viaccess·orca

COMPASS

intelligent content discovery

Version 3.1 SR22

RT API Developers Guide

Oct 2022

Table of Contents

Introduction 6
Copyright 6
Trademarks 6
Intended Audience 6
Required Skills 6
Conventions 6
Deprecated Functionality 7
Technical API Overview 8
Actions 8
Entities 8
Protocol 8
URL syntax 8
Response Types 9
Response Format 9
Authentication 9
Profile Identity 10
Social Identity 10
Multilingual Support 10
Paging 10
General Error Codes 11
HTTP Response Codes 11
Calculating URL Signature 11
XML Formatter Reference 13
General 13
Single Entity Response 13
List of Entities Response 13
Result Object Response 14
Exception Response 15
JSON Formatter Reference 16
General 16
General Response Format 16
Single Entity Response 16
List of Entities Response 16
Result Object Response 17
Exception Response 17
JavaScript Code Sample 17
Entities 19
BroadcastAvailability 19
Cast 19
CastPerson 20
Category 20
Channel 20
Content 21

- ContentAvailability 21
- ContentRating 22
- CustomProperty 23
- ExcludedContent 23
- ExcludedData 23
- ExcludedGenre 24
- ExploreOption 24
- ExtendedSocialIdentity 24
- Genre 25
- Image 26
- Language 26
- LikedContent 26
- LiveContentAvailability 27
- LiveRecommendation 28
- PopularContributorRecommendation 30
- Profile 31
- ProfileImage 32
- ProfileOrder 33
- ProfileRating 33
- ProfileType 34
- Property 34
- ReceivedSocialIdentity 34
- ReceivedSocialMessage 35
- RecommendationReason 35
- Result 36
- SearchSocialIdentity 37
- Season 37
- SentSocialIdentity 38
- SentSocialMessage 38
- ServiceAvailability 39
- SocialExternalIdentity 39
- SocialIdentity 39
- SocialImage 40
- VideoContentAvailability 41
- VideoRecommendation 42
- ViewedContent 43
- WishListRecommendation 44
- WishListUsageContent 45

Actions 47

- AcceptInvitation 47
- ActivateSocialIdentity 47
- AddContentToProfileExcludeList 48
- AddProfile 48
- AddToProfileExcludeList 50
- AddToWishList 51
- BlockSocialIdentity 52

CheckSocialNameAvailability 53
ClearProfileContentExcludeList 53
ClearProfileExcludeList 54
CreateExternalIdentity 54
CreateHousehold 55
CreateProfileImage 57
CreateSocialIdentity 58
CreateSocialImage 59
DeactivateSocialIdentity 60
DeleteExternalIdentity 61
DeleteFriend 61
DeleteProfile 62
DeleteProfileHistory 62
DeleteProfileImage 63
DeleteSocialIdentity 63
DeleteSocialImage 64
GetChannelList 64
GetContent 65
GetContentList 65
GetContentRatingList 66
GetDefaultGenreList 66
GetExploreOptionsList 67
GetExploredContentList 68
GetExploredLiveContentList 70
GetExternalIdentities 72
GetFriendList 73
GetFrontEndAppsConfig 74
GetGenreList 74
GetLanguageList 74
GetLiveRecommendationList 75
GetLiveSeriesAvailabilityList 79
GetProfile 81
GetProfileChannelList 81
GetProfileContentExcludeList 82
GetProfileExcludeList 82
GetProfileGenreList 83
GetProfileImage 83
GetProfileImageList 84
GetProfileImplicitChannelList 84
GetProfileImplicitGenreList 85
GetProfileList 86
GetProfileOrderList 86
GetProfileRatingList 86
GetProfileTypeList 87
GetProfileWishList 87
GetProfileWishlistUsageData 88

GetReceivedSocialMessages 89
GetReceivedSocialMessagesArchived 90
GetRecommendationListByPopularContributor 92
GetSentSocialMessages 94
GetSentSocialMessagesArchived 96
GetSeriesSeasonList 97
GetSocialIdByExtId 97
GetSocialIdentity 98
GetSocialImage 99
GetVideoRecommendationList 99
GetVideoSeasonContentList 105
GetViewedContentList 108
IgnoreInvitation 108
InviteFriend 109
IsAlive 110
Login 110
Logout 111
OptInProfile 111
OptOutProfile 112
RegisterEvent 112
RemoveContentFromProfileExcludeList 117
RemoveFromProfileExcludeList 117
RemoveFromWishList 118
RemoveProfilePinCode 119
ResetProfilePinCode 120
SearchSocialIdByExtId 120
SearchSocialIdentities 121
SendRecommendations 123
SendSocialMessage 124
UnblockSocialIdentity 126
UpdateProfile 126
UpdateProfileChannelList 128
UpdateProfileGenreList 128
UpdateProfileImage 129
UpdateProfilePinCode 130
UpdateSocialIdentity 131
UpdateSocialImage 132
UpdateSocialMessageStatus 133

Introduction

The purpose of this guide is to define the interface and the mechanism whereby external clients or portals can interact with the COMPASS runtime platform in order to enable the implementation of the user interface using various client technologies.

