "active"

```



154 List My SVOD Subscriptions

154.1 Overview

The My Subscriptions section on the My Library screen lists the customer's subscriptions to allow them to access those services directly.



154.2 Requests and Responses

Request: [GetHouseholdServiceList](#) (from RiGHTv Front-end API)

Gets a list of the account's services. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [ServiceSubscription](#) (from RiGHTv Front-end API)

The *ServiceSubscription* entity lists all services to which the account is subscribed. The following parameters are relevant:

Parameter	Description
<i>externalId</i>	The external ID of the service.
<i>name</i>	The name of the service.
<i>model</i>	The model of service. Only services of model SVOD are relevant to the My Library screen.

154.3 Requests and Responses

To get the subscription's image to display, use *GetServiceList*.

Request: [GetServiceList](#) (from RiGHTv Front-end API)

Returns a range of information about all services in the system. No parameters are specified.

Response: [Service](#) (from RiGHTv Front-end API)

Property	Description
<i>attachments</i>	An array of image and video attachments for the service. See 149 Display Images (p.402) .

For more information, see the RiGHTv Front-end API Developers Guide.

154.4 Related Tutorials

To	Refer to Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)

154.5 Example

<https://<operator>.tvaas.com/RTEFacade/GetHouseholdServiceList?client=json>

This request gets a list of all the account's services.

In the sample response below, the account can be seen to be subscribed to an SVOD service called Disney. It's *externalId* is Disney.

```
{
  "responseElementType": "ServiceSubscription",
  "subscriptionExpiration": "",
  "externalId": "Disney", 
  "model": "SVOD",
  "id": 4,
  "serviceName": "Disney" 
},
```

<https://<operator>.tvaas.com/RTEFacade/GetServiceList?client=json>

This call obtains information on all services in the system. The response for the service Disney is shown below. The response includes attachment image information.

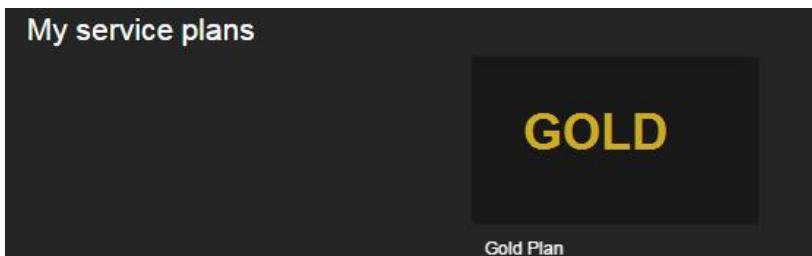


```
{  
    "responseElementType": "Service",  
    "simultaneousViewsLimit": -1,  
    "quotaSizeGB": "",  
    "attachments": [  
        {  
            "responseElementType": "Attachment",  
            "assetId": "",  
            "name": "HorizontalImage", ←  
            "assetName": "",  
            "value": "/attachments/mickey.gif"  
        }  
    ],  
    "price": 0,  
    "name": "Disney",  
    "extrafields": [],  
    "description": "Disney content",  
    "externalId": "Disney",  
    "model": "SVOD",  
    "serviceCatalogFolders": [  
        {  
            "responseElementType": "FolderReference",  
            "externalId": "Disney",  
            "id": 903  
        }  
    ],  
    "id": 4  
},|
```

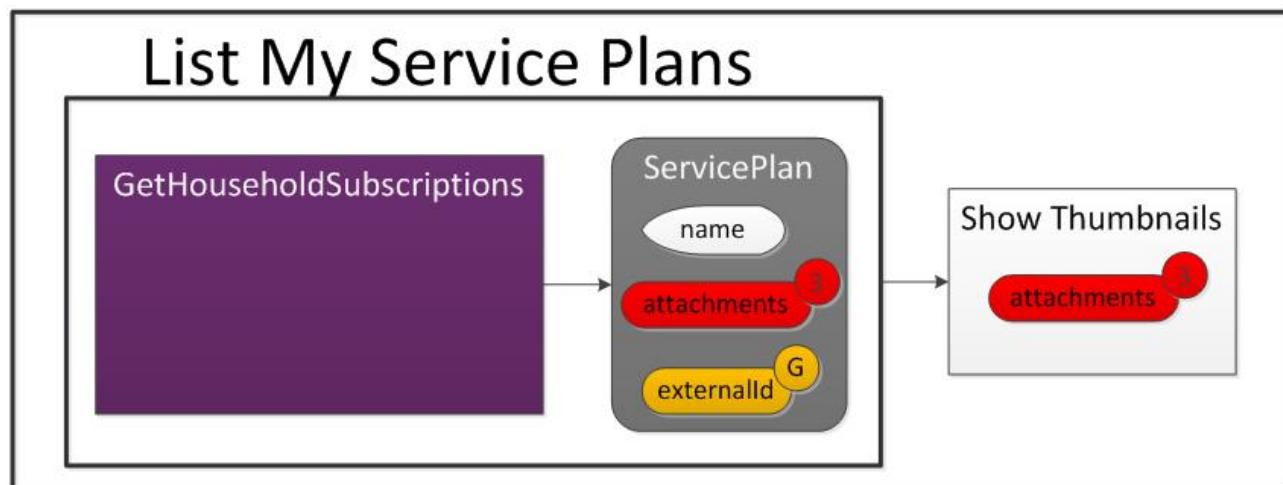
155 List My Service Plans

155.1 Overview

The My Service Plans section on the My Library screen lists the Service Plans to which the logged in account is subscribed.



155.2 Requests and Responses



Request: [GetHouseholdSubscriptions](#) (from RiGHTv Front-end API)

This call obtains a list of the account's subscriptions. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: HouseholdSubscriptions (from RiGHTv Front-end API)

This entity lists all the account's subscriptions. Significant properties include:

Property	Description										
<i>ServicePlanSubscriptionInfoList</i>	<p>Lists all the account's Service Plan subscriptions with their details, each in a <i>servicePlanSubscriptionInfo</i> response.</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>ServicePlan</i></td><td>Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:</td></tr> <tr> <td><i>externalId</i></td><td>The Service Plan external ID.</td></tr> <tr> <td><i>name</i></td><td>The Service Plan name.</td></tr> <tr> <td><i>attachments</i></td><td>The <i>name</i> and <i>value</i> (path and file name) of the Service Plan posters. See 149 Display Images (p.402) to learn how to retrieve the image files.</td></tr> </tbody> </table>	Property	Description	<i>ServicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:	<i>externalId</i>	The Service Plan external ID.	<i>name</i>	The Service Plan name.	<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Service Plan posters. See 149 Display Images (p.402) to learn how to retrieve the image files.
Property	Description										
<i>ServicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:										
<i>externalId</i>	The Service Plan external ID.										
<i>name</i>	The Service Plan name.										
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Service Plan posters. See 149 Display Images (p.402) to learn how to retrieve the image files.										



155.3 Related Tutorials

To	Refer to Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)

155.4 Example

https://<operator>.tvaas.com/RTEFacade/GetHouseholdSubscriptions?client=json&language_code=en

This request gets a list of all the account's services.

In the sample response below, the account is subscribed to a Service Plan called Gold Plan. Its *externalId* is Gold_Plan001 and costs \$15 per month.

```

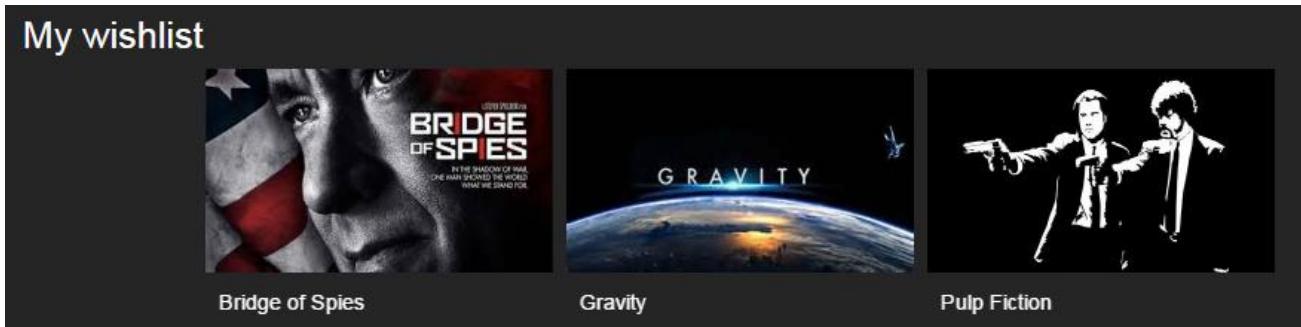
81 "servicePlanSubscriptionInfoList": [
82   {
83     "responseElementType": "ServicePlanSubscriptionInfo",
84     "servicePlan": {
85       "attachments": [],
86       "excludedFromSelfService": false,
87       "servicePlanOption": "FIXED_BY_DURATION_END_DAY",
88       "bouquets": [],
89       "externalId": "Gold_Plan001", ←
90       "description": "oscars & basic pacakge",
91       "fixedEndDate": "",
92       "services": [ ],
139     "responseElementType": "ServicePlan",
140     "removalDate": 1493556000000,
141     "activationOption": "IMMEDIATE",
142     "price": 15, ←
143     "consumptionDurationInDays": 10,
144     "channelPackages": [ ],
182     "name": "Gold Plan", ←
183     "activationDate": "",
184     "extraFields": []
185   },

```

156 Show My Wishlist

A Wishlist is a list of movies from a VOD catalog that the user selects and marks, so that later on they have direct access to either purchase or view them from a single location.

This section on the My Library screen lists the content in the customer's Wishlist.



156.1 Requests and Responses

Request: [GetProfileWishList](#) (from COMPASS RT API)

Gets the profile's wish list. No parameters are required.

Response: [WishListRecomendation](#) (from COMPASS RT API)

The returned entity lists all items in the profile's wish list. The significant properties include:

Property	Description
<i>externalContentId</i>	The external ID of the content.
<i>name</i>	The name of the content.
<i>images</i>	An array of attachment information for getting thumbnail information. See 149 Display Images (p.402) .

See the COMPASS RT API documentation for more information.

156.2 Related Tutorials

To	Refer to Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)

156.3 Example

<https://<operator>.tvaas.com/compass/GetProfileWishList?client=json>

This request calls for the list of the profile's Wish List items.

The sample response shows that the movie Cinderella is on the user's wish list. Attachment information is provided.

```
"images": [
  {
    "name": "HorizontalImage",
    "value": "/attachments/CinderellaOutHorizontalImage.jpg"
  },
  {
    "name": "VerticalImage",
    "value": "/attachments/CinderellaVerticalImage.jpg"
  }
],
"prLevel": 3,
"availabilities": [],
"blend": "Default",
"prName": "TV-14",
"ContentProperties": [],
"recommended": 1,
"addToListTime": 1456033917566,
"engine": "promotions",
"genres": [
  {
    "name": "Fantasy",
    "externalId": "Fantasy",
    "id": "Fantasy"
  },
  {
    "name": "Drama",
    "externalId": "Drama",
    "id": "Drama"
  },
  {
    "name": "Family",
    "externalId": "Family",
    "id": "Family"
  }
],
"name": "Cinderella",
"externalContentId": "Cinderella",
"id": 22301,
"contentType": "Movie"
```

157 Show My Reminders

157.1 Overview

Users should be able to set reminders to alert them before a specified program is scheduled to start. Reminders are managed entirely by the client application. The back-end does not provide support for reminders.

The My Library screen should list the user's reminders, and enable the user to remove each of them.

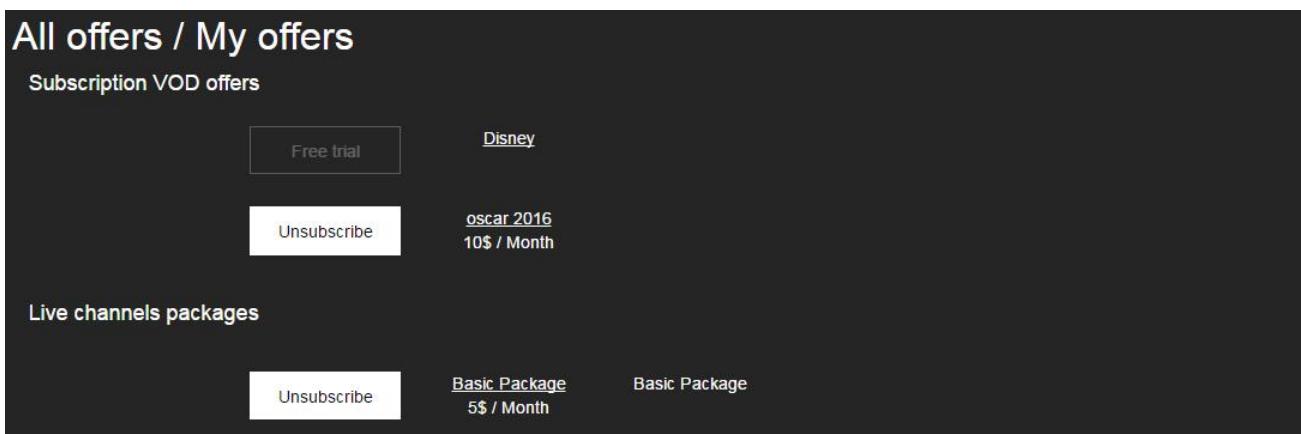
157.2 Related Tutorials

To	Refer To Tutorial
Build a My Library screen	151 Creating a My Library Screen (p.413)
Set reminders for live programs	74 Setting Program Reminders (p.218)

158 My Offers / All Offers Screen

158.1 Overview

The Service Delivery Platform enables operators to offer customers a number of subscription services. A customer pays a monthly fee for each service and continues to have access to the service while paying the subscription. A customer can choose to cancel a subscription at any time.



The following service subscriptions are available:

- Subscription VOD
- Live Channel Packages
- Service Plans
- Catch-Up services

The application's My Offers/All Offers Screen should include:

- All of an account's current subscriptions
- All other available offers

For each service displayed, the following information should be displayed:

- Name
- Description
- Price
- Expiration Date
- Service Image (for services that have images defined)
- Service Type

For each service, the user should be able to:

- Progress to the service's screen (for SVOD subscriptions only)
- Subscribe to an unsubscribed service
- Unsubscribe from a subscribed service

The sections below show how to obtain the required information for the My Offers/All Offers screen.

158.2 Related Tutorials

To	Refer To Tutorial
Get a list of all available SVOD offers, and identify those to which the account is already subscribed	159 List All Available SVOD Offers (p.431)
Get a list of all available Channel Packages, and identify those to which the account is already subscribed	160 List Channel Package Offers (p.438)
Get a list of all available Service Plans, and identify those to which the account is already subscribed	161 List All Available Service Plan Offers (p.444)
List available Catch-Up offers, and identify those to which the account is already subscribed	167 List Catch-Up Offers (p.465)

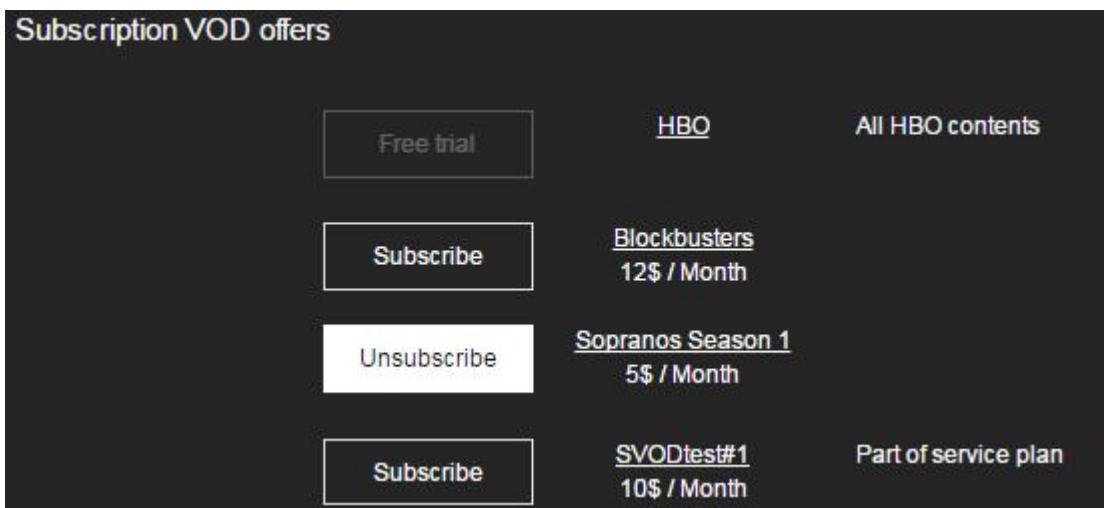
159 List All Available SVOD Offers

159.1 Overview

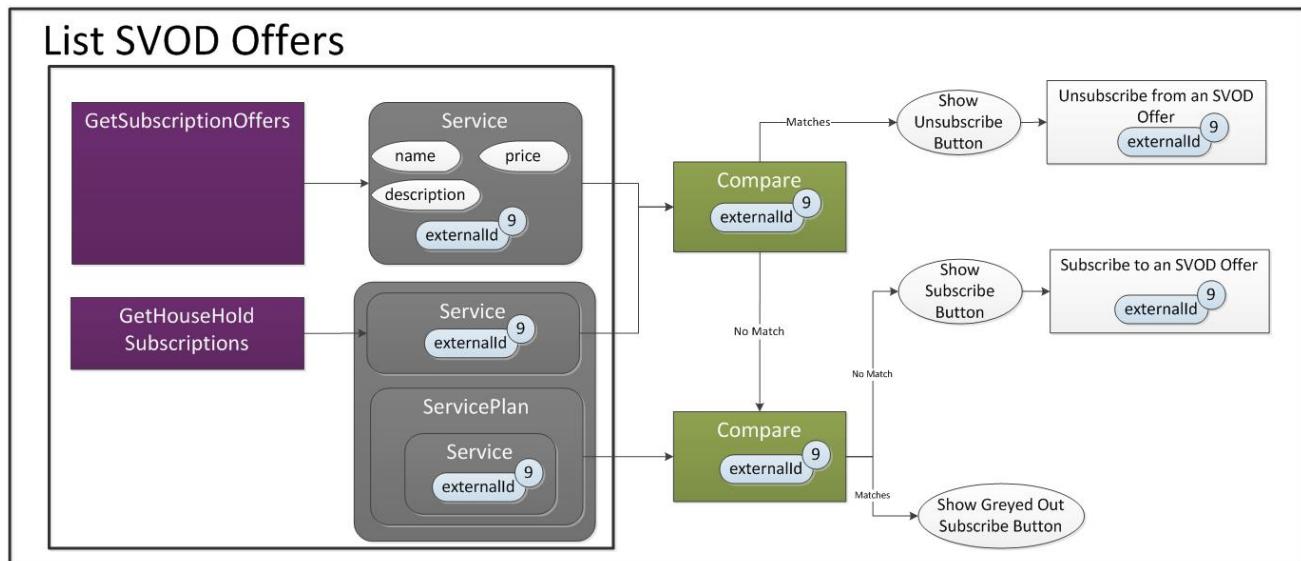
The My Offers/All Offers page must list all available SVOD offers. It must also provide the ability to subscribe to unsubscribed SVOD offers and unsubscribe from subscribed ones. The account may also be subscribed to an SVOD offer indirectly, through a Service Plan. In such a case, the Subscribe button should be greyed out.

For SVOD offers to which the account is subscribed, the scheduled subscription end date (if any) should be displayed.

Note that SVOD Offers are referred to throughout the APIs as Services.



159.2 Requests and Responses



Two action calls are required to get the information needed for SVOD offers for the My Offers/All Offers screen:

- `GetSubscriptionOffers`, which gets a list of all available subscription offers, including SVOD offers, together with additional details on each.
- `GetHouseHoldSubscriptions`, which gets a list of all subscription offers, including SVOD offers, to which the account is subscribed. Also gets a list of all Service Plans to which the account is subscribed, and the SVOD offers included in each Service Plan are also listed.

To determine whether a Subscribe, Unsubscribe, or greyed out Subscribe button should be displayed for each offered SVOD offer:

- Check whether any of the SVOD offers listed in the response to `GetHouseHoldSubscriptions` match the offered SVOD offer.

If there is a match, show an Unsubscribe button.

If there is no match:

1. Create a list of all SVOD offers that are included in all Service Plans returned by `GetHouseHoldSubscriptions`.
2. Check if any of those match the offered SVOD offer.

If there is a match, show a greyed out Subscsribe button.

If there is no match, show a Subscribe button.

159.2.1 Requests and Responses: GetSubscriptionOffers

Request: [GetSubscriptionOffers](#) (from RiGHTv Front-end API)

This call obtains a list of all available subscription offers, including SVOD offers. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: OffersInfo (from RiGHTv Front-end API)

This entity lists all available subscription offers, including SVOD offers. Significant properties include:

Property	Description	
<i>serviceList</i>	Lists all available SVOD offers with their details, in <i>Service</i> entities. See below.	
	Property	Description
	<i>externalId</i>	The SVOD offer External ID.
	<i>name</i>	The name of the SVOD offer.
	<i>description</i>	The description of the SVOD offer.
	<i>price</i>	The monthly price of the SVOD offer.

For more properties, see the RiGHTv Front-end API Developers Guide.

159.2.2 Requests and Responses: GetHouseholdSubscriptions

Request: GetHouseholdSubscriptions (from RiGHTv Front-end API)

This call obtains a list of all the SVOD offers and Service Plans to which the account is subscribed. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: HouseholdSubscriptions (from RiGHTv Front-end API)

This entity lists all subscribed SVOD offers and Service Plans.

Property	Description											
<i>serviceSubscriptionInfoList</i>	<p>Lists all the account's SVOD offer subscriptions with their details, each in a <i>ServiceSubscriptionInfo</i> response. Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>service</i></td><td>Full details of the SVOD offer (in a <i>Service</i> entity). Significant properties include:</td></tr> <tr> <td><i>externalId</i></td><td>The SVOD offer external ID.</td></tr> <tr> <td><i>expirationDate</i></td><td>The date the account's subscription to the SVOD offer is planned to end.</td></tr> <tr> <td><i>storeType</i></td><td>Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.</td></tr> </tbody> </table>		Property	Description	<i>service</i>	Full details of the SVOD offer (in a <i>Service</i> entity). Significant properties include:	<i>externalId</i>	The SVOD offer external ID.	<i>expirationDate</i>	The date the account's subscription to the SVOD offer is planned to end.	<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.
Property	Description											
<i>service</i>	Full details of the SVOD offer (in a <i>Service</i> entity). Significant properties include:											
<i>externalId</i>	The SVOD offer external ID.											
<i>expirationDate</i>	The date the account's subscription to the SVOD offer is planned to end.											
<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.											

Property	Description																	
<i>servicePlanSubscriptionInfoList</i> <i>SubscriptionInfoList</i>	<p>Lists all the account's Service Plan subscriptions with their details, each in a <i>servicePlanSubscriptionInfo</i> response. Includes a list of SVOD offers included in each Service Plan. Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>servicePlan</i></td><td>Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:</td></tr> <tr> <td><i>externalId</i></td><td>The Service Plan external ID.</td></tr> <tr> <td><i>services</i></td><td> <p>Lists all the SVOD offers included in the Service Plan, each in a Service entity. Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The SVOD offer external ID.</td></tr> </tbody> </table> </td></tr> <tr> <td><i>expirationDate</i></td><td>The date the account's subscription to the Service Plan is planned to end.</td></tr> <tr> <td><i>storeType</i></td><td>Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.</td></tr> </tbody> </table>		Property	Description	<i>servicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:	<i>externalId</i>	The Service Plan external ID.	<i>services</i>	<p>Lists all the SVOD offers included in the Service Plan, each in a Service entity. Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The SVOD offer external ID.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The SVOD offer external ID.	<i>expirationDate</i>	The date the account's subscription to the Service Plan is planned to end.	<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.
Property	Description																	
<i>servicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:																	
<i>externalId</i>	The Service Plan external ID.																	
<i>services</i>	<p>Lists all the SVOD offers included in the Service Plan, each in a Service entity. Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The SVOD offer external ID.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The SVOD offer external ID.													
Property	Description																	
<i>externalId</i>	The SVOD offer external ID.																	
<i>expirationDate</i>	The date the account's subscription to the Service Plan is planned to end.																	
<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.																	

159.3 Related Tutorials

To	Refer to Tutorial
Build a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Subscribe to an SVOD service	130 Subscribing to an SVOD Service (p.345)



Unsubscribe from an SVOD service	137 Unsubscribing from an SVOD Service (p.362)
----------------------------------	--

159.4 Example

https://<operator>-atp.tvaas.com/RTEFacade/GetSubscriptionOffers?client=json&language_code=en

This call requests a list of all subscription services available.

The following Service response is shown within OffersInfo > serviceList. Note that the externalId of the service is Oscar 2016.

```
{
  "quotaSizeGB": "",
  "attachments": [
    {
      "responseElementType": "Attachment",
      "assetId": "",
      "name": "HorizontalImage",
      "assetName": "",
      "value": "/attachments/oscar.png"
    }
  ],
  "description": "",
  "externalId": "oscar 2016",
  "serviceCatalogFolders": [
    {
      "responseElementType": "FolderReference",
      "externalId": "Oscar_SVOD",
      "id": 5305
    }
  ],
  "responseElementType": "Service",
  "simultaneousViewsLimit": "",
  "price": 10,
}
```

https://dotscreen-atp.tvaas.com/RTEFacade/GetHouseholdSubscriptions?client=json&language_code=en

This call requests a list of all subscription offers to which the account is subscribed.

The following ServiceSubscriptionInfo response is shown within HouseholdSubscriptions > serviceSubscriptionInfoList. Note that the externalId of the service is Oscar 2016. Conclude that the account does subscribe to the SVOD offer shown in the response above.

```
        ],
        "description": "",
        "externalId": "oscar_2016",
        "serviceCatalogFolders": [
            {
                "responseElementType": "FolderReference",
                "externalId": "Oscar_SVOD",
                "id": 5305
            }
        ],
        "responseElementType": "Service",
        "simultaneousViewsLimit": "",
        "price": 10,
        "servicePlanReferenceList": [
            {
                "responseElementType": "ServicePlanReference",
                "name": "Rotem",
                "externalId": "Rotem"
            },
            {
                "responseElementType": "ServicePlanReference",
                "name": "Fixed By Duration",
                "externalId": "Fixed By Duration"
            }
        ],
        "name": "oscar 2016",
        "extrafields": [],
        "model": "SVOD",
        "id": 21
    }]
```



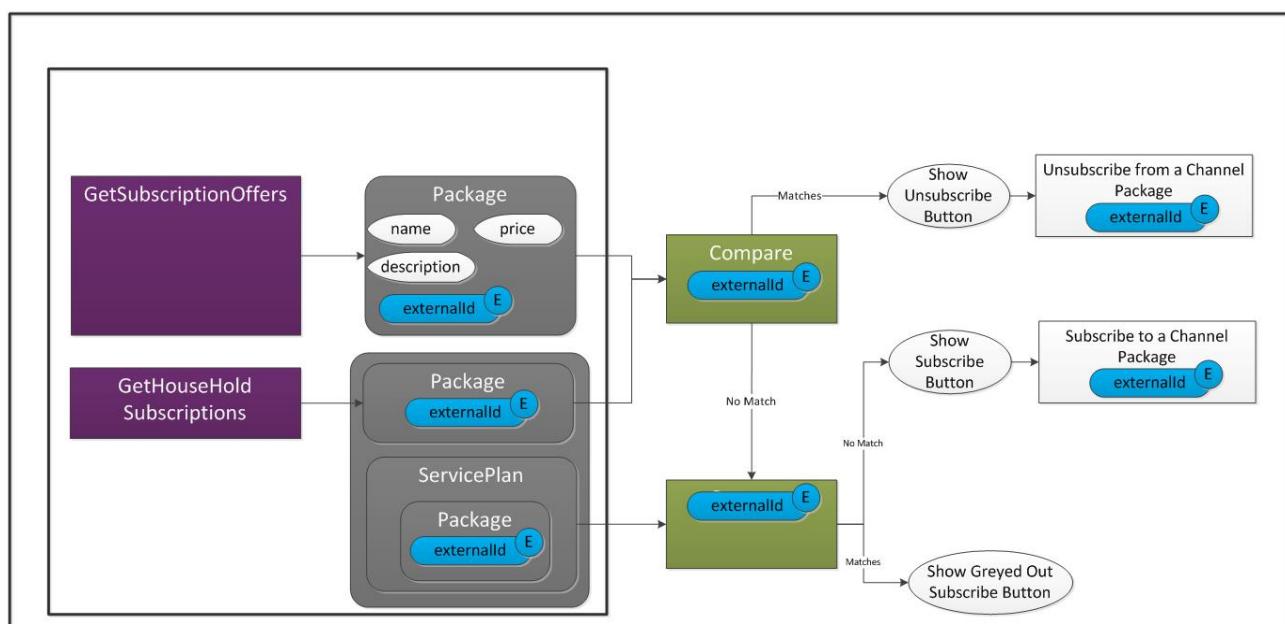
160 List Channel Package Offers

The My Offers/All Offers page must list all available Channel Packages. It must also provide the ability to subscribe to unsubscribed Channel Packages and unsubscribe from subscribed ones. The account may also be subscribed to a Channel Package indirectly, through a Service Plan. In such a case, the Subscribe button should be greyed out.

For Channel Packages to which the account is subscribed, the scheduled subscription end date (if any) should be displayed.

Live channel packages		
Part of service plan	Basic Package 5\$ / Month	Basic Package
Unsubscribe	Channel 1 1\$ / Month	
Subscribe	Channel T 10\$ / Month	Part of service plan
Subscribe	One channel package 123\$ / Month	

160.1 Requests and Responses



Two action calls are required to get the information needed for Channel Packages for the My Offers/All Offers screen:

- *GetSubscriptionOffers*, which gets a list of all available subscription offers, including Channel Packages, together with additional details on each.
- *GetHouseholdSubscriptions*, which gets a list of all subscription offers, including Channel Packages, to which the account is subscribed. Also gets a list of all Service Plans to which the account is subscribed, and the Channel Packages included in each Service Plan are also listed.

To determine whether a Subscribe, Unsubscribe, or greyed out Subscribe button should be displayed for each offered Channel Package:

- Check whether any of the Channel Packages listed in the response to *GetHouseHoldSubscriptions* match the offered Channel Package.

If there is a match, show an Unsubscribe button.

If there is no match:

1. Create a list of all Channel Packages that are included in all Service Plans returned by *GetHouseHoldSubscriptions*.
2. Check if any of those match the offered Channel Package.

If there is a match, show a greyed out Subscsribes button.

If there is no match, show a Subscribe button.

160.1.1 Requests and Responses: GetSubscriptionOffers

Request: [GetSubscriptionOffers](#) (from RiGHTv Front-end API)

This call obtains a list of all available subscription offers, including Channel Packages. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [OffersInfo](#) (from RiGHTv Front-end API)

This entity lists all available subscription offers, including Channel Packages. Significant properties include:

Property	Description	
<i>channelPackageList</i>	Lists all available Channel Packages with their details, in <i>Package</i> entities. See below.	
Property	Property	Description
	<i>externalId</i>	The Channel Package External ID.
	<i>name</i>	The name of the Channel Package.
	<i>description</i>	The description of the Channel Package.
	<i>price</i>	The monthly price of the Channel Package.
	<i>storeProductReferenceId</i>	The ID of the app store product reference.



For more properties, see the RiGHTv Front-end API Developers Guide.

160.1.2 Requests and Responses: GetHouseholdSubscriptions

Request: [GetHouseholdSubscriptions](#) (from RiGHTv Front-end API)

This call obtains a list of all the Channel Packages and Service Plans to which the account is subscribed. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [HouseholdSubscriptions](#) (from RiGHTv Front-end API)

This entity lists all subscribed Channel Packages and Service Plans.

Property	Description											
<i>channelPackageSubscriptionInfoList</i>	<p>Lists all the account's Channel Package subscriptions with their details, each in a <i>ChannelPackageSubscriptionInfo</i> response.</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>channelPackage</i></td><td>Full details on the Channel Package (in a <i>Package</i> entity). Significant properties include:</td></tr> <tr> <td><i>externalId</i></td><td>The Channel Package external ID.</td></tr> <tr> <td><i>channelPackage</i></td><td>The date the account's subscription to the Channel Package is planned to end.</td></tr> <tr> <td><i>expirationDate</i></td><td>Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.</td></tr> </tbody> </table>		Property	Description	<i>channelPackage</i>	Full details on the Channel Package (in a <i>Package</i> entity). Significant properties include:	<i>externalId</i>	The Channel Package external ID.	<i>channelPackage</i>	The date the account's subscription to the Channel Package is planned to end.	<i>expirationDate</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.
Property	Description											
<i>channelPackage</i>	Full details on the Channel Package (in a <i>Package</i> entity). Significant properties include:											
<i>externalId</i>	The Channel Package external ID.											
<i>channelPackage</i>	The date the account's subscription to the Channel Package is planned to end.											
<i>expirationDate</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.											



Property	Description													
<i>servicePlanSubscriptionInfoList</i>	<p>Lists all the account's Service Plan subscriptions with their details, each in a <i>servicePlanSubscriptionInfo</i> response. Includes a list of Channel Packages included in each Service Plan.</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>servicePlan</i></td><td>Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:</td></tr> <tr> <td><i>externalId</i></td><td>The Service Plan external ID.</td></tr> <tr> <td><i>channelPackages</i></td><td>Lists all the Channel Packages included in the Service Plan, each in a Package entity.</td></tr> <tr> <td><i>externalId</i></td><td>The Channel Package external ID.</td></tr> <tr> <td><i>expirationDate</i></td><td>The date the account's subscription to the Service Plan is planned to end.</td></tr> </tbody> </table>		Property	Description	<i>servicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:	<i>externalId</i>	The Service Plan external ID.	<i>channelPackages</i>	Lists all the Channel Packages included in the Service Plan, each in a Package entity.	<i>externalId</i>	The Channel Package external ID.	<i>expirationDate</i>	The date the account's subscription to the Service Plan is planned to end.
Property	Description													
<i>servicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:													
<i>externalId</i>	The Service Plan external ID.													
<i>channelPackages</i>	Lists all the Channel Packages included in the Service Plan, each in a Package entity.													
<i>externalId</i>	The Channel Package external ID.													
<i>expirationDate</i>	The date the account's subscription to the Service Plan is planned to end.													



Property	Description	
	<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.

160.2 Related Tutorials

To	Refer to Tutorial
Build a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Subscribe to a Channel Package	140 Subscribe to a Channel Package (p.368)
Unsubscribe from a Channel Package	141 Unsubscribe from a Channel Package (p.371)

160.3 Example

https://<operator>.tvaas.com/RTEFacade/GetSubscriptionOffers?client=json&language_code=en

This call requests a list of all subscription services available.

The following Package response is shown. Note that the externalId of the Channel Package is Trial_Pack.

```
"channelPackageList": [
  {
    "responseElementType": "Package",
    "price": 5,
    "servicePlanReferenceList": [ ],
    "name": "Basic Package",
    "extrafields": [],
    "description": "Basic Package",
    "externalId": "Trial_Pack", 
    "id": 8005
  },
]
```

https://<operator>.tvaas.com/RTEFacade/GetHouseholdSubscriptions?client=json&language_code=en



This call requests a list of all subscription offers to which the account is subscribed.

The following *ChannelPackageSubscriptionInfo* response is shown within HouseholdSubscriptions > *channelPackageSubscriptionInfoList*. Note that the externalId of the Channel Package is Trial_Pack. Conclude that the account does subscribe to the Channel Package shown in the response above.

```
  ],
  "name": "Basic Package",
  "extrafields": [],
  "description": "Basic Package",
  "externalId": "Trial_Pack",
  "id": 8005
},
"activationDate": 1473846466260,
"expirationDate": ""
```



161 List All Available Service Plan Offers

161.1 Overview

The All Offers/My Offers screen should list all available Service Plans.



The following information should be displayed for each Service Plan:

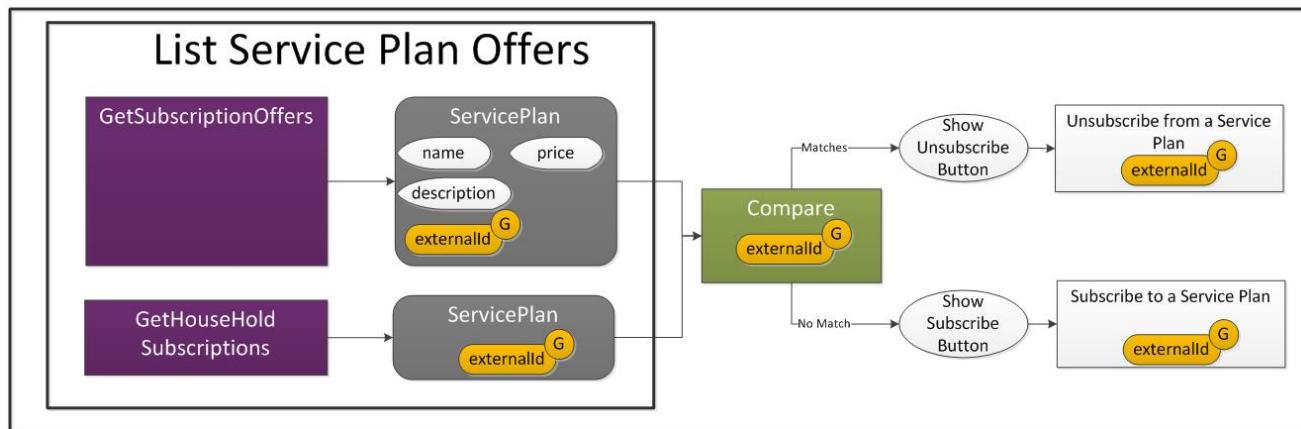
- A Subscribe/Unsubscribe button (depending on whether currently unsubscribed or subscribed)
- The Service Plan Name
- The Cost of the Service Plan
- The Description of the Service Plan

The following additional information may also be displayed:

- The Service Plan Image
- Details of the Service Plan's duration, which can take different forms:
 - Renewable: The Service Plan is renewed each month automatically, no end date is set.
 - Duration of X days, until beginning of last day: The Service Plan is available for x days after purchase, and expires at the start of the last day.
 - Duration of X days, until end of last day: The Service Plan is available for x days after purchase, and expires at the end of the last day.
 - Expires on X: The Service Plan expires at the specified date.
 - For subscribed Service Plans, the planned expiration date (if exists).

161.2 Requests and Responses

161.2.1 GetSubscriptionOffers



Two action calls are required to get the information needed for Service Plans for the My Offers/All Offers screen:

- *GetSubscriptionOffers*, which gets a list of all available subscription offers, including Service Plans. Details of each are provided.
- *GetHouseholdSubscriptions*, which gets a list of all services (including Service Plans) to which the account is subscribed.

Use *GetSubscriptionOffers* to list all Service Plans and get their details. Use *GetHouseholdSubscriptions* to determine which Service Plans are subscribed to (and thus require Unsubscribe buttons), and which are not subscribed to (and thus require Subscribe buttons).

Request: [GetSubscriptionOffers](#) (from RiGHTv Front-end API)

This call obtains a list of all available subscription offers, including Service Plans. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [OffersInfo](#) (from RiGHTv Front-end API)

This entity lists all available subscription offers, including Service Plans. Significant properties include:

Property	Description
<i>servicePlanList</i>	Lists all available Service Plans with their details, in <i>ServicePlan</i> entities. See below.

Property	Description	
	Property	Description

Property	Description
	<i>externalId</i>

Property	Description	
	<i>name</i>	The name of the Service Plan.
	<i>description</i>	The description of the Service Plan.
	<i>price</i>	The monthly price of the Service Plan.
	<i>removalDate</i>	The date from which the Service Plan can no longer be ordered.
<i>servicePlanOption</i>	Specifies how the duration of the Service Plan is defined, as follows:	
	Value	Description
	RENEWED	The Service Plan is renewed each month automatically, no end date is set.
	FIXED_BY_DURATION	The Service Plan is available for x days after purchase, and expires at the start of the last day. x is specified by <i>consumptionDurationInDays</i> .
	FIXED_BY_DURATION_END_DAY	The Service Plan is available for x days after purchase, and expires at the end of the last day. x is specified by <i>consumptionDurationInDays</i> .
	FIXED_BY_DATE	The Service Plan expires at the date specified by <i>fixedEndDate</i> .
	<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Service Plan posters. See 149 Display Images (p.402) to learn how to retrieve the image files.



161.2.2 GetHouseholdSubscriptions

Request: [GetHouseholdSubscriptions](#) (from RiGHTv Front-end API)

This call obtains a list of the account's subscriptions. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [HouseholdSubscriptions](#) (from RiGHTv Front-end API)

This entity lists all the account's subscriptions. Significant properties include:

Property	Description	
<i>servicePlanSubscriptionInfoList</i>	<p>Lists all the account's Service Plan subscriptions with their details, each in a <i>servicePlanSubscriptionInfo</i> response.</p> <p>Significant properties include:</p>	
Property	Property	Description
	<i>servicePlan</i>	Full details on the Service Plan (in a <i>ServicePlan</i> entity). Significant properties include:
Property	Property	Description
	<i>externalId</i>	The Service Plan external ID.
	<i>expirationDate</i>	The date the account's subscription to the Service Plan is planned to end.
	<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.

161.3 Related Tutorials

To	Refer to Tutorial
Build a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Subscribe to a Service Plan	131 Subscribe to a Service Plan (p.348)
Unsubscribe from a Service Plan	132 Unsubscribing from a Service Plan (p.351)

161.4 Example

https://<operator>.tvaas.com/RTEFacade/GetSubscriptionOffers?client=json&language_code=en

This call gets a list of all subscriptions offered in the system with a range of information on each.

An extract of the response for one Service Plan is shown below. Note that the *externalId* of the Service Plan is Gold_Plan001.

```
"servicePlanList": [
  {
    "attachments": [],
    "excludedFromSelfService": false,
    "servicePlanOption": "FIXED_BY_DURATION_END_DAY",
    "bouquets": [],
    "externalId": "Gold_Plan001", 
    "description": "The perfect plan for the whole family",
    "fixedEndDate": " "
  }
]
```

https://<operator>.tvaas.com/RTEFacade/GetHouseholdSubscriptions?client=json&language_code=en

This call gets a list of all the account's subscriptions.

A sample response is shown below. Note that the *externalId* of the Service Plan is Gold_Plan001, indicating that the Service Plan shown above is subscribed to by this account.