This chapter describes this guide's intended audience, specifies the required skills, and lists the conventions used in this guide.

Copyright

Trademarks

Intended Audience

This guide is designed for developers implementing portals or client applications which are based on or integrated with COMPASS's runtime APIs.

Required Skills

The COMPASS runtime API is designed for developers who have the following skills:

- Experience using the HTTP protocol.
- Experience using XML or JSON data formats.

Conventions Used in this Guide

The following type and symbol conventions are used in this document:

- Square brackets ([]) are used after an entity name to identify arrays (Foo[]), or around optional arguments ([bar]).
- A hyphen (-) used before properties indicates object properties.
- **Bold** used in functions indicates mandatory parameters.

- Gray used in objects indicates inherited parameters.
- `Monospace` indicates code examples and references.
- Dates given as parameters or returned in entities are always specified in milliseconds since 1970, in GMT.

Deprecated Functionality

Functionality which is deprecated is marked with a special label as part of its description. For example, *Deprecated since version 1.0.*

This means that this functionality currently exists, but will be supported in only one more major version, after the deprecation version.

For example, a function deprecated in version 1.0 will still be available in version 2.0 but will not be available in version 3.0. Processes that include this functionality should be modified and include the new relevant APIs instead.

Technical API Overview

This chapter describes the main concepts related to the usage of Orca's COMPASS runtime API.

The COMPASS runtime API can be used for the following functionality:

- Acquiring recommendations.
- Usage data collection.
- Profile management.

COMPASS runtime APIs are the main interface through which subscriber client applications gain access to all the services provided by the COMPASS recommendation platform. COMPASS is a J2EE-based application which is accessed via an HTTP-based interface.

Actions

The APIs exposed by COMPASS are referred to throughout this document as actions. There are three main types of actions:

- Authentication - These actions are used for authentication (Login) or changing the identity of an active profile.
- Data retrieval - These actions are used for the retrieval of a single data entity or a list of data entities from the server. These action names usually start with the "Get" prefix.
- Data modification actions - These actions are used for updating data in the server or execution of business logic, such as updating profile preferences or registering usage data.

Entities

COMPASS's data model is constructed from entities which define a recommendation model that can be integrated with any middleware service. These entities such as Profile, LiveRecommendation and VideoRecommendation, which are described in the entities section of this document, are retrieved using the data retrieval actions mentioned above.

Protocol

The COMPASS runtime APIs are based on the HTTP 1.1 protocol, and support the inherent cache control mechanism of the HTTP 1.1 protocol. All actions can be accessed using HTTP GET or POST, although GET is the preferred access method for data retrieval actions. POST is used only in cases where data is sent to the server as parameters of data modification actions, such as the `UpdateProfile` action.

URL syntax

http://<host name>:<port>/compass/<action name>?parameters

For actions that are related to SocialTV (marked with SocialTV), the URL syntax is:

http://<host name>:<port>/compass/social/<action name>?parameters

Example:

```
http://127.0.0.1:8080/compass/Login?household_id=1&language_code=en
```

```
http://127.0.0.1:8080/compass/social/GetSocialIdentity
```

See the following sections for more details regarding general or specific action parameters.

Response Types

Each action call results in one of the following responses:

- A single entity - When calling a data retrieval action defined as returning a single entity.
- A list of entities - When calling a data retrieval action defined as returning a list (array) of entities.
- A result object - When calling authentication or data modification actions.
- An exception object - In case a system or applicative error occurs during the action call.

Response Format

The COMPASS runtime APIs can support multiple response formats used for supporting multiple client types. By default, the following formats are supported:

- JSON - Used mostly for JavaScript based clients.
- XML - Used for Java or other technology-based clients.

Additional formats also can be supported per specific integration requirements. The required response format is defined by adding the client parameter to any action call. The value of the parameter should be either json or xml. In case a specific response format is defined for a specific client integration, the name of the client should be used. See XML and JSON format sections for specific details.

Authentication

Prior to any action call, the client can be authenticated by COMPASS.

The authentication of users in COMPASS enables defining and managing a personal profile, and based on this profile receiving personal recommendations or communicating with other users in the COMPASS SocialTV Network. Users that are not authenticated (referred to as anonymous users) do not have a profile and get only non-personal recommendations. They cannot use the SocialTV capabilities of COMPASS.

The authentication is performed by calling the Login action and providing identification for the household and profile. If the profile is protected with a PIN code, the user must supply the correct PIN code in order to be successfully authenticated.

The authentication method is a plug-in. In its default implementation, the Login process expects to identify the household and profile by their ID in the middleware. Once the authentication is performed successfully, an encrypted session cookie is generated by COMPASS and is used to identify the calling client in all subsequent action calls.