```
"responseElementType": "ServicePlanSubscriptionInfo", ←  
"servicePlan": {  
    "attachments": [],  
    "excludedFromSelfService": false,  
    "servicePlanOption": "FIXED_BY_DURATION_END_DAY",  
    "bouquets": [],  
    "externalId": "Gold_Plan001", ←  
    "description": "The perfect plan for the whole family",  
    "fixedEndDate": "",  
    "....."
```

162 Implementing Start Over

162.1 Overview

The Start Over service allows the subscriber to view a program that is currently being broadcast from its beginning.

To implement Start Over it is necessary to:

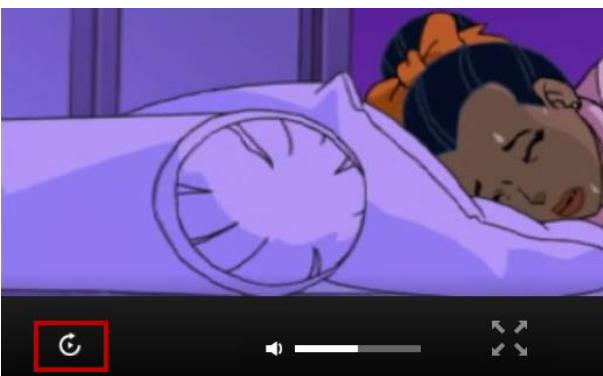
- Indicate whether a program supports Start Over and enable the user to launch Start Over viewing. See [163 Identify Start Over Enabled Programs \(p.454\)](#).
- Obtain the playing URL for the Start Over stream. See [164 Playing a Live Channel through Start Over \(p.458\)](#).
- Provide player controls suitable for viewing a Start Over stream. See [165 Guidelines for Start Over Player Controls \(p.462\)](#).



163 Identify Start Over Enabled Programs

163.1 Overview

If a program supports Start Over, that should be indicated on a Live screen on which the channels are played, on the TV Guide screen on which programs are listed, and on the Program screen itself.

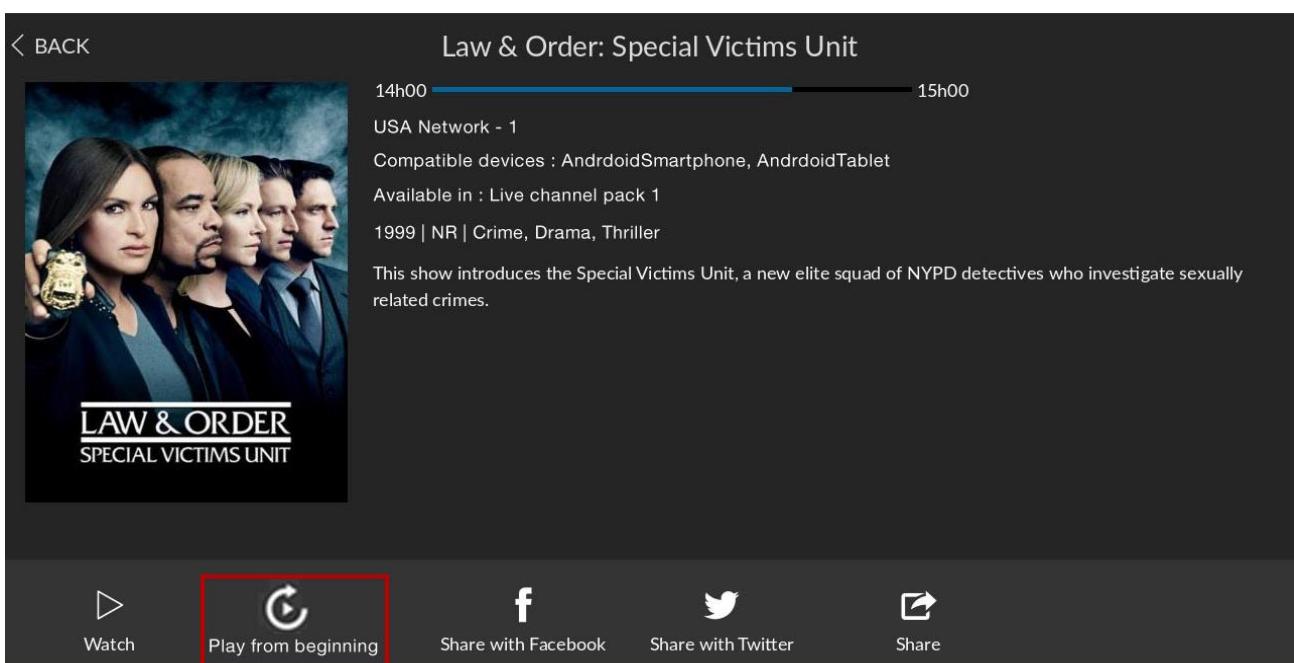
	
On the Live screen	On the Program screen

While the program is being broadcast, a button to launch a Start Over stream should be displayed, providing that Start Over is authorized for the program. For example, on the Program screen of a program currently in progress, both *Play* and a *Play from Beginning* buttons should be displayed.

Note

The option to launch a Start Over stream should not be displayed outside of the program's planned broadcast time.

In the image below, the program is currently in progress and thus the *Play from beginning* button is shown.



Two different requests are provided for determining whether a program supports Start Over or not:

- To get information for a single program, use *GetExtendedProgram*. This is most useful when creating a Program screen.
- To get information on multiple programs, use *GetProgramList*. This is most useful when creating a Now Playing or EPG screen. Within the Now Playing screen, the Start Over information should be read for the program scheduled to be played at the current time.

163.2 Requests and Responses: GetExtendedProgram

Request: [GetExtendedProgram](#) (from RiGHTv Front-end API)

This call obtains detailed information on a specified program.

The table below presents the most commonly used parameters:

Parameter	Description
program_external_id	The external ID of the particular program.

Response: [ExtendedProgram](#) (from RiGHTv Front-end API)

This entity lists detailed information on the specified program. For verifying whether Start Over is supported, the relevant property is:

Property	Description
flags	A bitwise description of the program properties, where <i>isStartUpEnabled</i> is indicated by bit (8).

For more properties, see the [RiGHTv Front-end API Developers Guide](#).

163.3 Requests and Responses: GetProgramList

Request: [GetProgramLists](#) (from RiGHTv Front-end API)

Gets a list of all programs scheduled for the specified channel(s), together with a range of information on each one. Significant parameters include:

Property	Description
channel_external_ids	A list of the ID's of all channels for which program info is required.
start_date	The earliest date from which programs should be returned. Specified in milliseconds from 1/1/1970).
end_date	The latest date from which programs should be returned. Specified in milliseconds from 1/1/1970).

Note

The start date must always start at the beginning of a day, and the end date must be the end of the day.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Program](#) (from RiGHTv Front-end API)

Lists all programs that match the parameters specified by *GetProgramLists* action, and provides a range of information on each program listed, including whether Start Over is supported or not. For verifying whether Start Over is supported, the relevant property is:

Property	Description
flags	A bitwise description of the program properties, where <i>isStartOverEnabled</i> is indicated by bit (8).
startDate, endDate	The start and end dates of the program, specified in milliseconds from 1/1/1970.

For more properties, see the RiGHTv Front-end API Developers Guide.



163.4 Related Tutorials

To	Refer to Tutorial
Learn about implementing Start Over functionality	162 Implementing Start Over (p.453)
Create a TV Guide screen	63 Create a TV Guide (p.188)
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

163.5 Example

https://<operator>.tvaas.com /RTEFacade/GetExtendedProgram?program_external_id=q3u89um5oiarp6s3n88ivnpqvo&language_code=en&client=json

This call gets information on a specific program.

As the *flags* value is 9, and Start Over support is indicated by (8), this program does support Start Over.

```
],
"endDate": 1486566000000,
"year": 1999,
"contentId": "",
"flags": 9,
```



164 Playing a Live Channel through Start Over

164.1 Overview

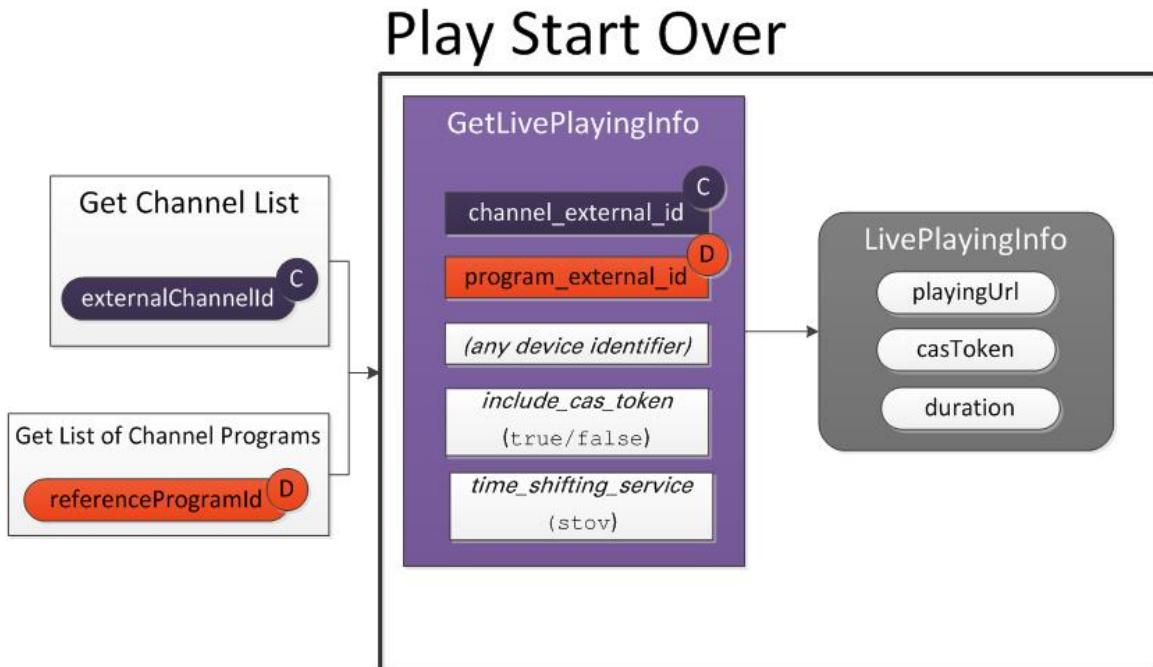
When a user attempts to launch a Start Over viewing session of a live channel, the relevant playback URL must be obtained.

The following verifications must be performed before playing the channel:

- Check that the account is subscribed to the channel. See [66 Identify Subscribed Channels \(p.197\)](#). If the account is not subscribed to the channel, within the web app the user should be provided with valid purchase options.
- Check that the parental rating of the program does not exceed the accounts' limit. See [185 Parental Control \(p.509\)](#).

This tutorial explains how to obtain the Start Over playback URL. For design guidelines for the player controls, see [165 Guidelines for Start Over Player Controls \(p.462\)](#).

164.2 Requests and Responses



Request: [GetLivePlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for playing the channel through Start Over. This request requires that the user be logged in. The following parameters are required:

Parameter	Description
channel_external_id	The ID of the channel to play.
One of: <i>mac_address</i> <i>serial_number</i> <i>smart_card_id</i>	One the device (terminal) identifier for the requesting device. <i>serial_number</i> is recommended.
include_cas_token	If set to <code>true</code> requests that the CAS token should be calculated and included in the result. The default value is <code>false</code> . A CAS token is required if a license is needed to play the channel.
program_external_id	The ID of the program to be played.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [LivePlayingInfo](#) (from RiGHTv Front-end API)

Provides the playback URL for the requested channel, and if requested, the CAS token required for viewing licensed content. The program's duration is also provided.

Property	Description
playingUrl	The playback URL.
casToken	The CAS token, if requested.
duration	The duration of the program in milliseconds. The duration provided takes into account the lag and lead time set in the system configuration, thus ensuring that the entire program is included, even if it started somewhat earlier or later than planned. The duration is needed for the suggested Player implementation, see 165 Guidelines for Start Over Player Controls (p.462) .

For more properties, see the RiGHTv Front-end API Developers Guide.



164.3 Related Tutorials

To	Refer to Tutorial
Learn about implementing Start Over functionality	162 Implementing Start Over (p.453)
Create a TV Guide screen	63 Create a TV Guide (p.188)
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)
Modify the Player controls to support Start Over	165 Guidelines for Start Over Player Controls (p.462)

164.4 Example

https://<operator>.tvaas.com /RTEFacade/GetLivePlayingInfo?program_external_id=aj2nm79fgepsvm5f04p580ru4s&channel_external_id=Tvaas-MSMA-TTML-Arte&serial_number=MOZILLA_214CA0CB-4151-2021-0FD3-6E1B615987FE&include_cas_token=true&time_shifting_service=stov&client=json&language_code=en

This call requests the Start Over playing info for the specified program. The response is shown below.

```
{  
    "metadata": {  
        "request": "GetLivePlayingInfo",  
        "timestamp": 1486884803683  
    },  
    "response": {  
        "duration": 1800000,  
        "responseElementType": "LivePlayingInfo",  
        "casToken": "QvXlrjMKS+iNTS0m12aVwo/7S//crAt0QFqWj0woEWrYHm86X1SPSGJpCMYD  
            +qyfGl7nZ6Ufw3oGCDaEw3ZBnk  
            +Dq52tzc23cYYw83dpBVWkTC2oCMvSEeKBDseTFuwvqdj1X7DDqXQgn18AgW0k6ymDmvbz  
            +n4h4jfCS4oORH/5BZMX0hXPKLvoif0q9KB",  
        "parentalRatings": [],  
        "resumePosition": "",  
        "channelId": 8016,  
        "playingUrl": "http://harmonic2.e2e.purpledrm.com.edgesuite.net/Content/SS  
            /StartOver/Channel(name=Tvaas-MSMA-TTML-Arte,startTime=148688460000000000  
            ,endTime=1486886400000000).isml/Manifest",  
        "casTokenExpiration": 1486884923670,  
        "urlExpiration": 1486884803670  
    }  
}
```

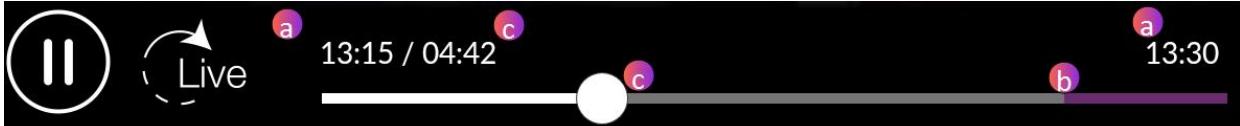


165 Guidelines for Start Over Player Controls

165.1 Overview

This tutorial provides general guidelines for the Player controls when playing a channel through Start Over.

1. For instructions on how to obtain the playback URL, see [164 Playing a Live Channel through Start Over \(p.458\)](#).
2. The Player should display relevant information, as indicated below.



3. The Player should provide required functionality, as indicated below.
A screenshot of a player interface. It includes a play/pause button with two vertical bars, a 'Live' button with a curved arrow, and a progress bar. The progress bar has a white marker and three colored circles (purple, red, and blue) above it. The text '13:15 / 04:42' is shown above the bar, and '13:30' is shown to the right. Points 'a', 'b', and 'c' are marked above the circles on the progress bar.
 - a. Play/pause: Unlike when playing live content, the user should be able to pause (and then play) the stream when viewing in Start Over mode.
 - b. Go to Live: The user should be able to leave Start Over mode and return to the live stream of the channel. See [76 Playing Live Programs \(p.220\)](#).
 - c. Interactive Progress Bar: The user should be able to drag the current position marker and/or tap any position within the progress bar to jump to any position within the Start Over stream. If the user selects a position that is ahead of the live stream itself, the marker should jump to the current position of the live stream.
4. Once the Start Over stream has been viewed until the end of the program, a pop-up message should display, offering the user the option to switch back to regular live viewing.

5. Channel zap events should be registered in the following cases:

- After the Start Over stream has been viewed for a configurable amount of time.
- When the user switches back to the live stream.

See [201 Register Event: Channel Zap \(p.540\)](#).

165.2 Related Tutorials

To	Refer to Tutorial
Learn about implementing Start Over functionality	162 Implementing Start Over (p.453)
Play a live program	76 Playing Live Programs (p.220)
Play a live program in Start Over mode	164 Playing a Live Channel through Start Over (p.458)
Register a channel zap event	201 Register Event: Channel Zap (p.540)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)



166 Implementing Catch-Up

166.1 Overview

The Catch-Up service allows the subscriber to view a program that has already been broadcast.

To implement Catch-Up it is necessary to:

- List available Catch-Up services on the My Offers/All Offers screen. See [167 List Catch-Up Offers \(p.465\)](#).
- Enable customers to subscribe to Catch-Up services. See [168 Subscribe to a Catch-Up Service \(p.471\)](#).
- Enable customers to unsubscribe from Catch-Up services. See [169 Unsubscribing from a Catch-Up Service \(p.474\)](#).
- List Catch-Up services to which the account is subscribed in the My Library screen. See [172 List My Catch-Up Services \(p.479\)](#).
- Indicate whether a program supports Catch-Up and enable the user to launch Catch-Up viewing. See [173 Identify Catch-Up Enabled Programs \(p.483\)](#).
- Indicate until when catch-up on a completed program will be available. See [174 Determining the Catch-Up Availability Period \(p.487\)](#).
- Check whether the account is subscribed to a Catch-Up service. See [175 Checking if Account is Subscribed to Catch-Up \(p.490\)](#).
- Obtain the playing URL for the Catch-Up stream. See [176 Playing a Program through Catch-Up \(p.492\)](#). The program should only be played if the account is subscribed to the relevant channel (see [66 Identify Subscribed Channels \(p.197\)](#)) and if the account is subscribed to a Catch-Up service (see [175 Checking if Account is Subscribed to Catch-Up \(p.490\)](#)). Parental control must also be taken into account, see [185 Parental Control \(p.509\)](#).
- Provide player controls suitable for viewing a Catch-Up stream. See [178 Guidelines for Catch-Up Player Controls \(p.498\)](#).



167 List Catch-Up Offers

167.1 Overview

The All Offers/My Offers screen should list all available Catch-Up offers.



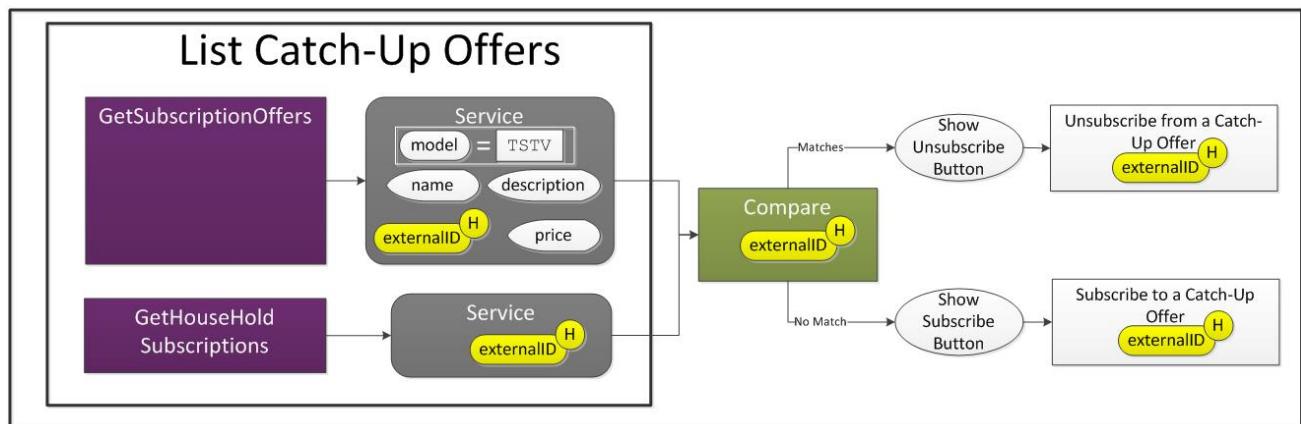
The following information should be displayed for each Catch-Up offer:

- A Subscribe/Unsubscribe button (depending on whether currently unsubscribed or subscribed)
- The Catch-Up service name
- The Cost of the Catch-Up service
- The Description of the Catch-Up service

The following additional information may also be displayed:

- For subscribed Catch-Up services, the planned expiration date (if exists).

167.2 Requests and Responses



Two action calls are required to get the information needed for Catch-Up services for the My Offers/All Offers screen:

- *GetSubscriptionOffers*, which gets a list of all available subscription offers, including Catch-Up services. Details of each are provided.
- *GetHouseHoldSubscriptions*, which gets a list of all services (including Catch-Up services) to which the account is subscribed.

Use *GetSubscriptionOffers* to list all Catch-Up services and get their details. Use

GetHouseHoldSubscriptions to determine which Catch-Up services are subscribed to (and thus require Unsubscribe buttons), and which are not subscribed to (and thus require Subscribe buttons).

167.2.1 GetSubscriptionOffers

Request: [GetSubscriptionOffers](#) (from RiGHTv Front-end API)

This call obtains a list of all available subscription offers, including Catch-Up services. No parameters are required, unless only listing offers available through a mobile app store:.

Parameter	Description
<code>store_id</code>	The ID of the app store for in-app purchasing.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [OffersInfo](#) (from RiGHTv Front-end API)

This entity lists all available subscription offers, including Catch-Up services. Significant properties include:

Property	Description																
<i>serviceList</i>	<p>Lists all available services with their details, in <i>Service</i> entities. See below.</p> <table border="1" data-bbox="512 399 1432 1410"> <thead> <tr> <th data-bbox="520 411 878 480">Property</th><th data-bbox="878 411 1432 480">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="520 480 878 550"><i>externalId</i></td><td data-bbox="878 480 1432 550">The Catch-Up service external ID.</td></tr> <tr> <td data-bbox="520 550 878 619"><i>name</i></td><td data-bbox="878 550 1432 619">The name of the Catch-Up service.</td></tr> <tr> <td data-bbox="520 619 878 781"><i>description</i></td><td data-bbox="878 619 1432 781">The description of the Catch-Up service.</td></tr> <tr> <td data-bbox="520 781 878 920"><i>price</i></td><td data-bbox="878 781 1432 920">The monthly price of the Catch-Up service.</td></tr> <tr> <td data-bbox="520 920 878 1060"><i>model</i></td><td data-bbox="878 920 1432 1060">Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td data-bbox="520 1060 878 1275"><i>attachments</i></td><td data-bbox="878 1060 1432 1275">The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.</td></tr> <tr> <td data-bbox="520 1275 878 1399"><i>storeProductReferenceId</i></td><td data-bbox="878 1275 1432 1399">The ID of the app store product reference.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The Catch-Up service external ID.	<i>name</i>	The name of the Catch-Up service.	<i>description</i>	The description of the Catch-Up service.	<i>price</i>	The monthly price of the Catch-Up service.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.	<i>storeProductReferenceId</i>	The ID of the app store product reference.
Property	Description																
<i>externalId</i>	The Catch-Up service external ID.																
<i>name</i>	The name of the Catch-Up service.																
<i>description</i>	The description of the Catch-Up service.																
<i>price</i>	The monthly price of the Catch-Up service.																
<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.																
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.																
<i>storeProductReferenceId</i>	The ID of the app store product reference.																

167.2.2 GetHouseholdSubscriptions

Request: [GetHouseholdSubscriptions](#) (from RiGHTv Front-end API)

This call obtains a list of the account's subscriptions. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.



Response: HouseholdSubscriptions (from RiGHTv Front-end API)

This entity lists all the account's subscriptions. Significant properties include:

Property	Description														
<i>s erviceSubscriptionInfoList</i>	<p>Lists all the account's service subscriptions with their details, each in a <i>ServiceSubscriptionInfo</i> response.</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>service</i></td><td> <p>Full details on the service (in a <i>Service</i> entity).</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> </tbody> </table> </td></tr> <tr> <td><i>expirationDate</i></td><td>The date the account's subscription to the Catch-Up service is planned to end (if any).</td></tr> <tr> <td><i>storeType</i></td><td>Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.</td></tr> </tbody> </table>	Property	Description	<i>service</i>	<p>Full details on the service (in a <i>Service</i> entity).</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The service external ID.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>expirationDate</i>	The date the account's subscription to the Catch-Up service is planned to end (if any).	<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.
Property	Description														
<i>service</i>	<p>Full details on the service (in a <i>Service</i> entity).</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The service external ID.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.								
Property	Description														
<i>externalId</i>	The service external ID.														
<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.														
<i>expirationDate</i>	The date the account's subscription to the Catch-Up service is planned to end (if any).														
<i>storeType</i>	Indicates the type of app store through which the account subscribed. For example, APPLE or GOOGLE for Apple and Android devices respectively. Will be empty if subscribed through PC.														



167.3 Related Tutorials

To	Refer To Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Build a My Offers/All Offers screen	158 My Offers / All Offers Screen (p.429)
Subscribe to a Catch-Up service	168 Subscribe to a Catch-Up Service (p.471)
Unsubscribe from a Catch-Up service	169 Unsubscribing from a Catch-Up Service (p.474)

167.4 Example

https://<operator>.tvaas.com/RTEFacade/GetSubscriptionOffers?client=json&language_code=en

This call gets a list of all subscriptions offered in the system with a range of information on each.

An extract of the response for one Catch-Up service is shown below. Note that the *externalId* of the Catch-Up service is 1.

```
{
  "quotaSizeGB": "",
  "attachments": [],
  "description": "",
  "externalId": "1",
  "serviceCatalogFolders": [],
  "responseElementType": "Service",
  "simultaneousViewsLimit": "",
  "price": 1,
  "servicePlanReferenceList": [],
  "name": "Catch-Up",
  "extrafields": [],
  "model": "TSTV",
  "id": 81
},
```

https://<operator>.tvaas.com/RTEFacade/GetHouseholdSubscriptions?client=json&language_code=en

This call gets a list of all the account's subscriptions.



A sample response is shown below. Note that the *externalId* of the Catch-Up service is 1, indicating that the Catch-Up service shown above is subscribed to by this account.

```
{  
    "responseElementType": "ServiceSubscriptionInfo",  
    "service": {  
        "quotaSizeGB": "",  
        "attachments": [],  
        "description": "",  
        "externalId": "1",  
        "serviceCatalogFolders": [],  
        "responseElementType": "Service",  
        "simultaneousViewsLimit": "",  
        "price": 1,  
        "servicePlanReferenceList": [],  
        "name": "Catch-Up",  
        "extrafields": [],  
        "model": "TSTV",  
        "id": 81  
    },  
    "activationDate": 1486557480129,  
    "expirationDate": ""
```



168 Subscribe to a Catch-Up Service

168.1 Overview

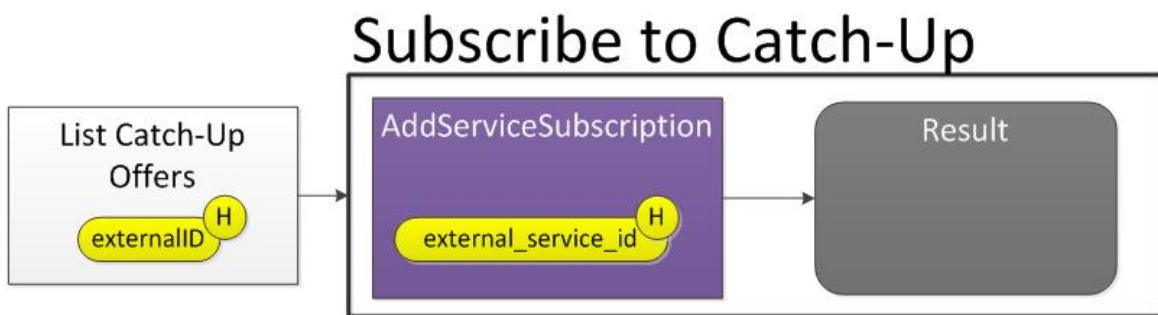
A customer can subscribe to a Catch-Up service via the PC portal. The customer is billed periodically through the online payment system.

If the recurring payment fails, the online payment system will notify RiGHTv and the customer is removed from the service.

Catch-Up subscription can be done from PC apps only.

Customers can also unsubscribe from a Catch-Up service, but only from a PC app.

168.2 Requests and Responses



Request: [AddServiceSubscription](#) (from RiGHTv Front-end API)

This request subscribes the account to the specified Catch-Up service. The following parameters are required:

Parameter	Description
<code>external_service_id</code>	The external ID of the Catch-Up service.
For In-App Purchasing Only:	
<code>store_id</code>	The ID of the app store through which the catch-up service is being subscribed.

Parameter	Description
<i>purchase_reference</i>	The reference to a purchase made for the specified catch-up service at the app store. Use the POST method to pass this parameter. (Mandatory if <i>store_id</i> is specified)
<i>store_product_reference</i>	The reference assigned to the catch-up service purchased at a particular app store. Relevant for the Google App store only.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a <code>SUCCESS</code> or <code>FAILURE</code> .
<i>message</i>	A short message from the server

168.3 Related Tutorials

To	Refer To Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
List Catch-Up offers in a My Offers/All Offers screen	167 List Catch-Up Offers (p.465)

168.4 Example

https://<operator>.tvaas.com/RTEFacade/AddPackageSubscription?external_id=Trial%20Package%20&client=json

This request subscribes the account to the channel package with external ID Trial Package 2.

The response below indicates successful subscription to the channel package.



```
[{"metadata": {  
    "request": "AddPackageSubscription",  
    "timestamp": 1460461967144  
},  
"response": {  
    "message": "'Add Package Subscription' operation finished successfully",  
    "status": "SUCCESS"  
}  
}
```

169 Unsubscribing from a Catch-Up Service

Two APIs exist for unsubscribing accounts from a Catch-Up service:

- [170 Finish Catch-Up Subscription \(p.475\)](#)
For Online accounts. The customer will continue to have access to the Catch-Up service until the next planned billing date, at which time the Catch-Up service will terminate and no charge will be levied.
- [171 Remove Catch-Up Subscription \(p.477\)](#)
For operator accounts. The customer loses access to the Catch-Up service immediately, and no further charges are levied. Any prepayment is forfeited.

For identifying the accounts type, see [118 Identify Account Type \(p.308\)](#).

169.1 Related Tutorials

To	Refer To Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
List Catch-Up offers in a My Offers/All Offers screen	167 List Catch-Up Offers (p.465)
Identify whether an account is an online or traditional account	118 Identify Account Type (p.308)
Unsubscribe an online account from a Catch-Up service	170 Finish Catch-Up Subscription (p.475)
Unsubscribe a traditional account from a Catch-Up service	171 Remove Catch-Up Subscription (p.477)



170 Finish Catch-Up Subscription

170.1 Overview

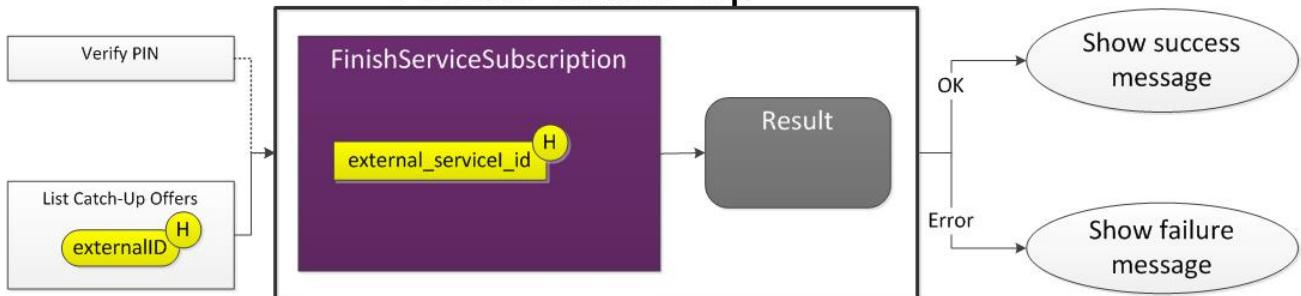
Online accounts can unsubscribe from a Catch-Up service. Unsubscribing using *FinishServicePlan Subscription* sets the end date of the Catch-Up service to the next planned debit date. The account will have access to the Catch-Up service until that date.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

170.2 Requests and Responses

Finish Catch-Up



Request: [FinishServiceSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified Catch-Up service. The following parameters are required:

Parameter	Description
<i>external_service_id</i>	The Catch-Up service external ID.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

170.3 Related Tutorials

To	Refer To Tutorial
Learn about unsubscribing accounts from a Catch-Up service	169 Unsubscribing from a Catch-Up Service (p.474)

170.4 Example

https://<operator>.tvaas.com/RTEFacade/FinishServiceSubscription?external_service_id=1&client=json

This request ends the account's subscription to the Catch-Up service with external ID 1.

The response below indicates successful ending of subscription to the Catch-Up service.

```
[{"metadata": {  
    "request": "FinishServiceSubscription",  
    "timestamp": 1489399589973  
},  
 "response": {  
    "message": "'Finish Service Subscription' operation finished successfully",  
    "status": "SUCCESS"  
}  
}
```

171 Remove Catch-Up Subscription

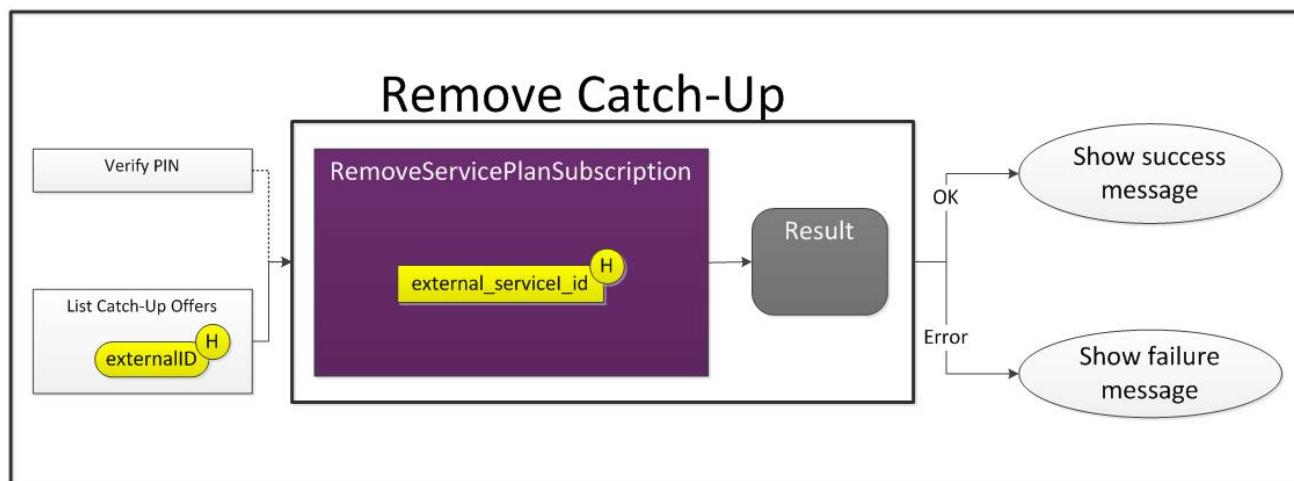
171.1 Overview

Removing Catch-Up services is intended for use with operator accounts. Unsubscribing using *RemoveServicePlanSubscription* ends the Catch-Up service immediately.

Note

Subscription termination for services subscribed to through a mobile app store is described in [In-App Purchasing](#).

171.2 Requests and Responses



Request: [RemoveServiceSubscription](#) (from RiGHTv Front-end API)

This request ends a user's subscription to the specified Catch-Up service. The following parameters are required:

Parameter	Description
<i>external_service_id</i>	The Catch-Up service external ID

Response: Result (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

171.3 Related Tutorials

To	Refer to Tutorial
Learn about unsubscribing accounts from a Catch-Up service	169 Unsubscribing from a Catch-Up Service (p.474)

171.4 Example

https://<operator>.tvaas.com/RTEFacade/RemoveServiceSubscription?external_service_id=1&client=json

This request ends the account's subscription to the Catch-Up service with external ID 1.

The response below indicates successful ending of subscription to the Service Plan.

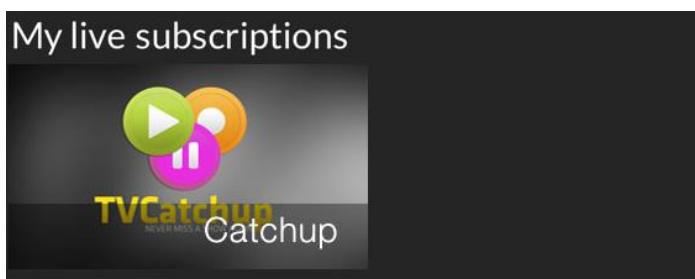
```
{
  "metadata": {
    "request": "RemoveServiceSubscription",
    "timestamp": 1488111294907
  },
  "response": {
    "message": "'Remove Service Subscription' operation finished successfully",
    "status": "SUCCESS"
  }
}
```



172 List My Catch-Up Services

172.1 Overview

The My Live Subscriptions section on the My Library screen should list the Catch-Up services to which the account is subscribed.



172.2 Requests and Responses

Request: [GetHouseholdSubscriptions](#) (from RiGHTv Front-end API)

This call obtains a list of the account's subscriptions. No parameters are required.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [HouseholdSubscriptions](#) (from RiGHTv Front-end API)

This entity lists all the account's subscriptions. Significant properties include:

Property	Description

<p><i>serviceSubscriptionInfoList</i></p>	<p>Lists all the account's service subscriptions with their details, each in a <i>ServiceSubscriptionInfo</i> response.</p> <p>Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>service</i></td><td> <p>Full details on the service (in a <i>Service</i> entity). Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>name</i></td><td>The name of the Catch-Up service.</td></tr> <tr> <td><i>description</i></td><td>The description of the Catch-Up service.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>attachments</i></td><td>The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.</td></tr> </tbody> </table> </td></tr> <tr> <td><i>expirationDate</i></td><td>The date the account's subscription to the Catch-Up service is planned to end (if any).</td></tr> </tbody> </table>	Property	Description	<i>service</i>	<p>Full details on the service (in a <i>Service</i> entity). Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>name</i></td><td>The name of the Catch-Up service.</td></tr> <tr> <td><i>description</i></td><td>The description of the Catch-Up service.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>attachments</i></td><td>The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The service external ID.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>name</i>	The name of the Catch-Up service.	<i>description</i>	The description of the Catch-Up service.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.	<i>expirationDate</i>	The date the account's subscription to the Catch-Up service is planned to end (if any).
Property	Description																				
<i>service</i>	<p>Full details on the service (in a <i>Service</i> entity). Significant properties include:</p> <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>externalId</i></td><td>The service external ID.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>name</i></td><td>The name of the Catch-Up service.</td></tr> <tr> <td><i>description</i></td><td>The description of the Catch-Up service.</td></tr> <tr> <td><i>model</i></td><td>Indicates the type of service. Returns TSTV for Catch-Up services.</td></tr> <tr> <td><i>attachments</i></td><td>The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.</td></tr> </tbody> </table>	Property	Description	<i>externalId</i>	The service external ID.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>name</i>	The name of the Catch-Up service.	<i>description</i>	The description of the Catch-Up service.	<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.	<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.						
Property	Description																				
<i>externalId</i>	The service external ID.																				
<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.																				
<i>name</i>	The name of the Catch-Up service.																				
<i>description</i>	The description of the Catch-Up service.																				
<i>model</i>	Indicates the type of service. Returns TSTV for Catch-Up services.																				
<i>attachments</i>	The <i>name</i> and <i>value</i> (path and file name) of the Catch-Up service posters. See 149 Display Images (p.402) to learn how to retrieve the image files.																				
<i>expirationDate</i>	The date the account's subscription to the Catch-Up service is planned to end (if any).																				



172.3 Related Tutorials

To	Refer to Tutorial
Create a My Library screen	151 Creating a My Library Screen (p.413)
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)

172.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHouseholdServiceList?client=json>

This call gets a list of all the account's subscriptions.

A sample response is shown below. Note that the model values TSTV indicates a Catch-Up service.