As of COMPASS 2.1, there is an additional option to receive the identity as a string in the Login action result and send it as a parameter named *compass_identity* in all subsequent action calls. This enables support for clients that cannot use cookies.

SocialTV Authentication

Authenticated users can create a Social Identity that will be associated with the logged-in profile. In the default implementation, if the logged-in profile has an associated Social Identity, the Social Identity is stored in the profile cache and then used in all SocialTV action calls. If the logged-in profile does not have an associated Social Identity, an exception is raised when calling a SocialTV action.

Profile Identity

The profile identity represents the active profile. Once the authentication process is successfully performed, a profile identity is created. The identity contains the channels and services to which the household is subscribed. This information is used for verification that the recommended content is available for consumption.

The profile can be protected with a PIN code that must be supplied during the authentication.

Social Identity

The Social Identity represents an active profile in the COMPASS SocialTV network. Only one Social Identity can be associated with a single profile. The Social Identity enables the user to communicate with other users in the SocialTV Network, create a friends list of other Social Identities, sends recommendations on content items to friends and more.

The Social Identity can be associated with the user profiles in other social networks, such as Facebook. This association can be used to connect from the social network to COMPASS, import friends from the external network into the COMPASS SocialTV network and more.

Multilingual Support

All data retrieval actions can return the data in one of the languages defined in the system. The desired data language is specified explicitly by adding the **language_code** parameter to each one of the data retrieval action calls. The value of this parameter is the ISO code of the desired language, one of the languages defined in the system.

Paging

In order to support the retrieval of large lists, all data retrieval actions which return a list of entities support the paging mechanism, which allows the retrieval of a specific part of the list (sub-list). In order to enable the paging option, the following parameters are added to the actions call (both parameters must be supplied):

- from - A zero-based index of the item in the list that is returned as the first item of the sub-list.

- count - The number of items to be returned in the sub-list.

In the case when the number of items that appears in the list from the index specified by the **from** parameter until the end of the list, is less than the value specified by the **count** parameter, the sub-list's item count is less than that specified by the **count** parameter.

General Error Codes

As mentioned above, any error that occurs during an action execution results in an exception object response. The exception object contains the properties: **code** and **message**. The valid values for the code property per action type are listed in the **Throws** section of each action's documentation. In addition, the following general error codes may be thrown:

- GENERAL_SERVER_ERROR - A general server error occurred.
- MISSING_PARAMETER - In case a mandatory parameter was not passed to an action.
- INVALID_PARAMETER - In case an invalid parameter was passed to an action.
- INVALID_PARAMETER_VALUE - In case an invalid parameter value was passed to an action.
- INVALID_CREDENTIALS - In case there is an error authenticating the user (this usually happens when an action is performed without logging into COMPASS).

Before using SocialTV actions, the user must be authenticated. If the action is invoked without authenticating first, or the logged-in profile does not have an associated Social Identity, an INVALID_CREDENTIALS error is thrown.

HTTP Response Codes

By default, all action calls are responded to with the 200 OK response code, including cases where the response is of an exception type.

There might be some cases where an unexpected error occurs in the server, which results in one of the 5xx HTTP error codes. In case of cache-control validations, relevant calls are responded to with the 304 Not Modified response code.

Calculating URL Signature

Some of the APIs are designed to be used by external applications, such as Facebook applications. These applications need to perform some operations before the user is logged-in to the system. In order to make sure that only trusted applications are performing these operations, a signature parameter is calculated by the client and sent for some APIs. The sender application must generate the signature by using a simple DES encryption algorithm, using a secret key defined in the server that should be shared with the external application. If the system cannot validate the signature, an exception is raised.

Signature Calculation

1. Sort the parameters alphabetically by key.
2. Concatenate all key/value pairs together in the format "k=v" (omitting the signature parameter itself, since that is what we are calculating).
3. Use DES/ECB/PKCS5Padding encryption on the concatenated string, using the secret key.
4. BASE64: Encode the result using UTF-8.
5. Append the signature parameter to the request.

For example, consider the following request, where the secret key is `12345678`.

```
http://<host name>:<port>/compass/<action name>?
network_type=facebook&external_id=1001
```

- The sorted parameters: `external_id=1001, network_type=facebook`
- Concatenated string: `external_id=1001network_type=facebook`
- Encrypted string (before BASE64 encoding) should be: `EyZjJZTrowgvKcuJAf3UXxwrLR0dfwRGrjkt4+syA0n2nnJWr4sOoQ==`
- Signature (binary): `[B@356f144c`
- Signature (string): `RXlaakpaVHJvd2d2S2N1SkFmM1VYeHdyTFIwZGZ3Ukdyamt0NCtzeUEwbjJubkpXcjRzT29RPT0=`

XML Formatter Reference

General

The XML format is intended to be used by clients implemented by various technologies with the capabilities of XML parsing. Each entity in the response is represented as an XML element. Simple properties (string or numeric) of the entity are represented as attributes of the element. In case the type of a property is an array of another entity type, the property is represented as a child element, which contains child elements representing the items in the array (see examples below).