```
{
  "responseElementType": "ServiceSubscriptionInfo",
  "service": {
    "quotaSizeGB": "",
    "attachments": [],
    "description": "",
    "externalId": "1",
    "serviceCatalogFolders": [],
    "responseElementType": "Service",
    "simultaneousViewsLimit": "",
    "price": 1,
    "servicePlanReferenceList": [],
    "name": "Catch-Up",
    "extrafields": [],
    "model": "TSTV", 
    "id": 81
  },
  "activationDate": 1486557480129,
  "expirationDate": ""
}
```

173 Identify Catch-Up Enabled Programs

173.1 Overview

If a program supports Catch-Up, that should be indicated on the TV Guide screen on which programs are listed, and on the Program screen itself.

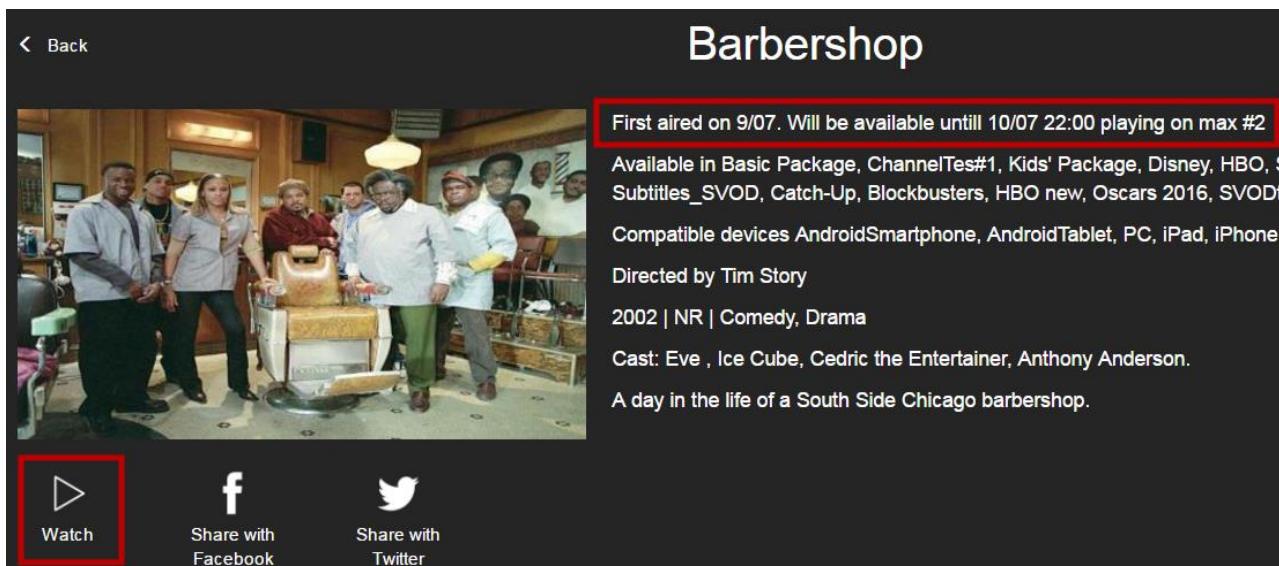
A button to launch a Catch-Up stream should be displayed if Catch-Up is authorized for the program. Typically, a regular Watch button can be used.

Note: The button to launch Catch-Up should only be displayed after the program has ended.

In the image below, the program has ended and thus the *Watch* button is shown.

Metadata should display:

- The original time at which the program was broadcast.
- The time until when the catch-up of the program will be available. (See [174 Determining the Catch-Up Availability Period \(p.487\)](#).)



Two different requests are provided for determining whether a program supports Catch-Up or not:

- To get information for a single program, use *GetExtendedProgram*. This is most useful when creating a Program screen.
- To get information on multiple programs, use *GetProgramList*. This is most useful when creating a Now Playing or EPG screen. Within the Now Playing screen, the Catch-Up information should be read for the program scheduled to be played at the current time.

173.2 Requests and Responses: GetExtendedProgram

Request: [GetExtendedProgram](#) (from RiGHTv Front-end API)

This call obtains detailed information on a specified program.

The table below presents the most commonly used parameters:

Parameter	Description
program_external_id	The external ID of the particular program.

Response: [ExtendedProgram](#) (from RiGHTv Front-end API)

This entity lists detailed information on the specified program. For verifying whether Catch-Up is supported, the relevant property is:

Property	Description
flags	A bitwise description of the program properties, where <i>isCatchUpEnabled</i> is indicated by bit (4).
startDate, endDate	The start and end dates of the original broadcast, specified in milliseconds from 1/1/1970.

For more properties, see the RiGHTv Front-end API Developers Guide.

173.3 Requests and Responses: GetProgramList

Request: [GetProgramLists](#) (from RiGHTv Front-end API)

Gets a list of all programs scheduled for the specified channel(s), together with a range of information on each one. Significant parameters include:

Property	Description
channel_external_ids	A list of the ID's of all channels for which program info is required.
start_date	The earliest date from which programs should be returned. Specified in milliseconds from 1/1/1970).
end_date	The latest date from which programs should be returned. Specified in milliseconds from 1/1/1970).

Note

The start date must always start at the beginning of a day, and the end date must be the end of the day.

For more parameters, see the RiGHTv Front-end API Developers Guide.



Response: **Program** (from RiGHTv Front-end API)

Lists all programs that match the parameters specified by the *GetProgramLists* action, and provides a range of information on each program listed, including whether Catch-Up is supported or not. For verifying whether Catch-Up is supported, the relevant property is:

Property	Description
<i>flags</i>	A bitwise description of the program properties, where <i>isCatchUpEnabled</i> is indicated by bit (4).
<i>startDate</i> , <i>endDate</i>	The start and end dates of the original broadcast, specified in milliseconds from 1/1/1970.

For more properties, see the RiGHTv Front-end API Developers Guide.

173.4 Related Tutorials

To	Refer to Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Determine when Catch-Up availability for a program ends	174 Determining the Catch-Up Availability Period (p.487)
Determine whether the account is subscribed to a Catch-Up service	175 Checking if Account is Subscribed to Catch-Up (p.490)
Create a TV Guide screen	63 Create a TV Guide (p.188)
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

173.5 Example

https://<operator>.tvaas.com /RTEFacade/GetExtendedProgram?program_external_id=q3u89um5oiarp6s3n88ivnpqvo&language_code=en&client=json

This call gets information on a specific program.

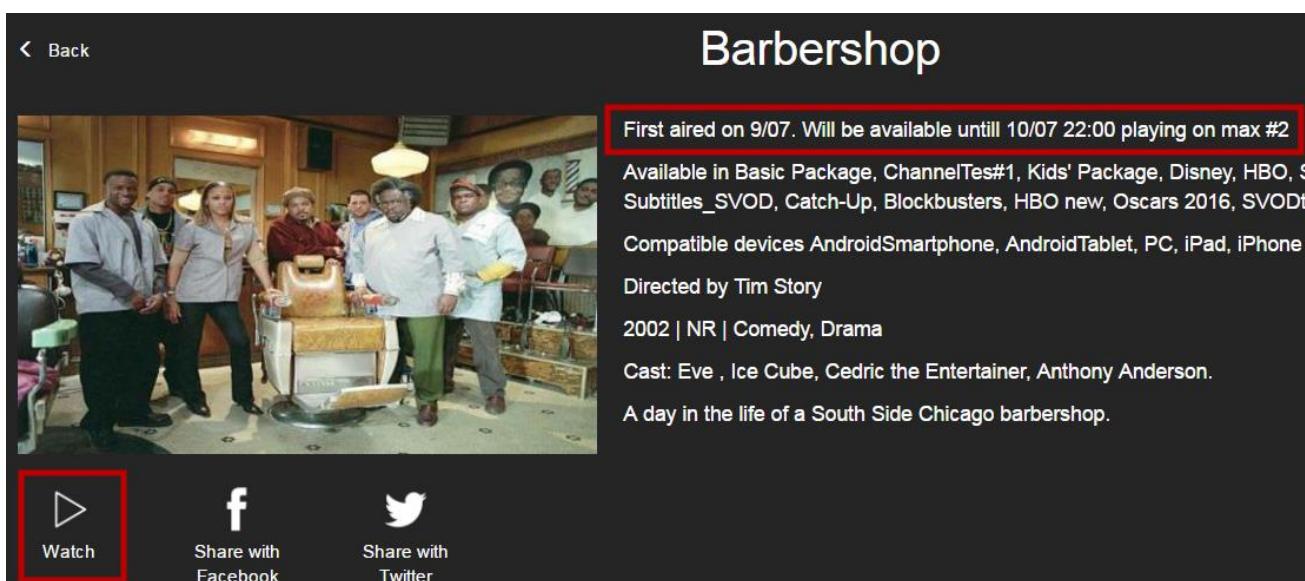
As the *flags* value is 4, and Catch-Up support is indicated by (4), this program does support Catch-Up.

```
],  
"endDate": 1486566000000,  
"year": 1999,  
"contentId": "",  
"flags": 4,
```

174 Determining the Catch-Up Availability Period

174.1 Overview

The Program screen should indicate the date and time until when catch-up viewing of the program will be available.



174.2 Requests and Responses

Request: [GetTimeShiftingServicesConfig](#) (from RiGHTv Front-end API)

This call obtains configuration information for all time-shifting services. No parameters are required.

Response: [TimeShiftingServicesConfig](#) (from RiGHTv Front-end API)

This entity lists configuration settings for all time-shifting services. Properties relevant to catch-up are listed below.

Property	Description
<i>catchUpAvailabilityPeriodInHours</i>	The number of hours following the completion of the program during which catch-up viewing is available. To display the catch-up expiry date and time, add the number of hours specified by this property to the program end time, obtained using GetExtendedProgram (see 173 Identify Catch-Up Enabled Programs (p.483)).

174.3 Related Tutorials

To	Refer To Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Identify Catch-Up enabled programs	173 Identify Catch-Up Enabled Programs (p.483)
Determine whether the account is subscribed to a Catch-Up service	175 Checking if Account is Subscribed to Catch-Up (p.490)
Create a TV Guide screen	63 Create a TV Guide (p.188)
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

174.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetTimeShiftingServicesConfig?client=json`

This call retrieves time-shifting configuration settings.

The response indicates that catch-up viewing is available for 24 hours after the program ends.



```
[{"metadata": {  
    "request": "GetTimeShiftingServicesConfig",  
    "timestamp": 1499672762676  
},  
"response": {  
    "catchUpAvailabilityPeriodInHours": 24,  
    "responseElementType": "TimeShiftingServicesConfig",  
    "npvrAvaliabilityPeriod": 3,  
    "npvrExpirationPeriod": "",  
    "npvrExpirationPeriodUnit": "",  
    "leadTimeInMinutes": 5,  
    "npvrAvaliabilityPeriodUnit": 4,  
    "cpvrAvailabilityPeriodInHours": 120,  
    "lagTimeInMinutes": 5  
}  
}]
```



175 Checking if Account is Subscribed to Catch-Up

175.1 Overview

When a user clicks a button to launch a catch-up viewing session of a past program, the program should only play if both of the following are true:

- The account is subscribed to the relevant channel. See [66 Identify Subscribed Channels \(p.197\)](#).
- The account is subscribed to a Catch-Up service, as explained below.

Parental control must also be taken into account, see [185 Parental Control \(p.509\)](#).

175.2 Requests and Responses

Request: [GetHouseholdServiceList](#) (from RiGHTv Front-end API)

This call gets a list of all services to which the logged in account is subscribed.

No parameters are necessary.

Response: [ServiceSubscription](#) (from RiGHTv Front-end API)

Each *ServiceSubscription* entity returned details of one service to which the account is subscribed. For verifying whether the account is subscribed to a Catch-Up service, the relevant property is:

Property	Description
<i>model</i>	A service of model TSTV indicates that the account is subscribed to a Catch-Up service.

For more properties, see the RiGHTv Front-end API Developers Guide.

175.3 Related Tutorials

To	Refer To Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Identify programs that support Catch-Up	173 Identify Catch-Up Enabled Programs (p.483)
Create a TV Guide screen	63 Create a TV Guide (p.188)

To	Refer To Tutorial
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

176 Playing a Program through Catch-Up

176.1 Overview

When a user attempts to launch a Catch-Up viewing session of a past program, the relevant playback URL must be obtained.

The following verifications must be performed before playing the channel:

- Check that the account is subscribed to the channel. See [66 Identify Subscribed Channels \(p.197\)](#). If the account is not subscribed to the channel, within the web app the user should be provided with valid purchase options.
- Check that the account is subscribed to a Catch-Up service. See [175 Checking if Account is Subscribed to Catch-Up \(p.490\)](#).
- Check that the parental rating of the program does not exceed the accounts' limit. See [185 Parental Control \(p.509\)](#).

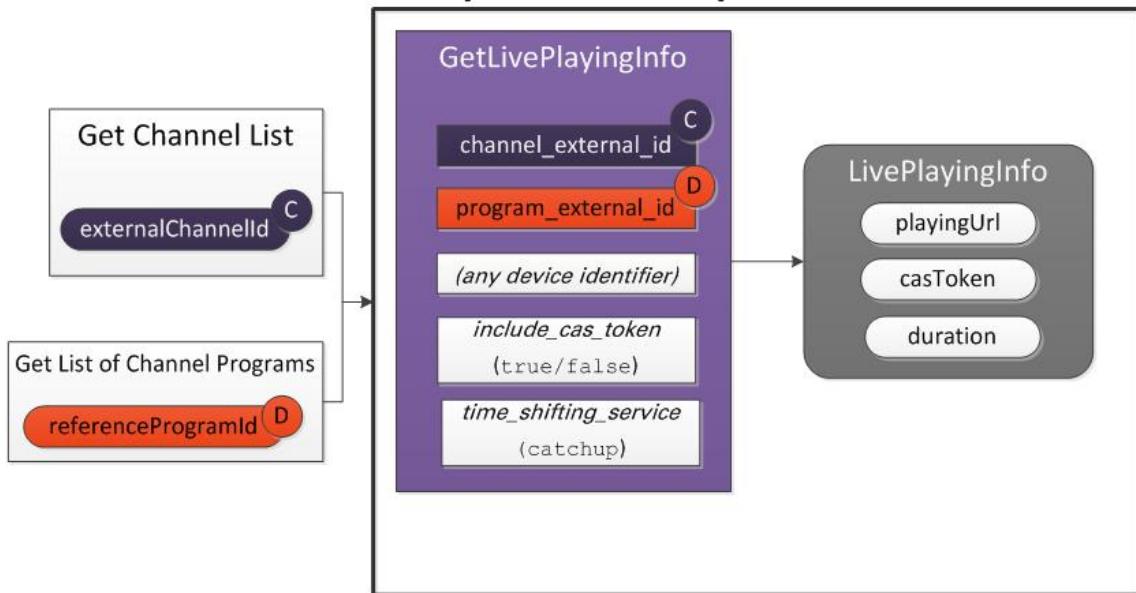
This tutorial explains how to obtain the Catch-Up playback URL. For design guidelines for the player controls, see [178 Guidelines for Catch-Up Player Controls \(p.498\)](#).

Note that resume viewing is supported when playing catch-up programs, so that if the user stops watching a program in catch-up mode and then resumes playing it later, even on a different device, the program will continue from the last watched position. The last watched position is returned with the playback URL. For the correct resume position to be returned, the last watched position must be marked whenever the user pauses or closes a program being watched in catch-up mode. See [177 Marking Last Watched Position \(p.496\)](#).



176.2 Requests and Responses

Play Catch-Up



Request: [GetLivePlayingInfo](#) (from RiGHTv Front-end API)

Gets the playback URL for playing the channel through Catch-Up. This request requires that the user be logged in. The following parameters are required:

Parameter	Description
channel_external_id	The ID of the channel to play.
One of: <i>mac_address</i> <i>serial_number</i> <i>smart_card_id</i>	One the device (terminal) identifier for the requesting device. <i>serial_number</i> is recommended.
include_cas_token	If set to <code>true</code> requests that the CAS token should be calculated and included in the result. The default value is <code>false</code> . A CAS token is required if a license is needed to play the channel.
program_external_id	The ID of the program to be played.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [LivePlayingInfo](#) (from RiGHTv Front-end API)

Provides the playback URL for the requested channel, and if requested, the CAS token required for viewing licensed content. The program's duration is also provided.



Property	Description
playingUrl	The playback URL.
casToken	The CAS token, if requested.
duration	The duration of the program in milliseconds. The duration provided takes into account the lag and lead time set in the system configuration, thus ensuring that the entire program is included, even if it started somewhat earlier or later than planned. The duration is needed for the suggested Player implementation, see 178 Guidelines for Catch-Up Player Controls (p.498) .
resumePosition	The last position marked by the subscriber, in milliseconds, from the beginning of the program.

For more properties, see the RiGHTv Front-end API Developers Guide.

176.3 Related Tutorials

To	Refer to Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Adapt the Player controls to support Catch-Up	178 Guidelines for Catch-Up Player Controls (p.498)
Create a TV Guide screen	63 Create a TV Guide (p.188)
Create a Program screen	69 Create a Program Screen (p.202)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)



Record the last watched position so as to allow the user to continue viewing later from the same point	177 Marking Last Watched Position (p.496)
--	---

176.4 Example

https://<operator>.tvaas.com /RTEFacade/GetLivePlayingInfo?program_external_id=aj2nm79fgepsvm5f04p580ru4s&channel_external_id=Tvaas-MSMA-TTML-Arte&serial_number=MOZILLA_214CA0CB-4151-2021-0FD3-6E1B615987FE&include_cas_token=true&time_shifting_service=catchup&client=json&language_code=en

This call requests the Catch-Up playing info for the specified program. The response is shown below.

```
[{"metadata": {  
    "request": "GetLivePlayingInfo",  
    "timestamp": 1486884803683  
},  
 "response": {  
    "duration": 1800000,  
    "responseElementType": "LivePlayingInfo",  
    "casToken": "QvXlrjMKS+iNTS0m12aVwo/7S//crAtOQFqWj0woEWnYHm86X1SPSGJpCMYD  
+qyfG17nZ6Ufw3oGCDaEw3ZBnkg  
+Dq52tzc23cYYw83dpBVWkTC2oCMvSEeKBDseTFuwvqdj1X7DDqXQgn18AgW0k6ymDmvbz  
+n4h4jfCS4oORH/5BZMX0hXPKLvoif0q9KB",  
    "parentalRatings": [],  
    "resumePosition": "",  
    "channelId": 8016,  
    "playingUrl": "http://harmonic2.e2e.purpledrm.com.edgesuite.net/Content/SS  
/StartOver/Channel(name=Tvaas-MSMA-TTML-Arte,startTime=148688460000000000  
,endTime=148688640000000000).isml/Manifest",  
    "casTokenExpiration": 1486884923670,  
    "urlExpiration": 1486884803670  
}  
}]
```



177 Marking Last Watched Position

177.1 Overview

Resume viewing is supported when playing catch-up programs, so that if the user stops watching a program in catch-up mode and then resumes playing it later, even on a different device, the program will continue from the last watched position. The last watched position is returned with the playback URL.

For the correct resume position to be returned, the last watched position must be saved whenever the user pauses or closes a program being watched in catch-up mode.

177.2 Requests and Responses

Request: [MarkRecordingPosition \(from RiGHTv Front-end API\)](#)

Saves the last watched position in a catch-up, NPVR, or Start Over viewing session. For catch-up viewing, the following parameters are required:

Parameter	Description
position	The number of milliseconds elapsed since the beginning of the program.
program_id	The internal ID of the program. (If specifying <i>program_external_id</i> instead, <i>channel_external_id</i> must also be specified.)

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Result \(from RiGHTv Front-end API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
status	Indicates whether the request resulted in a SUCCESS or FAILURE.
message	A short message from the server



177.3 Related Tutorials

To	Refer to Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Play a live program in Catch-Up mode	176 Playing a Program through Catch-Up (p.492)

177.4 Example

`https://<operator>.tvaas.com /RTEFacade/MarkRecordingPosition?position=38219381&program_external_id=aj2nm79fgepsvm5f04p580ru4s&channel_external_id=Tvaas-MSMA-TTML-Arte&client=json&language_code=en`

This call saves the last watched position in a catch-up viewing session of a program. The program is defined by its *program_external_id* and *channel_external_id*.

178 Guidelines for Catch-Up Player Controls

178.1 Overview

This tutorial provides general guidelines for the Player controls when playing a program through Catch-Up.

1. For instructions on how to obtain the playback URL, see [176 Playing a Program through Catch-Up \(p.492\)](#).
2. The Player should display relevant information, as indicated below.



- a. The current playback position (indicated by time and by marker on progress bar).
- b. The program's duration. (The program's duration (difference between its start time and end time can be obtained from the *Program* response to *GetProgramList*, see [65 Get List of Channel Programs \(p.193\)](#), or from the *ExtendedProgram* response to *GetExtendedProgram*, see [70 Get Program Metadata \(p.205\)](#)).

3. The Player should provide required functionality:



4. Play/pause: Unlike when playing live content, the user should be able to pause (and then play) the stream when viewing in Catch-Up mode.
5. Interactive Progress Bar: The user should be able to drag the current position marker and/or tap any position within the progress bar to jump to any position within the Catch-Up stream.
6. Once the Catch-Up stream has been viewed until the end of the program, a pop-up message should display, offering the user the option to switch back to regular live viewing of the channel.
7. Channel zap events should be registered in the following cases:
 - After the Catch-Up stream has been viewed for a configurable amount of time.
 - When the user switches back to the live stream.

See [201 Register Event: Channel Zap \(p.540\)](#).

178.2 Related Tutorials

To	Refer to Tutorial
Learn about implementing Catch-Up	166 Implementing Catch-Up (p.464)
Play a live program in Catch-Up mode	176 Playing a Program through Catch-Up (p.492)
Register an event for channel zapping	201 Register Event: Channel Zap (p.540)
Play live programs	76 Playing Live Programs (p.220)
Create a Now Playing screen	79 Create a Now Playing Screen (p.225)

179 Working with Languages

179.1 Overview

Multiple languages can be defined in the system. Information within the system, such as metadata for movies, services and channels, can be stored in multiple languages.

When calls for such information are made, the information will be sent back in the user's preferred language, if so translated.

179.2 Related Tutorials

Managing languages involves the following:

- Providing the ability to specify the preferred language, for example, from a drop-down list:
 - [180 Get a List of Supported Languages \(p.501\)](#)
 - [181 Set the Language for a New Account \(p.503\)](#)
 - [182 Set the Language for an Existing Account \(p.504\)](#)
 - [183 Get the Account's Current Language \(p.506\)](#)
- Specifying the required language of the response within an API request.
 - [184 Specifying the Response Language \(p.508\)](#)

180 Get a List of Supported Languages

180.1 Overview

You can obtain a list of supported languages to display to the user for selection.

180.2 Requests and Responses

Request: [GetLanguageList \(from RiGHTv Front-end API\)](#)

Obtains a list of supported languages. No parameters are required.

Response: [Language \(from RiGHTv Front-end API\)](#)

A list of supported languages. The entity includes the following properties:

Property	Description
<i>code</i>	The two or three letter language code, for example en for English or es for Spanish.
<i>name</i>	The full name of the language as defined in the database, for example English.

180.3 Related Tutorials

To	Refer To Tutorial
Learn about working with languages	179 Working with Languages (p.500)

180.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetLanguageList?&client=json`

Obtains a list of supported languages.

The sample response shows that English, Spanish and Portuguese are supported.



```
{  
  "metadata": {  
    "request": "GetLanguageList",  
    "fullLength": 3,  
    "timestamp": 1459235884399  
  },  
  "response": [{  
    "responseElementType": "Language",  
    "code": "en",  
    "name": "English"  
  },  
  {  
    "responseElementType": "Language",  
    "code": "pt",  
    "name": "Portuguese"  
  },  
  {  
    "responseElementType": "Language",  
    "code": "es",  
    "name": "Spanish"  
  }]  
}
```

181 Set the Language for a New Account

181.1 Overview

Once the account's language is defined, all metadata returned by action calls will be returned in that language, if so translated.

The account language can be defined when the account is first created using the RegisterHousehold action, by providing the language_code parameter.

Depending on the system configuration, either a two or three letter language code is used.

For more information on registering new accounts, see [81 Create an Account \(p.229\)](#).

181.2 Related Tutorials

[179 Working with Languages \(p.500\)](#)

181.3 Example

`https://<operator>.tvaas.com/RTEFacade/RegisterHousehold?username=reuserg@gmail.com&password=testabc1&language_code=pt&client=json`

This request sets the new account's default language to Portuguese, using the two letter code pt.

182 Set the Language for an Existing Account

182.1 Overview

The account language for an existing account can be changed.

182.2 Requests and Responses

Request: [SetLanguage](#) (from RiGHTv Front-end API)

This request sets the language for the logged-in account. Only one parameter is specified:

Parameter	Description
<i>new_language</i>	The language code, either a two or three letter code, depending on the system configuration.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server.

182.3 Related Tutorials

To	Refer To Tutorial
Learn about working with languages	179 Working with Languages (p.500)

182.4 Example

https://<operator>.tvaas.com/RTEFacade/SetLanguage?new_language_code=pt&client=json

This request sets the logged-in account's language to Portuguese, using the code pt.

Sample response of a successful SetLanguage call is shown below.

```
[  
  "metadata": {  
    "request": "SetLanguage",  
    "timestamp": 1459236365468  
  },  
  "response": {  
    "message": "Updated language code successfully",  
    "status": "SUCCESS"  
  }  
]
```

183 Get the Account's Current Language

183.1 Overview

Although all calls will return information in the account's defined language, if so translated, you may still want to read the account's current defined language, for example, to show on a Settings or Options screen. Use the *GetHousehold* action to get this information.

183.2 Requests and Responses

Request: [GetHouseHold](#) (from RiGHTv Front-end API)

The *GetHousehold* action returns a range of information about the logged-in account, including its defined language. No parameters are required.

Response: [Household](#) (from RiGHTv Front-end API)

The *Household* entity returns a range of information about the account, including:

Property	Description
languageCode	The two or three letter language code representing the account's defined language.

183.3 Related Tutorials

To	Refer to Tutorial
Learn about working with languages	179 Working with Languages (p.500)

183.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHousehold?&client=json>

This request calls for account information for the logged-in account.

In the sample below, the account's language is seen to be Spanish, represented by the code es.

```
[{"metadata": {  
    "request": "GetHousehold",  
    "timestamp": 1459239392014  
},  
"response": {  
    "cluster": {},  
    "timezone": "INSTALLATION_TZ",  
    "billingMethod": "postpaid",  
    "externalId": "HH_Trial",  
    "discount": 0,  
    "groups": [],  
    "languageCode": "es",  
    "transactionsPaymentMode": "normal",  
    "spentTotal": 122,  
    "isHouseholdScope": true,  
    "responseElementType": "Household",  
    "paymentAuthentication": "always",  
    "spendingLimit": -1,  
    "phone": "",  
    "maxTerminals": "",  
    "extrafields": [],  
    "billingDay": 31,  
    "internalPrepaidBalance": 0,  
    "plannedInactivationDate": "",  
    "id": 10300,  
    "parentalPin": "0000", ←  
    "username": "1"  
}  
}
```



184 Specifying the Response Language

184.1 Overview

By default, the response to any call will be returned in the account's defined language. However, in any action call you can specify the desired language for the response.

This is useful, for example, when:

- The user has not logged in but has selected an interface language.
- A logged-in user wishes to view data in a language other than that defined for the account, and does not want to change the account's defined language.

184.2 Requests and Responses

Key Parameter: `language_code` (from RiGHTv Front-end API)

Any relevant request can include the parameter `language_code`, specifying the two or three letter language code.

184.3 Related Tutorials

To	Refer to Tutorial
Learn about working with languages	179 Working with Languages (p.500)

184.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetAggregatedVideo?external_asset_id=Cinderella&language_code=es&client=json`

This request gets aggregated video data for a specific movie, specifying that the response should be sent using the Spanish (es) translation, if available.

185 Parental Control

185.1 Overview

The RiGHTv Parental Control system enables restriction of all content including video assets, channels and programs. Parental control extends to search results, recommendations, and all types of orders and tickets. All of these entities include parental rating metadata, allowing the application to prevent them from being viewed or accessed without the parental PIN.

Each account has a parental pin defined. The parental PIN must be 4 characters in length, and can include numbers and letters. The parental PIN can be changed.

The application must request the parental pin before showing adult content, compare that to the parental pin stored in the Service Delivery Platform, and only show the content if the correct pin was entered.

Typical applications of parental control include:

- Anonymous (not logged-in) users can't see any Adult content. Even Adult category tabs are not shown.
- Accessing an Adult category screen requires parental PIN entry.
- Accessing a movie screen that exceeds a certain parental level threshold requires parental PIN entry.
- Recommendation lists are filtered to exclude content above a specified parental rating level.

185.2 Related Tutorials

To	Refer to Tutorial
Check whether the Parental PIN entered by the user is correct	186 Verify the Parental Pin (p.511)
Obtain a list of parental ratings used in the system	191 Get a List of Parental Ratings Used (p.519)
Determine which parental rating was assigned to an entity	187 Determine Parental Level of an Entity (p.513)
Change the account's Parental PIN	188 Changing Parental Pins (p.514)



Set the parental control threshold level to apply	192 Set the Parental Control Threshold (p.521)
Exclude adult content from API responses	193 Exclude Adult Content from Returned Lists (p.522)

186 Verify the Parental Pin

186.1 Overview

Whenever parental control is required, the application should request the user to enter the parental PIN. That should be compared to the stored parental PIN. The account's parental PIN must thus be retrieved.

186.2 Requests and Responses

Request: [GetHousehold](#) (from RiGHTv Front-end API)

The *GetHousehold* action returns a range of information about the logged-in account, including its parental PIN. No parameters are required.

Response: [Household](#) (from RiGHTv Front-end API)

The Household entity returns a range of information about the account, including:

Property	Description
<i>parentalPin</i>	The currently defined parental PIN for the logged-in account.

186.3 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)

186.4 Example

<https://<operator>.tvaas.com/RTEFacade/GetHousehold?&client=json>

This request calls for account information for the logged-in account.

In the sample below, the account's parental PIN is seen to be 0000.



```
[{"metadata": {  
    "request": "GetHousehold",  
    "timestamp": 1459239392014  
},  
"response": {  
    "cluster": {},  
    "timezone": "INSTALLATION_TZ",  
    "billingMethod": "postpaid",  
    "externalId": "HH_Trial",  
    "discount": 0,  
    "groups": [],  
    "languageCode": "es",  
    "transactionsPaymentMode": "normal",  
    "spentTotal": 122,  
    "isHouseholdScope": true,  
    "responseElementType": "Household",  
    "paymentAuthentication": "always",  
    "spendingLimit": -1,  
    "phone": "",  
    "maxTerminals": "",  
    "extrafields": [],  
    "billingDay": 31,  
    "internalPrepaidBalance": 0,  
    "plannedInactivationDate": "",  
    "id": 10300,  
    "parentalPin": "0000", ←  
    "username": "1"  
}  
}]
```

187 Determine Parental Level of an Entity

187.1 Overview

Every relevant entity within the system includes the following properties:

- *prLevel*, the parental rating level, in number format.
- *prName*, the name of the parental rating level, as a string.

Compare the entity's parental rating level to the threshold level to determine whether parental control must be applied or not.

The client application itself is responsible for setting the parental threshold level. The client application can either hard code, or set per the user's preference, the threshold level, as a number from 0 to 999. So for example, if the threshold level is set to 500, a movie with *prLevel* set to 800 must not be played, but one with a *prLevel* of 400 should be played.

Entities with parental levels specified include:

- *VideoAsset*, for VOD movies
- *Channel*, for live channels
- *ExtendedProgram*, *Program*, for live programs
- *SearchedProgram*, for programs returned by a search
- *SearchedRecording*, for recordings returned by a search
- *Order*, for payment transactions
- *VideoTicket*, for tickets for videos
- *ProgramTicket*, for tickets for paid programs

187.2 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)



188 Changing Parental Pins

188.1 Overview

The parental pin can be updated or reset:

- Updating the parental pin involves submitting a new pin chosen by the user. The application should verify the user's identity by requesting and verifying the old parental PIN before allowing a new one to be submitted.
- Resetting the parental pin results in the main member receiving an email with a new randomly generated pin. This would typically be used in a case in which the current parental pin has been forgotten.

188.2 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)
Update the account's Parental PIN	189 Updating the Parental Pin (p.515)
Reset the account's Paretnal PIN	190 Resetting the Parental Pin (p.517)



189 Updating the Parental Pin

189.1 Overview

Updating the parental pin involves submitting a new pin chosen by the user. The application should verify the user's identity by requesting and verifying the old parental PIN before allowing a new one to be submitted.

189.2 Requests and Responses

Request: [UpdateParentalPin](#) (from RiGHTv Front-end API)

The UpdateParentalPin action submits a new parental PIN for the logged-in account. Only one parameter is required:

Parameter	Description
<i>parental_pin</i>	Specifies the new parental PIN.

For more parameters, see the RiGHTv Front-end API Developers Guide.

Response: [Result](#) (from RiGHTv Front-end API)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

189.3 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)
Changing the account's Parental PIN	188 Changing Parental Pins (p.514)

189.4 Example

This request updates the account's parental PIN to 1234.

https://<operator>.tvaas.com/RTEFacade/UpdateParentalPin?&parental_pin=1234&client=json



In the sample below, the account's parental PIN has been successfully updated.

```
{  
  "metadata": {  
    "request": "UpdateParentalPin",  
    "timestamp": 1459255148838  
  },  
  "response": {  
    "message": "Updated Parental PIN successfully",  
    "status": "SUCCESS"  
  }  
}
```

190 Resetting the Parental Pin

190.1 Overview

Resetting the parental pin results in the main member receiving an email with a new randomly generated pin. This would typically be used in a case in which the current parental pin has been forgotten.

An operator account's Parental PIN can only be reset if an e-mail address has been defined.

190.2 Requests and Responses

Request: [ResetParentalPin \(from RiGHTv Front-end API\)](#)

The ResetParentalPin action sends a reset email to the account. No parameters are required.

Response: [Result \(from RiGHTv Front-end API\)](#)

Indicates whether the action succeeded or not. The entity includes the following attributes:

Property	Description
<i>status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>message</i>	A short message from the server

190.3 Related Tutorials

To	Refer to Tutorial
Learn about parental control	185 Parental Control (p.509)
Changing the account's Parental PIN	188 Changing Parental Pins (p.514)

190.4 Example

This request resets the account's parental PIN.

`https://<operator>.tvaas.com/RTEFacade/ResetParentalPin?&client=json`

In the sample below, the account's parental PIN has been successfully reset.

```
[{"metadata": {  
    "request": "ResetParentalPin",  
    "timestamp": 1459255769932  
},  
"response": {  
    "message": "Successfully sent email with reset parental pin instructions",  
    "status": "SUCCESS"  
}]
```

191 Get a List of Parental Ratings Used

191.1 Overview

The client application can get a list of all parental ratings used in the system using the *GetParentalRatingList* action. This list can be used, for example, in the My Settings screen where the user selects the level of parental rating to apply.

191.2 Requests and Responses

Request: [GetParentalRatingList](#) (from RiGHTv Front-end API)

The *GetParentalRatingList* action returns an array of all parental rating levels in the system. No parameters are required.

Response: [ParentalRating](#) (from RiGHTv Front-end API)

The ParentalRating entity returns a list of parental rating levels:

Property	Description
<i>id</i>	The ID of the parental rating level.
<i>name</i>	The name of the parental rating level.
<i>level</i>	The numeric level of the parental rating level, where 0 is no rating and 999 is the highest.

191.3 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)

191.4 Example

`https://<operator>.tvaas.com/RTEFacade/GetParentalRatingList?&client=json`

This request calls for a list of parental rating levels defined in the system.

The sample below shows a number of parental rating levels.

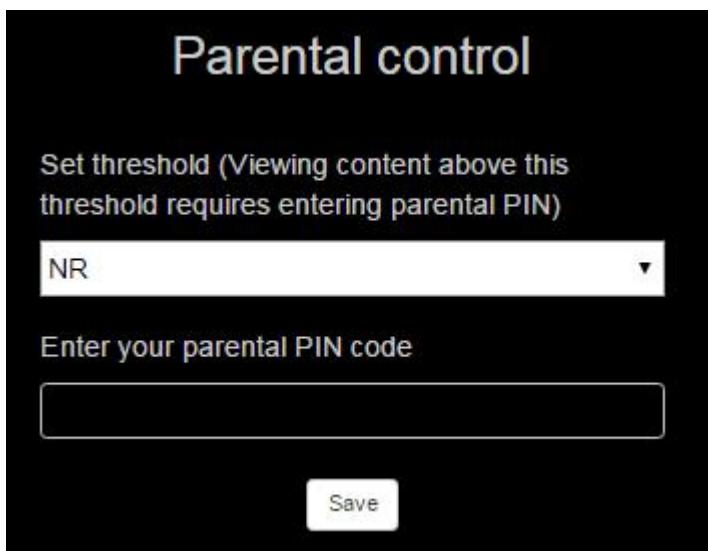


```
{  
    "responseElementType": "ParentalRating",  
    "level": 3,  
    "name": "TVY7",  
    "id": 4037  
},  
{  
    "responseElementType": "ParentalRating",  
    "level": 4,  
    "name": "R",  
    "id": 4038  
},  
{  
    "responseElementType": "ParentalRating",  
    "level": 5,  
    "name": "NC-17",  
    "id": 4039  
},  
{  
    "responseElementType": "ParentalRating",  
    "level": 5,  
    "name": "TVMA",  
    "id": 4040  
},  
{  
    "responseElementType": "ParentalRating",  
    "level": 9999,  
    "name": "Block",  
    "id": 3003  
}
```

192 Set the Parental Control Threshold

192.1 Overview

The user must be able to specify the maximum parental rating of content that will not require the Parental PIN to be entered.



The parental control level set must be stored within the app itself. No back-end support is provided.

The Parental PIN must be requested before allowing this setting to be changed.

192.2 Related Tutorials

To	Refer to Tutorial
Check whether the Parental PIN entered by the user is correct	186 Verify the Parental Pin (p.511)
Obtain a list of parental ratings used in the system	191 Get a List of Parental Ratings Used (p.519)
Learn about parental control	185 Parental Control (p.509)

193 Exclude Adult Content from Returned Lists

193.1 Overview

Many request calls can use filters to exclude content whose parental rating exceeds the threshold limit.

This section provides an example of filtering video recommendations. The same approach can be used for filtering catalog information returned by the call *GetUnifiedList*, or for filtering lists of live channels returned by *GetChannelList*, for example.

193.2 Requests and Responses

Request: [GetVideoRecommendationList](#) (from COMPASS RT API)

The recommendation lists provided are customized based on parameters in the action call. For limiting the results by parental level, the relevant parameter is:

Parameter	Description
<i>max_pr_level</i>	Specify the maximum parental rating level of movies to be listed.

For more parameters, see the Compass RT Developers Guide.

Response: [VideoRecommendation](#) (from COMPASS RT API)

Only videos that match the parental level requirements are returned. The returned entity includes various properties, including:

Property	Description
<i>prLevel</i>	The parental level rating.

193.3 Related Tutorials

To	Refer To Tutorial
Learn about parental control	185 Parental Control (p.509)
Set the parental control threshold level	192 Set the Parental Control Threshold (p.521)



193.4 Example

https://<operator>.tvaas.com/compass/GetVideoRecommendationList?type=most_popular&content_type=Movie&client=json

This request calls for a list of recommended Movies, as rated by their Popularity.

https://<operator>.tvaas.com/compass/GetVideoRecommendationList?type=most_popular&max_pr_level=2&content_type=Movie&client=json

This request calls for a list of recommended Movies, as rated by their Popularity, limited to movies with parental rating levels not higher than 2.

Extracts of the two sample responses are shown below. Note how the first response includes 5 movies, while the second only includes 1. Also, the first movie listed in the first response had a parental rating level of 3. That has been excluded in the second response.

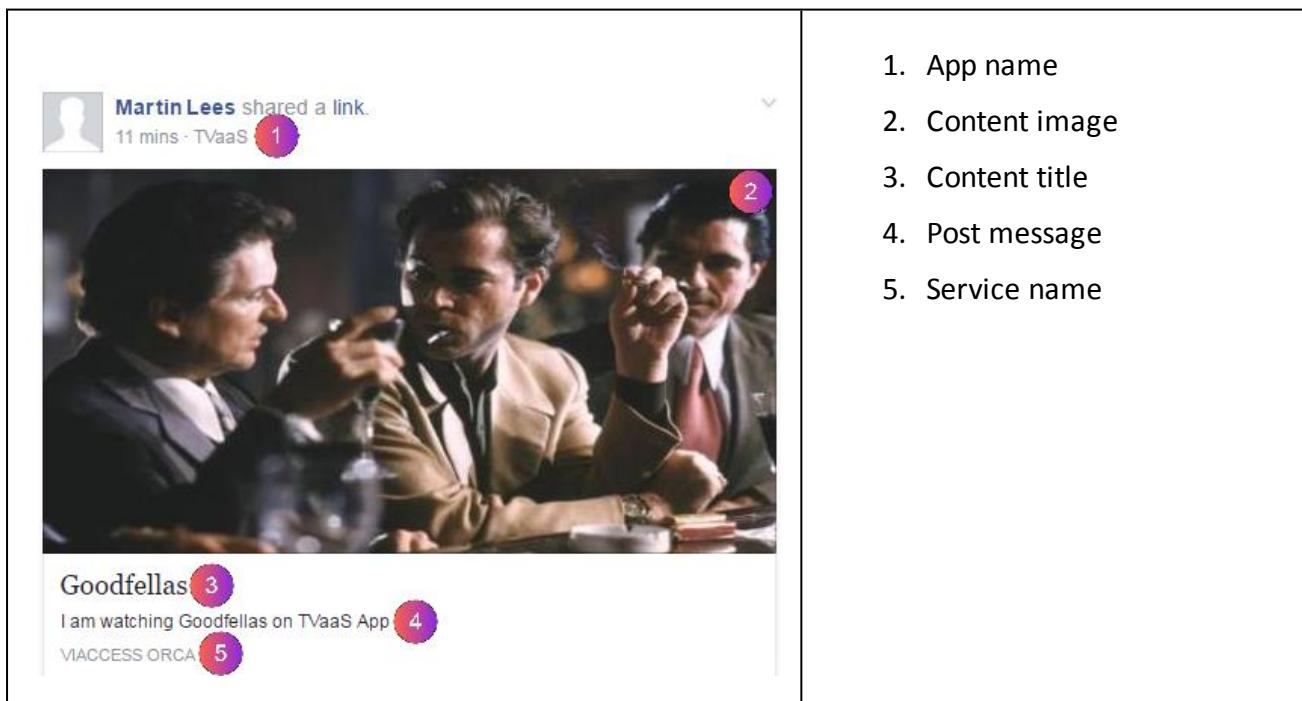
Sample response without Parental Filtering	Sample response with Parental Filtering
<pre>{ "metadata": { "request": "GetVideoRecommendationList", "fullLength": 5, "timestamp": 1459315046892 }, "response": [{ "images": [{ "name": "VerticalImage", "value": "/attachments/jurassic_world_ver.jpg" }, { "name": "HorizontalImage", "value": "/attachments/jurassic_world_hor.jpg" }], "prLevel": 3, "genres": [{ "name": "Action", "externalId": "Action", "id": "Action" }, { "name": "Adventure", "externalId": "Adventure", "id": "Adventure" }], "availabilities": [...] }] }</pre>	<pre>{ "metadata": { "request": "GetVideoRecommendationList", "fullLength": 1, "timestamp": 1459315089527 }, "response": [{ "images": [{ "name": "HorizontalImage", "value": "/attachments/minions_hor_for_nadav.jpg" }, { "name": "VerticalImage", "value": "/attachments/minions_ver.jpg" }], "prLevel": 2, "genres": [{ "name": "Comedy", "externalId": "Comedy", "id": "Comedy" }, { "name": "Animation", "externalId": "Animation", "id": "Animation" }], "availabilities": [...] }] }</pre>

194 Sharing Via Facebook

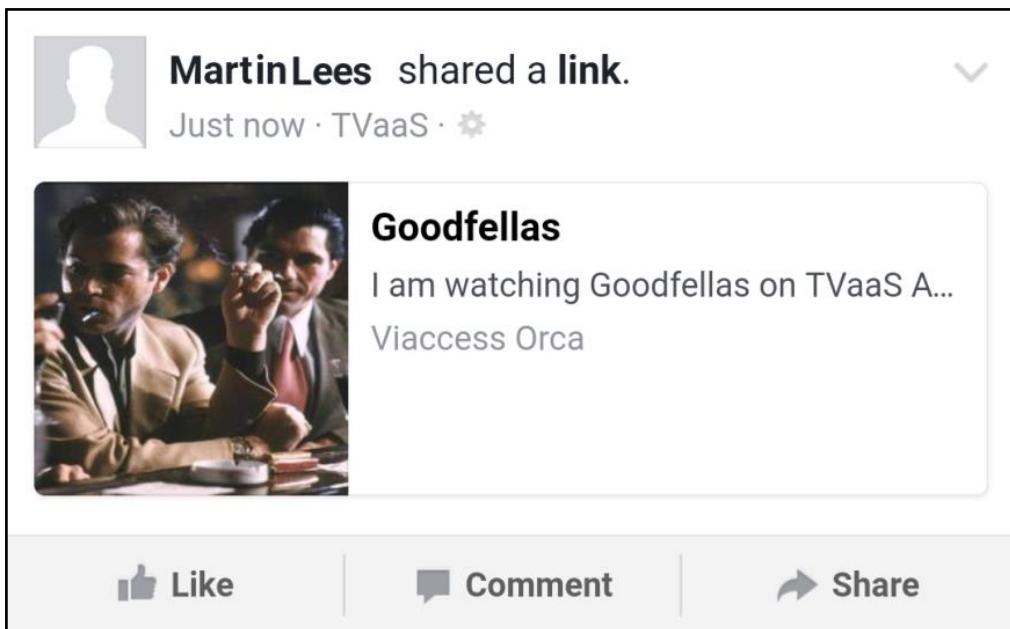
By allowing users to share content recommendations over social media, the Operator can greatly increase exposure to the services offered.

General guidelines for designing a Facebook sharing feature are provided below. For technical instructions on implementing Facebook sharing, consult Facebook's online resources.

A typical Facebook sharing post is shown below. Its five components are listed.



Depending on the devices used for posting and viewing, the post may be displayed differently, for example:



Guidelines are provided below.

194.1 App Name

The name entered when applying for a Facebook App Id will always display as the app name above the post.

194.2 Content Image

To provide an optimal image in the sharing post and to prevent automatic cropping, follow the guidelines below:

- The horizontal image (PageHorizontalImage) associated with the content must be used.
- Images must have dimensions of 1204x692 pixels, as per the recommendations for the TVaaS Service.

194.3 Content Title

The Content Title should display the name of the asset. Refer to the table below to learn how to obtain the asset title for different content types.

Content Type	Action	Response Property	See
VOD Movie	GetVideoAsset	Name	34 Get Movie Metadata (p.108)
Program	GetExtendedProgram	Name	70 Get Program Metadata (p.205)

194.4 Post Message

The Operator can define the text to be used. The standard text should include placeholders in which to include the content title and the app name.

Within the TVE Apps, the message is configured within the Language.JSON file, in the parameter “FB_MESSAGE”.

194.5 Service Name

Displays the name of the Operator’s service.

In the TVE Apps, the Service Name reflects the Application Name, defined in the application settings.

195 Registering Events for Content Discovery

195.1 Why Content Discovery?

From the customer's point of view, being able to find relevant and appealing content quickly and easily is an important requirement of a TV app. To the Operator, the sooner a customer is able to find relevant content, the higher the chance of a conversion to sale. It is thus in both parties' interests that the app be able to present customized lists of recommended content to each individual user.

In addition to recommendations, Operators need also to promote new and exciting content that they wish to publicize to maximize their ROI.

TVaaS includes VO's Content Discovery Platform (COMPASS), providing Operators with content recommendation, promotion, and powerful search capabilities.

195.2 The Role of the Apps

The apps play an important role in providing Content Discovery capabilities. At the one end, the apps must display lists of recommended and promoted content, and implement content search functionality. The implementation of these is presented throughout this guide.

The apps play a second, yet crucial role in content discovery. In order for the Content Discovery Platform to customize recommendations per individual user, it needs to learn about each user's content preferences. It is thus necessary to register a number of types of user actions; based on the information collected, the Content Discovery Platform is able to create a profile based on which it can issue user-targeted recommendations.

195.3 An Example

The Operator is able to implement a number of different content discovery engines, and may even blend the results of multiple engines to create a single list of recommendations.

For example, the Viewer Preferences engine is a content discovery engine designed to provide personalized recommendations for live content.

In order to create a meaningful list of recommended live content per user, the engine needs to learn the user's viewing preferences. Examples of user actions that must be registered for this engine to obtain the data it requires include:

- Channel zapping
- Setting reminders
- Ordering content

195.4 Registered Events

The table below lists the events that must be registered to ensure that the apps provide proper support for the Content Discovery Platform.

Event Description	COMPASS API	Event Type (used in APIs)	For More Information
Orders a video	RegisterEvent	order	196 Register Event: Order Video (p.529)
Watches a video trailer		preview	197 Register Event: Preview Trailer (p.532)
Watches a movie until the end (or near the end)		content_end	198 Register Event: Content End (p.534)
Rates a movie		Rate	199 Register Event: Rating (p.536)
Sets a reminder for a live program		reminder	200 Register Event: Set Reminder (p.538)
Views a live program for a defined length of time		Zap	201 Register Event: Channel Zap (p.540)
Selects an entity listed in a set of Search results	RegisterSearchEvent	cast/content	202 Register Search Event (p.542)
Clicks a content item	RegisterEvent	click	203 Register Click Event (p.545)



196 Register Event: Order Video

196.1 Overview

An event should be registered with COMPASS when a user orders a video.

How order events must be registered depends on the offer type from which the video is ordered:

- TVOD
Register the Order event at time of purchase.
- Movie Packages
When the package is purchased, Order events should be registered for all videos in the package.
- SVOD
When an order is placed for each SVOD video, an Order event should be sent for that video.

196.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<i>Type</i>	The type of event, must be set to <code>order</code> .
<i>external_content_id</i>	The external ID of the content.
<i>recommended</i>	Set to <code>true</code> or <code>false</code> , depending on whether the movie was clicked from a list of recommendations or not.

The following are only relevant if *recommended* is set to `true`:

<i>shuffle_num</i>	The number of shuffles performed by the user.
<i>Blend</i>	The blend used when creating the recommendations list.
<i>Engine</i>	Specifies the recommendation engine that generated the recommendation, such as <code>most_popular</code> or <code>user_rating</code> .
<i>is_paid</i>	Set to <code>true</code> if bought through a TVOD offer, set to <code>false</code> if bought through another offer type.
<i>Price</i>	The price paid for the content.
<i>device_model</i>	The type of device on which the device is to be played.

For more parameters, see the Compass RT API Guide.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
<i>Status</i>	Indicates <code>success</code> or <code>failure</code> .



196.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

196.4 Example

`https://<operator>-atp.tvaas.com/compass/RegisterEvent?type=order&recommended=false&device_model=PC&client=json&external_content_id=Fifty+Shades+of+Grey`

This registers an order event for the movie Fifty Shades of Grey.

A sample response is shown below, showing that the event was successfully registered.