Single Entity Response

A single entity response always contains a single entity as the root element of the response.

Example

```
http://127.0.0.1:8080/compass/GetProfile?client=xml&language_code=en

<Profile id="1" name="John Smith" image="/avatars/Flower.jpg"
suspended="false" />
```

List of Entities Response

The root element of an entity list response is always a LIST having child elements representing the entities in the list. The LIST element contains a **fullLength** attribute which represents the number of items in the full list, regardless of whether paging parameters were applied.

Example

```
http://127.0.0.1:8080/compass/GetVideoRecommendationList?client=xml

<LIST fullLength="3" timestamp="1237118001710">

    <VideoRecommendation id="212" name="Die Hard" rating="4.2" type="cf"
prLevel="1300" prName="PG-13">

        <availabilities>

            <ServiceAvailability videoId="3648" serviceId="1" />

        </availabilities>

        <images>

            <Image name="image" value="/Action/Die_Hard.jpg" />

            <Image name="imagesmall" value="/Action/Die_Hard_Small.jpg" />
        </images>

    </VideoRecommendation>

    <VideoRecommendation id="217" name="The Dark Knight" rating="4.5" 
type="most_popular" prLevel="1500" prName="NC-17">

        <availabilities>

            <ServiceAvailability videoId="3459" serviceId="1" />

        </availabilities>

        <images>

            <Image name="image" value="/Action/The_Dark_Knight.jpg" />

        </images>

    </VideoRecommendation>

    <VideoRecommendation id="2" name="WALL-E" rating="4.3" 
type="promotions" prLevel="500" prName="G">

        <availabilities>

            <ServiceAvailability videoId="1743" serviceId="1" />

        </availabilities>

        <images>

            <Image name="image" value="/Animations/WALL-E.jpg" />

        </images>

    </VideoRecommendation>

</LIST>
```

Result Object Response

A result response contains a single **Result** element having two attributes:

- status - Result status, which is usually SUCCESS.
- message - An additional informative message.

Example

```
http://127.0.0.1:8080/compass/Login?household_id=1&profile_id=1&client=xml

<Result status="SUCCESS" message="'Login' action finished successfully" />
```

Exception Response

An exception response contains a single **Exception** element having two attributes:

- code - One of the valid error codes as defined in the Throws section of each action.
- message - An additional informative message.

Example

```
http://127.0.0.1:8080/compass/Login?
household_id=111&profile_id=1&client=xml

<Exception Code="INVALID_CREDENTIALS" Message="No such household: 111" />
```

JSON Formatter Reference

General

JSON format is commonly used for loading data from the server and converting it directly into JavaScript objects (AJAX).

For JSON format specifications see http://www.json.org/.

General Response Format

When using the JSON response format, a JavaScript object is created by applying the **eval** function on the responseText. The object created always contains two properties:

- response - The actual response to the action call which can be a single entity, a list of entities, a result object or an exception. See the following sections for specific formats.
- metadata - Metadata accompanying the action result, such as the timestamp and the name of the action. In case the action returns a list, the full length is also part of the metadata.

Single Entity Response

The response property contains a single entity as defined for the return type of the called action.

Example

```
http://127.0.0.1:8080/compass/GetProfile?client=json&language_code=en


{"response":{"id":1,"name":"John Smith","image":"/
avatars/Flower.jpg","suspended":0},"metadata":
{"timestamp":1237195642000,"request":"GetProfile"}}
```

List of Entities Response

The response property contains an array of entities as defined for the return type of the called action.

Example

```
http://127.0.0.1:8080/compass/GetVideoRecommendationList?client=json

{"response": [{"id":212,"name":"Die Hard","images":
[{"name":"image","value":"/Action/Die_Hard.jpg"},
{"name":"imagesmall","value":"/Action/
Die_Hard_Small.jpg"}],"rating":"4.2","prName":"PG-13","prLevel":"1300",
"type":"cf","availabilities":[{"serviceId":"1","videoId":"3648"}]},
{"id":217,"name":"The Dark Knight","images":[{"name":"image","value":"/
Action/The_Dark_Knight.jpg"}],"rating":"4.5","prName":"NC-17",
"prLevel":"1500","type":"most_popular","availabilities":
[{"serviceId":"1","videoId":"3459"}]}, {"id":2,"name":"WALL-E",
"images":[{"name":"image","value":"/Animations/WALL-E.jpg"}],
"rating":"4.3","prName":"G","prLevel":"500","type":"promotions","availabilities":
[{"serviceId":"1","videoId":"1743"}]}], "metadata":
{"timestamp":1237196154244,"request":"GetVideoRecommendationList","fullLength":3}}
```

Result Object Response

The response property contains a single result object which contains the following properties:

- message - An additional informative message.
- status - The result status, which is usually SUCCESS.
- properties - Any properties that the action returns. For example, if when creating a profile, this contains the ID of the new profile.