```
[{"metadata": {  
    "request": "RegisterEvent",  
    "timestamp": 1467876569679  
},  
 "response": {  
    "message": "Event registered successfully",  
    "properties": {},  
    "status": "SUCCESS"  
}  
}
```



197 Register Event: Preview Trailer

197.1 Overview

An event should be registered with COMPASS when a user watches a trailer.

197.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<i>type</i>	The type of event, must be set to <code>preview</code> .
<i>external_content_id</i>	The external ID of the content.
<i>recommended</i>	Set to <code>true</code> or <code>false</code> , if the movie was clicked from a list of recommendations or not.

The following are only relevant if *recommended* is set to `true`:

<i>shuffle_num</i>	The number of shuffles performed by the user.
<i>blend</i>	The blend used when creating the recommendations list.
<i>engine</i>	Specifies the recommendation engine that generated the recommendation, such as <code>most_popular</code> or <code>user_rating</code> .
<i>device_model</i>	The type of device on which the device is to be played.

For more parameters, see the Compass RT API Guide.



Response: **Result** (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
status	Indicates success or failure.

197.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

197.4 Example

`https://<operator>-atp.tvaas.com/compass/RegisterEvent?type=preview&recommended=false&device_model=PC&client=json&external_content_id=309`

This registers a preview event for the content with ID 309.

A sample response is shown below, showing that the event was successfully registered.

```
{
  "metadata": {
    "request": "RegisterEvent",
    "timestamp": 1467876569679
  },
  "response": {
    "message": "Event registered successfully",
    "properties": {},
    "status": "SUCCESS"
  }
}
```

*

198 Register Event: Content End

198.1 Overview

An event should be registered with COMPASS when a user plays a movie to its end (or close to its end, to take into account that viewers won't watch the credits through to the end).

198.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<code>type</code>	The type of event, must be set to <code>content_end</code> .
<code>external_content_id</code>	The external ID of the content.
<code>recommended</code>	Set to <code>true</code> or <code>false</code> , if the movie was clicked from a list of recommendations or not.

The following are only relevant if `recommended` is set to `true`:

<code>shuffle_num</code>	The number of shuffles performed by the user.
<code>blend</code>	The blend used when creating the recommendations list.
<code>engine</code>	Specifies the recommendation engine that generated the recommendation, such as <code>most_popular</code> or <code>user_rating</code> .
<code>device_model</code>	The type of device on which the device is to be played.

For more parameters, see the Compass RT API Guide.



Response: **Result** (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
<code>status</code>	Indicates success or failure.

198.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

198.4 Example

`https://<operator>.tvaas.com/compass/RegisterEvent?type=content_end&recommended=false&device_model=PC&client=json&external_content_id=Argo`

This registers a content end event for video Argo.

A sample response is shown below, showing that the event was successfully registered.

```
{
  "metadata": {
    "request": "RegisterEvent",
    "timestamp": 1467876569679
  },
  "response": {
    "message": "Event registered successfully",
    "properties": {},
    "status": "SUCCESS"
  }
}
```



199 Register Event: Rating

199.1 Overview

An event should be registered with COMPASS when a user rates content.

199.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<i>type</i>	The type of event, must be set to <code>rate</code> .
<i>external_content_id</i>	The external ID of the content.

For more parameters, see the Compass RT API Guide.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates <code>success</code> or <code>failure</code> .

199.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

199.4 Example

`https://<operator>.tvaas.com/compass/RegisterEvent?type=rate&recommended=false&device_model=PC&client=json&external_content_id=The+Revenant&rating=5`

This registers a rating event for the movie The Revenant.

A sample response is shown below, showing that the event was successfully registered.



```
[{"metadata": {  
    "request": "RegisterEvent",  
    "timestamp": 1467876569679  
},  
"response": {  
    "message": "Event registered successfully",  
    "properties": {},  
    "status": "SUCCESS"  
}  
}
```

200 Register Event: Set Reminder

200.1 Overview

An event should be registered with COMPASS when a user sets a reminder for content.

200.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<i>type</i>	The type of event, must be set to <code>reminder</code> .
<i>program_id</i>	ID of program.
<i>recommended</i>	Set to <code>true</code> or <code>false</code> , if the program was clicked from a list of recommendations or not.

The following are only relevant if *recommended* is set to `true`:

<i>shuffle_num</i>	The number of shuffles performed by the user.
<i>blend</i>	The blend used when creating the recommendations list.
<i>engine</i>	Specifies the recommendation engine that generated the recommendation, such as <code>most_popular</code> or <code>user_rating</code> .
<i>device_model</i>	The type of device on which the device is to be played.

For more parameters, see the Compass RT API Guide.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.



Property	Description
<code>status</code>	Indicates success or failure.

200.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

200.4 Example

`https://<operator>.tvaas.com/compass/RegisterEvent?type=reminder&recommended=false&device_model=PC&client=json&external_content_id=In+Her+Shoes&rating=5`

This registers a reminder event for the program In Her Shoes.

A sample response is shown below, showing that the event was successfully registered.

```
{
  "metadata": {
    "request": "RegisterEvent",
    "timestamp": 1467876569679
  },
  "response": {
    "message": "Event registered successfully",
    "properties": {},
    "status": "SUCCESS"
  }
}
```

201 Register Event: Channel Zap

201.1 Overview

An event should be registered with COMPASS when a user selects a channel and keeps it open for a predefined length of time.

201.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers an event with COMPASS. Significant parameters include:

Parameter	Description
<code>type</code>	The type of event, must be set to <code>zap</code> .
<code>program_id</code>	ID of program.
<code>recommended</code>	Set to <code>true</code> or <code>false</code> , if the program was clicked from a list of recommendations or not.

The following are only relevant if `recommended` is set to `true`:

<code>shuffle_num</code>	The number of shuffles performed by the user.
<code>blend</code>	The blend used when creating the recommendations list.
<code>engine</code>	Specifies the recommendation engine that generated the recommendation, such as <code>most_popular</code> or <code>user_rating</code> .
<code>device_model</code>	The type of device on which the device is to be played.

For more parameters, see the Compass RT API Guide.



Response: **Result** (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
<code>status</code>	Indicates success or failure.

201.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

201.4 Example

`https://<operator>.tvaas.com/compass/RegisterEvent?type=zap&recommended=false&device_model=PC&client=json&external_content_id=In+Her+Shoes&rating=5`

This registers a channel zap event for the program In Her Shoes.

A sample response is shown below, showing that the event was successfully registered.

```
{
  "metadata": {
    "request": "RegisterEvent",
    "timestamp": 1467876569679
  },
  "response": {
    "message": "Event registered successfully",
    "properties": {},
    "status": "SUCCESS"
  }
}
```

202 Register Search Event

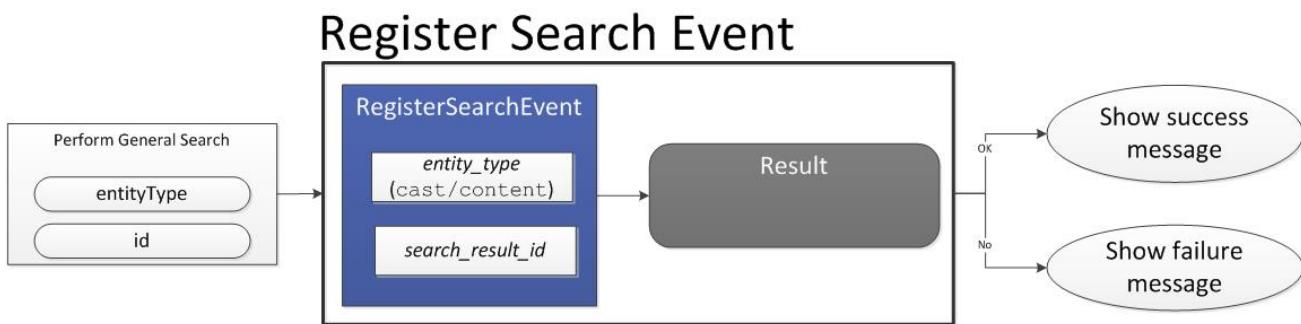
202.1 Overview

The suggestion algorithm can be continually improved by informing the system which of the search results the user actually selected. The system will take this information into account in future searches.

It is highly recommended to include this functionality within the client application.

Note for Analytics: The Analytics Platform extracts data sent when selecting a search result.

202.2 Requests and Responses



Request: [RegisterSearchEvent](#) (from COMPASS Search RT API)

Informs the system which of the returned search results was selected.

Parameter	Description
<i>entity_type</i>	The type of entity selected, e.g. content.
<i>search_result_id</i>	The ID of the result selected. Corresponds to the <i>id</i> parameter in <i>SearchResults</i> and in <i>SuggestResult</i> .

For more information, see the COMPASS Search RT API Developers Guide.

Response: [Result](#) (from COMPASS RT API)

Indicates whether the *RegisterSearchEvent* action was successful or not, and returns an error message if not.

Property	Description
<i>Status</i>	Indicates whether the request resulted in a SUCCESS or FAILURE.
<i>Message</i>	A short message from the server

202.3 Related Tutorials

To	Refer to Tutorial
Learn about registering events	195 Registering Events for Content Discovery (p.527)

202.4 Example

`https://<operator>.tvaas.com/search/compass/RegisterSearchEvent?entity_type=content&search_result_id=22335&client=json`

This request informs the Service Delivery Platform that the search result 22335 was selected.

The value 2235 is taken from the *SearchResults* or *SuggestResult* responses, as in the example below.