Example

```
http://127.0.0.1:8080/compass/AddProfile?name=George&client=json

{"response":{"message":"Add profile operation
finished successfully","status":"SUCCESS","properties":
{"profileId":"7"}},"metadata":
{"timestamp":1237203431703,"request":"AddProfile"}}
```

Exception Response

The response property contains a single exception object which contains the following properties:

- code - One of the valid error codes as defined in the Throws section of each action.
- message - An additional informative message.

Example

```
http://127.0.0.1:8080/compass/Login?
household_id=111&profile_id=1&client=json

{"response":{"message":"No such household:
111","status":"FAILURE","properties":
{},"code":"INVALID_CREDENTIALS"},"metadata":
{"timestamp":1237203548800,"request":"Login"}}
```

JavaScript Code Sample

```
var url = "http://127.0.0.1:8080/compass/GetProfile?
client=json&language_code=en";

var request = new XMLHttpRequest();

request.open("GET", url, true);

request.onreadystatechange = function() {
  if (request.readyState == 4) {

    if (request.status == 200 || request.status == 304) {

      var response = eval(request.responseText); // convert the response
text to javascript objects

      // do something with response

    }

  }

}

request.send(null);
```

Entities

This chapter describes the available entities in COMPASS.

BroadcastAvailability

since 2.1

This entity represents the availability of live content.

Properties

Name	Type	Description
programId	Number	The external ID of the program.
channelId	Number	The external ID of the channel in which the program is being broadcast.
startDate	Number	The program start date in milliseconds.
endDate	Number	The program end date in milliseconds.
AvailabilityProperties	CustomProperty[]	The custom properties for the availability.

See Also

- GetContentList

Cast

since 2.1

This entity represents a cast person.

Properties

Name	Type	Description
castExternalId	String	The external ID of the cast person.
castName	String	The full name of the cast person (first name and last name separated by a space).
roleName	String	The role of the cast person (Actor, Director or Writer).
images	Image[]	The array of images associated with the cast person (when available).

See Also

- GetExploreOptionsList
- GetRecommendationListByPopularContributor

CastPerson

since 3.1

This entity represents a cast person.

Properties

Name	Type	Description
castId	String	The external ID of the cast person.
personName	String	The full name of the cast person (first name and last name separated by a space).

See Also

- GetVideoRecommendationList

Category

since 3.0

This entity represents a service category.

Properties

Name	Type	Description
categoryId	String	The external ID of the category.
categoryName	String	The name of the category.

Channel

since 1.0

This entity provides information about a channel.

Properties

Name	Type	Description
id	String	*(Deprecated since 1.1)* The ID of the channel in the middleware.
name	String	The name of the channel.
externalId	String	The ID of the channel in the middleware.

See Also

- GetChannelList

Content

since 2.0

This entity provides information about a content item.

Properties

Name	Type	Description
images	Image[]	The list of images.
customProperties	CustomProperty[]	The custom properties for the content.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.

See Also

- GetContent

ContentAvailability

since 2.1

This entity represents a content item and all its service and broadcast availabilities.

Properties

Name	Type	Description
id	Number	The ID of the content item.
externalContentId	String	The external ID of the content item.
name	String	The name of the content item.
images	Image[]	The array of images associated with the recommended content item.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.
profileRating	Number	The value which the logged-in user rated the content item.
prName	String	The name of the Parental Rating for the content item.
prLevel	Number	The level of the Parental Rating for the content item.
genres	Genre[]	The array of genres associated with the recommended content item.

Name	Type	Description
ContentProperties	CustomProperty[]	The custom properties for the content item.
contentType	String	The type of content (Movie, Program or Season).
externalSeasonId	String	The external ID of the season (only if the type of content is 'Program').
seasonName	String	The name of the season (only if the type of content is 'Program').
contentSeq	Number	The sequential number for the content item in the season (only if the type of content is 'Program').
externalSeriesId	String	The external ID of the series (only if the type of content is 'Season' or 'Program').
seriesName	String	The name of the series (only if the type of content is 'Season' or 'Program').
serviceAvailabilities	ServiceAvailability[]	The array of the service availabilities for this content item.
broadcastAvailabilities	BroadcastAvailability[]	The array of the broadcast availabilities for this content item.

See Also

- GetContentList

ContentRating

since 2.1

This entity represents the rating and the number of raters of the content item.

Properties

Name	Type	Description
id	String	The external ID of the content item.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.

See Also

- GetContentRatingList

CustomProperty

since 1.2

This entity provides information about custom properties.

Properties

Name	Type	Description
name	String	The name of the property.
value	String	The value of the property.

See Also

- VideoRecommendation

ExcludedContent

since 2.1

This entity represents a content item from the profile excluded items list.

Properties

Name	Type	Description
id	String	The external ID of the content item.
name	String	The name of the content item.

See Also

- GetProfileExcludeList

ExcludedData

since 2.3

This entity represents an item from the profile excluded items list.

Properties

Name	Type	Description
contents	ExcludedContent[]	The array of content items to be excluded in recommendations.
genres	ExcludedGenre[]	The array of genres to be excluded in recommendations.

See Also

- GetProfileExcludeList

ExcludedGenre

since 2.3

This entity represents a genre from the profile excluded items list.

Properties

Name	Type	Description
id	String	The external ID of the genre in the middleware.
name	String	The name of the genre.

See Also

- GetProfileExcludeList

ExploreOption

since 2.1

This entity represents an explore option of a specific meta-data type (Genre or Cast).

Properties

Name	Type	Description
genre	Genre	A genre (in case genres were requested).
cast	Cast	A cast person (in case actors or directors were requested).

Comments

- Only one of the above properties will be returned - either Genre or Cast, depending on the request.

See Also

- GetExploreOptionsList

ExtendedSocialIdentity

SocialTV *since 2.3*

This entity represents a Social Identity with middleware and profile information .

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
socialName	String	The name of the Social Identity.
firstName	String	The first name of the user.
lastName	String	The last name of the user.

Name	Type	Description
cellPhone	String	The number of the cellular phone of the user.
email	String	The e-mail address of the user.
city	String	The city of the user.
birthdate	String	The user's birthdate in format yyyy-mm-dd.
gender	String	The gender of the user.
status	String	The status of the Social Identity.
publishWishlist	Boolean	Whether the profile allows his wish list to be published to other users and social networks.
externalProfileId	String	The profile external ID.
externalHouseholdId	String	The external ID of the household in the middleware.
images	SocialImage	The images of the Social Identity.
externalIdentities	SocialExternalIdentity	The external identities of the sender.

See Also

- GetFriendList

Genre

since 1.0

This entity provides information about a genre.

Properties

Name	Type	Description
id	String	*(Deprecated since 1.1)* The ID of the genre in the middleware.
name	String	The name of the genre.
externalId	String	The ID of the genre in the middleware.

See Also

- GetDefaultGenreList
- GetGenreList
- GetExploreOptionsList

Image

since 1.0

This entity represents an image reference.

Properties

Name	Type	Description
name	String	The name of the image.
value	String	The reference to the image location.

See Also

- GetVideoRecommendationList
- GetLiveRecommendationList

Language

since 1.0

This entity represents a language defined in the system.

Properties

Name	Type	Description
code	String	The ISO two-letter code of the language, for example: English = 'en'.
name	String	The name of the language.

See Also

- GetLanguageList

LikedContent

since 3.1

This entity represents a content entity - which is part of a recommendation reason.

Properties

Name	Type	Description
contentExternalId	String	The universal ID of the content
contentName	String	The content name.
similarityPercentage	Number	The percentage of people who have watched both recommended content and the 'Liked' content.

Name	Type	Description
reasonTemplateCode	Number	A number that represents the string (template with placeholders for the data), to be shown on the client side. 10000 - Liked content 10001 - Liked content with shared genres 10010 - Liked content with shared contributors 10011 - Liked content with shared genres and shared contributors 11000 - Liked content + usage data facts 11001 - Liked content + usage data facts with shared genres 11010 - Liked content + usage data facts with shared contributors 11011 – Liked content + usage data facts with shared genres and shared contributors.
genres	Genre[]	A list of similar genres between the recommended content and the 'Liked' content.
contributors	CastPerson[]	A list of cast members that took part in both the recommended content and the 'Liked' content.

See Also

- GetVideoRecommendationList

LiveContentAvailability

since 2.4

This entity represents live content availability.

Properties

Name	Type	Description
id	Number	The ID of the content item.
externalContentId	String	The external ID of the content item.
name	String	The name for the live content item.

Name	Type	Description
images	Image[]	The array of images associated with the content item.
prName	String	The name of the Parental Rating for the content item.
prLevel	Number	The level of the Parental Rating for the content item.
availabilities	BroadcastAvailability[]	The array of objects containing "programId" and "channelId" for each of the broadcasts available for this content item.
ContentProperties	CustomProperty[]	The custom properties for the content item.
contentType	String	The type of content (currently the only possible value is "Program").
externalSeasonId	String	The external ID of the season.
seasonName	String	The name of the season.
contentSeq	Number	The sequential number of the content item in the season.
externalSeriesId	String	The external ID of the series.
seriesName	String	The name of the series.

See Also

- GetLiveSeriesAvailabilityList

LiveRecommendation

since 1.2

This entity is a recommendation object for a live event.

Properties

Name	Type	Description
id	Number	The ID of the recommendation.
externalContentId	String	The external ID of the content.
name	String	The name of the live event.
AvailabilityProperties	CustomProperty[]	The custom properties for the availability.