```
{  
    "contentSeq": "3",  
    "prLevel": "0",  
    "year": "2011",  
    "entityType": "content",  
    "seriesName": "Glee S2",  
    "externalRating": "8.3",  
    "availabilities": [  
        {  
            "externalId": "Glee S3 Ep3"  
        }  
    ],  
    "externalId": "Glee S3 Ep3",  
    "genres": [  
        {  
            "name": "Drama"  
        },  
        {  
            "name": "Comedy"  
        },  
        {  
            "name": "Music"  
        }  
    ],  
    "seasonName": "3",  
    "name": "Glee S3 Ep3",  
    "id": "22335",  
    "contentType": "episode",  
    "isCatchup": 0  
},
```



203 Register Click Event

203.1 Overview

The suggestion algorithm can be continually improved by informing the system whenever the user clicks on a content item, whether it be from a catalog screen, a list of promotions, a list of recommendations, a banner or marketing campaign, etc.

Clicking a result from a list of search results is handled differently, see [202 Register Search Event \(p.542\)](#).

Note for Analytics: The Analytics Platform uses the data sent when registering clicks for constructing many of the charts and KPIs. In particular, this information is used when attributing eventual orders to order sources such as search, recommendations, promotions, etc.

203.2 Requests and Responses

Request: [RegisterEvent](#) (from COMPASS RT API)

This call registers a click event. The following parameters must be provided:

Parameter	Description
<i>type</i>	The type of event, must be set to click.
<i>external_content_id</i>	If a movie or TV Show season was clicked, specify its ID using this parameter.
<i>external_series_id</i>	If a TV Show series was clicked, specify its ID using this parameter. <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> Note Only one of external_content_id and external_series_id can be specified. </div>
<i>recommended</i>	Set to true if the video was clicked from a list of recommendations or promotions. Otherwise set to false.
The following are only relevant if recommended is set to true:	
<i>engine</i>	The recommendation engine that generated the recommendation, such as most_popular, user_rating or promotions.



Parameter	Description
<i>blend</i>	The blend used when creating the recommendations list.
<i>device_model</i>	The model ID of the device on which the click occurred.
<i>text</i>	The name of the screen on which the click occurred.

Note

When registering a click event on a banner or marketing campaign, use the parameter values provided in [203 Register a Banner Click Event \(p.547\)](#).

Response: Result (from COMPASS RT API)

Indicates whether the action was successful or not, and returns an error message if not.

Property	Description
<i>status</i>	Indicates success or failure.

203.3 Example

`https://<operator>-atp.tvaas.com/compass/RegisterEvent?type=click&recommended=true&engine=most_popular&device_model=PC&text=Home&client=json&external_content_id=309`

This registers a click event for the content with ID 309.

203.4 Related Tutorials

To	Refer To Tutorial
Register a click event on a banner or marketing campaign	203 Register a Banner Click Event (p.547)



203 Register a Banner Click Event

203.5 Overview

The Analytics Platform uses the data sent when registering clicks for constructing many of the charts and KPIs. In particular, this information is used when attributing eventual orders to order sources such as search, recommendations, promotions, banners or marketing campaigns, etc.

This section provides the parameter values to be used when registering a click on a banner or marketing campaign.

For general information on registering click events, see [203 Register Click Event \(p.545\)](#).

203.6 RegisterEvent Parameter Values

When using the *RegisterEvent* call to register that the user clicked on a banner or marketing campaign, use the values listed below for the *external_content_id*, *recommended*, and *engine* values.

Banner Type	RegisterEvent Parameter Values		
	external_content_id	recommended	engine
Content or episode	external_content_id	false	content_banner
Season	external_season_id	false	season_banner
Series	external_series_id	false	series_banner
Program	external_program_id	false	program_banner
VOD Category	external_vod_category_id	false	vod_category_banner
Channel	external_channel_id	false	channel_banner

Banner Type	RegisterEvent Parameter Values		
	external_content_id	recommended	engine
Product – SVOD	external_product_id	false	svod_banner
Product – VOD Package			vod_package_banner
Product - VOD card			vod_card_banner
Product - Catchup			catchup_banner
Product - Channel Package			channel_package_banner
Product - NPVR			npvr_banner
Product - Startover			startover_banner



204 Providing Support for Chrome v59 Onward

204.1 Procedure

Follow the process below for license acquisition when playing encrypted content through Chrome v59 and later:

1. When the portal is launched, the certificate should be received from https://wv-certificate.tvaas.com/cert_uat_widevine_com.bin.
2. When calling `setServiceCertificate`, provide the URL of the certificate to the `MediaKeys` object. For more information see <https://w3c.github.io/encrypted-media/#dom-medialkeys-setservercertificate>.
3. Acquire the license as usual.

205 Apps and Analytics

205.1 Why Analytics?

Like any business, effective Pay TV Operators continually evaluate their operations so as to maximize revenue and reduce costs. Which contents to purchase and how to market them are only two of many business decisions that Operators need to make on an ongoing basis.

TVaaS provides the Analytics Platform to enable Operators to make such decisions effectively. The Analytics Platform presents a series of dashboards providing actionable intelligence based on which Operators can make sound, evidence-based decisions.

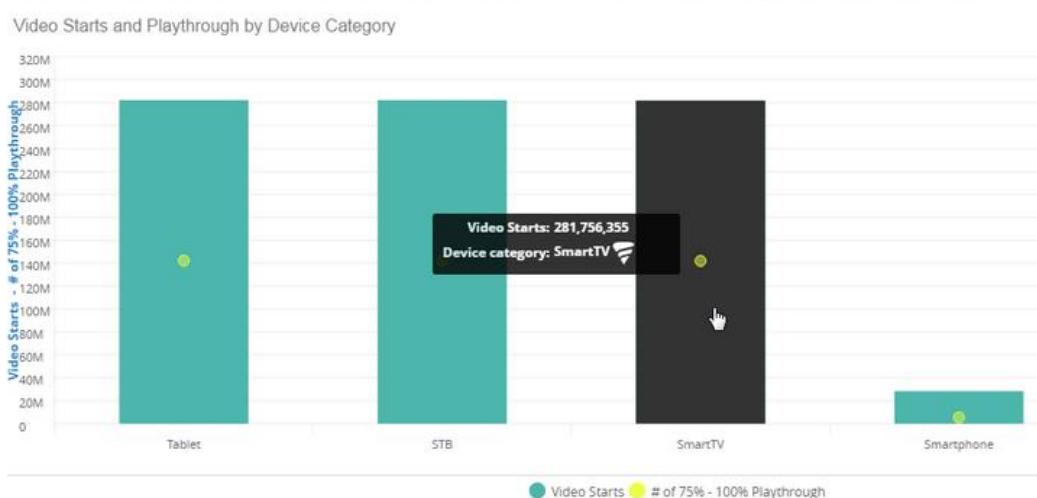
Most of the intelligence presented in the Analytics Platform's charts and KPIs is based on data collected on customer actions, such as ordering products, watching videos, and clicking through the catalog.

205.2 The Role of the Apps

The client apps play a central role in the collection of accurate data on which the Analytics charts and KPI's are based. The phrase "garbage in, garbage out" certainly applies. All relevant customer actions in the apps must be registered to provide the data used by the Analytics Platform.

For example, the chart below is designed to give the Operator an understanding of how engaged customers are by the content offered in general, and how viewing patterns differ per device category. The Analytics Platform relies on the apps to register all video start and stop events to be able to create this chart.

Chart: Video Starts and Playthrough by Device Category



This chapter provides details of all events that must be registered. Failure to follow these guidelines when developing a client application effectively renders the charts and KPIs presented in the Analytics Platform meaningless. See [205.3 Registered Events \(p.551\)](#).

This chapter also provides additional guidelines that are not related to specific events. See [205.4 Additional Guidelines \(p.558\)](#).

205.3 Registered Events

The following events must be registered to ensure that the apps provide the raw data required by the Analytics Platform.

- [205.3.1 Ordering a Video \(p.551\)](#)
- [205.3.2 Ordering a VOD Package \(p.552\)](#)
- [205.3.3 Watching Videos \(p.553\)](#)
- [205.3.4 Click Events \(p.555\)](#)
- [205.3.5 Searching Content \(p.557\)](#)

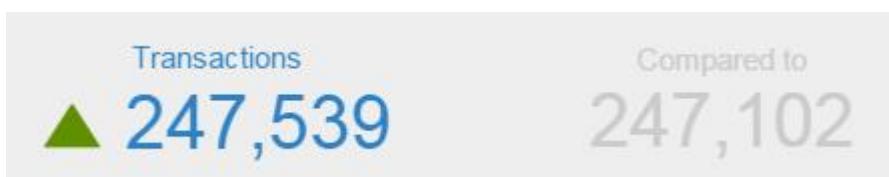
Details on each event are presented below.

205.3.1 Ordering a Video

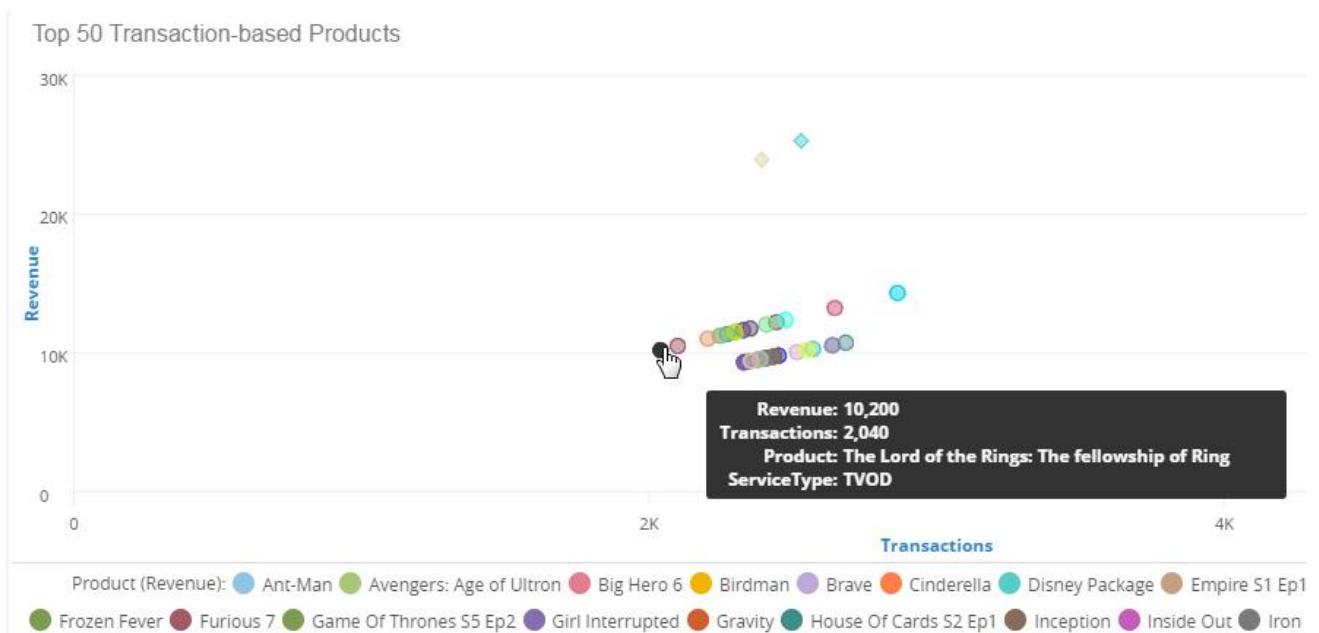
205.3.1.1 Usage in Analytics

Data on events in which customers place orders for videos is used in the construction of the majority of the charts and KPIs in the Analytics Platform.

For example, the KPI below shows the number of transactions (orders) registered during a selected time period, compared to the number of transactions in the previous period.



Similarly, the chart below plots the number of orders placed for the most popular movies against the revenue generated by each.



205.3.1.2 Implementation

Movies are ordered using the *OrderVideo* action in the RiGHTv Front-end API.

Order data is collected whenever a movie is ordered. This includes situations in which:

- The customer orders a video through a TVOD offer, meaning that the video is ordered on its own, for either rent or buy.

OrderVideo should be called only once per ordered movie, and should be called after the user has selected a purchase method and finalized the purchase.

- The customer plays a video offered through an SVOD service to which the customer is already subscribed.

OrderVideo should be called only once per movie in the SVOD service. It should be called only when the user plays the movie for the first time. First check if a ticket for the movie exists, using *GetVideoTicketList*. Only call *OrderVideo* if no ticket for the video is found in the response.

Instructions for implementing *OrderVideo* are provided in the sections referenced below:

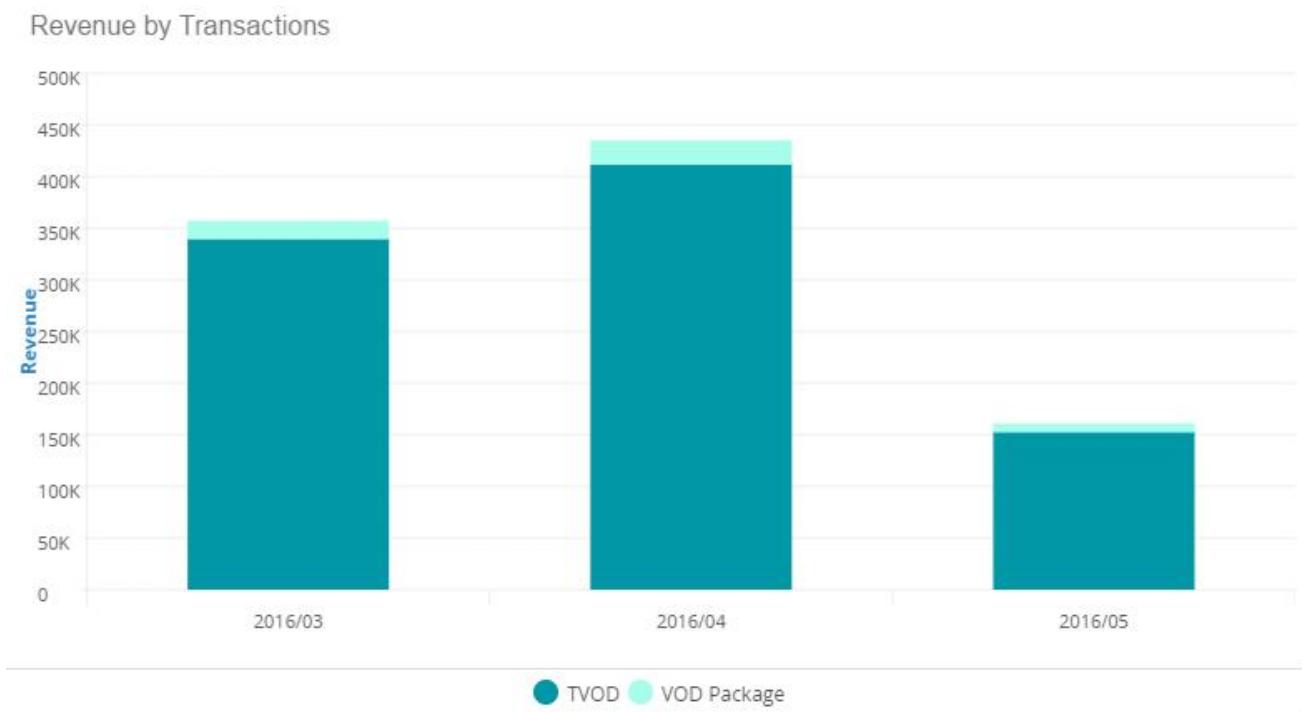
Business Model	Reference
TVOD	128 Ordering TVOD Content (p.336)
SVOD	135 Opening a Ticket for an SVOD Movie (p.357)

205.3.2 Ordering a VOD Package

205.3.2.1 Usage in Analytics

Data on events in which customers order VOD packages is used in the construction of a number of charts and KPIs in the Analytics Platform.

For example, the chart below compares the revenues generated by orders for TVOD and VOD Package products.



205.3.2.2 Implementation

VOD Packages are ordered using the *OrderVideoPackage* action in the RiGHTv Front-end API.

Order data is collected whenever a VOD Package is ordered.

Instructions for implementing *OrderVideoPackage* are provided in [129 Ordering a VOD Movie Package \(p.341\)](#).

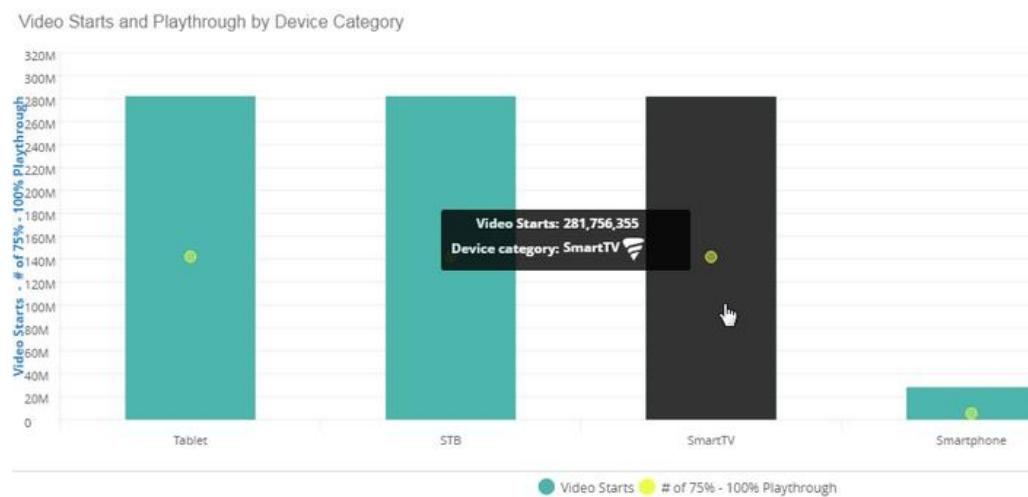
205.3.3 Watching Videos

205.3.3.1 Usage in Analytics

Various Charts and KPIs in the Analytics Platform are designed to provide the Operator with an understanding of customers' content preferences and viewing habits.

The Analytics Platform chart below is designed to give the Operator an understanding of how engaged customers are by the content offered in general, and how viewing patterns differ per device category.

Chart: Video Starts and Playthrough by Device Category



205.3.3.2 Implementation

Customer viewing data is collected by monitoring the API calls sent when:

- A movie is played (when the action *GetVideoPlayingInfo* is called). Instructions for implementation are provided in [49 Get Movie Playback URL \(p.151\)](#).

Note

GetVideoPlayingInfo must be sent without using caching.

- The position in a movie being watched is recorded (when the action *MarkPosition* is called).

The *MarkPosition* call is used by the Analytics Platform as an indication that the user has ended the current viewing session.

MarkPosition must thus only be called when:

- The user closes the video player
- The stream is stopped.
- The end of the movie has been reached, even if the user did not stop the content.

MarkPosition must not be called when:

- The content is paused.

When sending *MarkPosition*, the current position in the video must be specified. Use *GetVideoPlayingInfo* to obtain that information.

Note

MarkPosition must be sent without using caching.

Instructions for implementation are provided in [51 Mark Last Position \(p.155\)](#).

There is a need to call it on every play and stop without using caching. End of movie should send *Mark position* automatically as well even if the user did not stop the content.

205.3.4 Click Events

205.3.4.1 Usage in Analytics

A user's decision to order a content item can be attributed to prior exposure to promotions or recommendations for that content. Users may also reach the content on their own from other locations, such as through the catalog or search results.

Knowing the effectiveness of various order conversion sources can help Operators understand the value of those sources and improve their use of those sources in future.

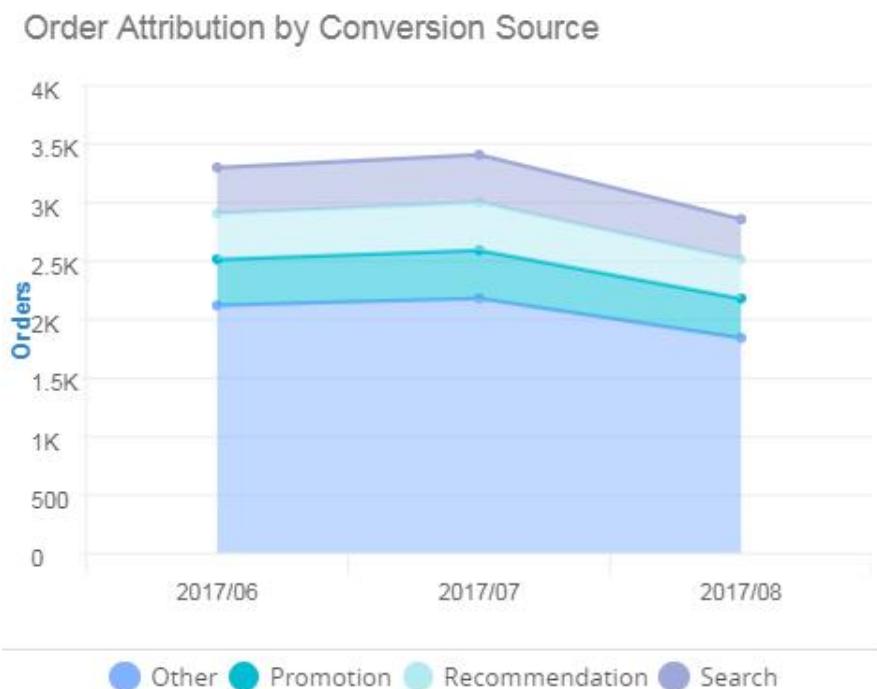
The Analytics Platform does not only consider the last event that occurred prior to the order when attributing the order to conversion sources, but rather examines all recent events that likely contributed to the decision to order.

The Analytics Platform provides multiple charts that display the relative effectiveness of each conversion source, for example:

- [205.3.4.2 Chart: Order by Conversion Source \(p.555\)](#)
- [205.3.4.3 Chart: Attribution by Conversion Source and Service \(p.556\)](#)

205.3.4.2 Chart: Order by Conversion Source

This chart indicates number of orders generated by different conversion sources over time, as well as the relative effectiveness of different conversion order sources.



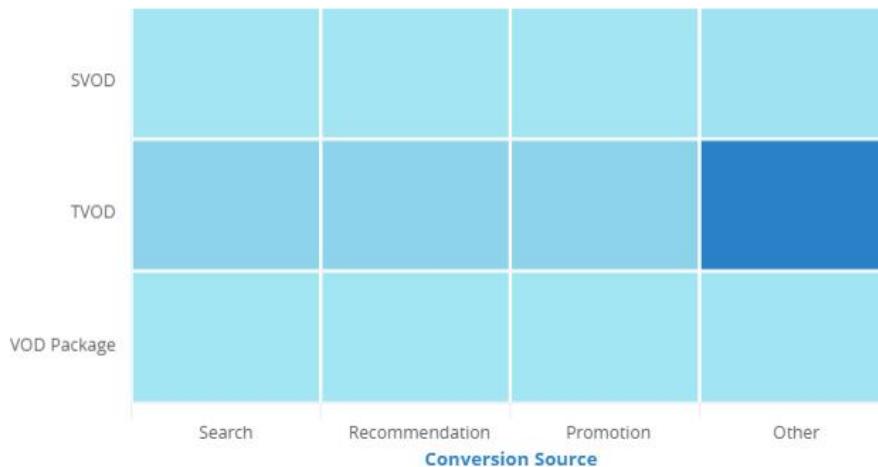
The chart is designed to help Operators answer the following questions:

- What is the most effective conversion source for generating orders?
- Did the relative effectiveness of conversion sources change over time?

205.3.4.3 Chart: Attribution by Conversion Source and Service

This chart is designed to indicate the effectiveness of each conversion source in generating orders of various business models (SVOD, TVOD, VOD Card, VOD Package). A darker block indicates that a relatively higher number of orders for the particular business model were attributable to the particular conversion source.

Attribution by Conversion Source and Service



The chart is designed to help Operators answer the following questions:

- For each business model, which conversion source was the most successful in generating orders?
- Which conversion source was the most successful in general?

205.3.4.4 Implementation

The two attribution charts described above are based on data collected each time a logged-in user clicks a content item within the app. Whenever the user clicks a content item anywhere in the app, that click must be registered. Examples include clicking content items in:

- The catalog
- A recommendations list
- A promotions list
- A list of episodes displayed on a TV Show season screen

The following information is collected for each click:

- The ID of the logged-in user
- The ID of the content item that was clicked
- The context of the click, i.e. whether the user clicked in a recommendations list, a promotions list, or the catalog
- If the click was on a recommended content item, the engine or blend used to generate the recommendation must be submitted
- The time
- The device type used
- The name of the screen in the application in which the click occurred

Clicks on all types of content items must be registered, including:

- Movies
- TV Show episodes
- TV Show series
- TV Show seasons
- Live programs

Follow the implementation instructions at [203 Register Click Event \(p.545\)](#) for registering clicks on VOD content (TVOD and SVOD movies and TV Show episodes) and on Live content (live programs).

Note

Clicking a result from a list of search results is handled differently, see [205.3.5.2 Clicking a Search Result \(p.557\)](#).

205.3.5 Searching Content

Analyzing users' content search terms provides the Operator with valuable insight into what contents users are interested in purchasing. By further analyzing the results of those searches the Operator can identify which searched content is currently provided and which content is currently missing from the offering.

This section provides guidelines for ensuring that the Operator's apps correctly provide the required data needed for search analysis.

Guidelines are provided for the following:

- [205.3.5.1 Searching for Videos \(VOD\) \(p.557\)](#)
- [205.3.5.2 Clicking a Search Result \(p.557\)](#)

Note

The Analytics Platform does not currently analyze searches performed for live programs (using *SearchLivePrograms*).

205.3.5.1 Searching for Videos (VOD)

Video searches are performed using the *SearchVideos* request. No additional event needs to be registered. For implementation instructions, see [147 Performing a Video Search \(p.390\)](#).

The Analytics Platform analyzes all *SearchVideo* requests received. The search terms are recorded, as are the results of the search:

- If no results were found, the search term is considered to be a zero result search term.
- If results were found, but the user did not click on any of them, the search term is considered to have resulted in irrelevant results only.

205.3.5.2 Clicking a Search Result

When the user clicks a search result, that action must be registered. The Analytics Platform requires this information to be able to determine whether a search performed found the content that the user was searching for. For implementation instructions, see [202 Register Search Event \(p.542\)](#).



The Analytics Platform automatically differentiates between results clicked from a regular search and from results clicked from search auto-suggestions, if implemented (see [148 Adding Auto-suggestion to Search \(p.397\)](#)), based on the search ID provided when registering the click on the search result.

If the user clicks a search result that is a TV Show season, that click should be registered as explained above (see [202 Register Search Event \(p.542\)](#)). However, if an episode within that season is clicked later, that event should not be registered as a search event. Instead, a regular click event should be registered (see [203 Register Click Event \(p.545\)](#)).

205.4 Additional Guidelines

This section provides additional guidelines that must be followed during app development, that are not related to specific events.

205.4.1 TV Show Implementation

The apps must make use of the new TV Shows model for implementing VOD TV shows. Using the previous Series-Season model will prevent the collection of the data on which TV show analytics is based.