Name	Type	Description
images	Image[]	The array of images associated with the recommended content. If no images are associated with the content, channel images are returned (if they exist).
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.
profileRating	Number	The value which the logged-in user rated the content item.
genres	Genre[]	The array of genres associated with the recommended content item.
type	String	The engine which generated the recommendation.
programId	Number	The ID of the program recommended.
channelId	Number	The ID of the channel in which the program is being broadcast.
startDate	Number	The program start date in milliseconds.
endDate	Number	The program end date in milliseconds.
flags	Number	The Bitwise description of some program properties: • isPaymentBased (1) • isStartOver (2) • isTimeShifted (4) .
prName	String	The name of the Parental Rating for the content.
prLevel	Number	The level of Parental Rating for the content.
contentType	String	The type of content recommendation (Movie, Program or Season).
externalSeasonId	String	The external ID of the season (only if the type of recommendation is 'Program').

Name	Type	Description
seasonName	String	The name of the season (only if the type of recommendation is 'Program').
contentSeq	Number	The sequential number for the content item in the season (only if the type of recommendation is 'Program').
externalSeriesId	String	The external ID of the series (only if the type of recommendation is 'Season' or 'Program').
seriesName	String	The name of the series (only if the type of recommendation is 'Season' or 'Program').

See Also

- GetLiveRecommendationList

PopularContributorRecommendation *since 2.2*

A recommendation object for a video (on-demand content item), based on its participating popular contributors.

Properties

Name	Type	Description
id	Number	The ID of the recommendation.
externalContentId	String	The external ID of the content.
name	String	The name for the video.
images	Image[]	The array of images associated with the recommended content item.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.
externalRating	float	The external rating for the content item (returned on recommendations from the external rating engine only).
prName	String	The name of the Parental Rating for the content item.
prLevel	Number	The level of Parental Rating for the content item.

Name	Type	Description
availabilities	ServiceAvailability[]	The array of objects containing "videoId" and "serviceId" for each of the services available for this recommendation.
genres	Genre[]	The array of genres associated with the recommended content item.
ContentProperties	CustomProperty[]	The custom properties for the content item.
contentType	String	The type of content recommendation (Movie, Program or Season).
externalSeasonId	String	The external ID of the season (only if the type of recommendation is 'Program').
seasonName	String	The name of the season (only if the type of recommendation is 'Program').
contentSeq	Number	The sequential number for the content item in the season (only if the type of recommendation is 'Program').
externalSeriesId	String	The external ID of the series (only if the type of recommendation is 'Season' or 'Program').
seriesName	String	The name of the series (only if the type of recommendation is 'Season' or 'Program').
relatedPopularCasts	Cast[]	The list of popular contributors participating both in the recommended content item and in the original content item.
sourceChannelId	String	The external ID of the live channel in which the content was originally broadcasted.

See Also

- GetRecommendationListByPopularContributor

Profile

since 1.0

This entity represents a profile.

Properties

Name	Type	Description
id	Number	The profile identifier.
name	String	The profile name (unique within the household).
image	String	The path for the image attached to the profile.
suspended	Boolean	The state of the profile. A suspended profile does not collect usage data and does not receive any personalized recommendations.
externalId	String	The identifier of the profile in the external profile management system.
pinCodeExist	boolean	The mark whether the profile has a PIN code or not.
images	Image[]	The array of images associated with the profile
profileProperties	CustomProperty[]	The custom properties for the profile.
type	String	The profile type.

See Also

- GetProfileList

ProfileImage

since 2.3

This entity represents a reference to either an internal or an external image of the profile.

Properties

Name	Type	Description
id	Number	The internal ID of the image.
name	String	The name of the image.
url	String	The reference to the image location.
type	String	*(Deprecated since 2.3)* When 'image_url' was supplied - 'external'. Otherwise - 'internal'.

See Also

- CreateProfileImage
- DeleteProfileImage
- GetProfileImage
- GetProfileImageList
- UpdateProfileImage

ProfileOrder

since 1.1

This entity represents content item ordered by the profile.

Properties

Name	Type	Description
externalProfileId	String	The identifier of the profile in the external system.
externalContentId	String	The external identifier of the ordered content item.
name	String	The name of the ordered content item.
orderTime	long	The timestamp when the order took place.

See Also

- GetProfileOrderList

ProfileRating

since 1.1

This entity represents a content item rated by the profile and the rating value.

Properties

Name	Type	Description
externalProfileId	String	The identifier of the profile in the external system.
externalContentId	String	The external identifier of the rated content item.
name	String	The name of the rated content item.
rating	int	The rating the profile gave the content item.
ratingTime	long	The timestamp when the profile rated or ordered the content (the latter of the two).

See Also

- GetProfileRatingList

ProfileType

since 3.1 SR9

This entity represents a profile type.

Properties

Name	Type	Description
type	String	The profile type.

See Also

- GetProfileTypeList

Property

SocialTV *since 2.0*

This entity provides information about message custom properties.

Properties

Name	Type	Description
name	String	The name of the property.
value	String	The value of the property.

See Also

- SocialMessage

ReceivedSocialIdentity

SocialTV *since 2.0*

This entity represents the sender's Social Identity in received messages.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
socialName	String	The name of the Social Identity.
firstName	String	The first name of the user.
lastName	String	The last name of the user.
status	String	The status of the Social Identity.
images	SocialImage	The images of the Social Identity.
externalIdentities	SocialExternalIdentity	The external identities of the sender.

See Also

- ReceivedSocialMessage

ReceivedSocialMessage

***SocialTV** since 2.0*

This entity represents a Social Message in the received messages list.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
type	String	The type of the message.
time	Number	The time at which the message was sent.
source	String	The source of the message.
message	String	The free text of the message.
receiverStatus	String	The status of the message for the Social Identity that received the message.
properties	Property[]	The list of the message properties.
SocialIdentity	ReceivedSocialIdentity	The social Identity of the sender.

See Also

- GetReceivedSocialMessages

RecommendationReason

since 3.1

This entity represents a reason to support the recommendation.

Properties

Name	Type	Description
type	String	The type of recommendation reason. Contains "engine" or "likedContent".
engineType	String	The name of the engine generatinged this recommendation. Will be displayed in case of a reason with the type "engine".
engineLabel	String	A label that indicates the type of information that is provided on the 'engineInfo' parameter.

Name	Type	Description
engineInfo	String	Additional information. Varies according to the type of engine. For the "promotion" engine - the promotion label. For the "external rating" engine - the name of the reviewer. For the "most popular" engine - the popularity period.
reasonTemplateCode	Number	A number that represents the string (template with placeholders for the data), to be shown on the client side. 1000 - Most popular 2000 - External rating 3000 - Promotions 4000 - Personal 5000 - Personal + liked content aggregator
likedContents	LikedContent[]	A list of contents that the profile liked, and then supported the recommendation. The list is displayed in case of a personal recommendation, and only when there is a significant link between the liked content and the recommended content.

See Also

- GetVideoRecommendationList

Result

since 1.0

Response object for Authentication and Data modification actions.

Properties

Name	Type	Description
status	String	The result status,which is usually SUCCESS.
message	String	An additional informative message.
properties	Map/Array	Any properties that the action returns. For example, if creating a profile, this contains the ID of the new profile.

SearchSocialIdentity

***SocialTV** since 2.0*

This entity represents the Social Identity that is returned from a search operation.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
socialName	String	The name of the Social Identity.
firstName	String	The first name of the user.
lastName	String	The last name of the user.
images	SocialImage[]	The images of the Social Identity.
externalProfileId	String	The profile external ID.
birthdate	Number	The user's birthdate in format yyyy-mm-dd.
publishWishlist	Boolean	Whether the profile allows his wish list to be published to other users and social networks.
externalIdentities	SocialExternalIdentity[]	The external identities of the sender.

See Also

- SearchSocialIdByExtId

Season

since 2.3 SR2

This entity represents a season content.

Properties

Name	Type	Description
id	String	The external ID of the season.
name	String	The name of the season.
contentSeq	Number	The sequential number for the season.
images	Image[]	The array of images associated with the season.
ContentProperties	CustomProperty[]	The custom properties for the season.

See Also

- GetSeriesSeasonList

SentSocialIdentity

SocialTV since 2.0

This entity represents the receiver's Social Identity in sent messages.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
socialName	String	The name of the Social Identity.
firstName	String	The first name of the user.
lastName	String	The last name of the user.
status	String	The status of the Social Identity.
images	SocialImage	The images of the Social Identity.
externalIdentities	SocialExternalIdentity	The external identities of the receiver.

See Also

- SentSocialMessage

SentSocialMessage

SocialTV since 2.0

The entity represents a Social Message in the sent messages list.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
type	String	The type of the message.
time	Number	The time at which the message was sent.
source	String	The source of the message.
message	String	The free text of the message.
senderStatus	String	The status of the message for the Social Identity that sent the message.
properties	Property[]	The list of message properties.
SocialIdentity	SentSocialIdentity	The social Identity of the sender.

See Also

- GetSentSocialMessages

ServiceAvailability

since 1.2

Video may be purchased in more than one way in the middleware (VOD or SVOD for example). This entity represents the manner in which video content can be purchased.

Properties

Name	Type	Description
videoId	String	The ID of the video in the middleware.
serviceId	String	The ID of the service in which it can be purchased.
startTime	Number	The start time from which the content can be consumed.
AvailabilityProperties	CustomProperty[]	The custom properties for the availability.
categories	Category[]	The categories of the availability.
images	Image[]	The images of the availability.

See Also

- GetVideoRecommendationList

SocialExternalIdentity

SocialTV *since 2.0*

This entity represents an External Identity of the Social Identity, associated with the external social network.

Properties

Name	Type	Description
externalId	String	The ID in the external network.
networkType	String	The type of network to which the External Identity is associated.

See Also

- CreateExternalIdentity
- GetExternalIdentities
- GetSocialIdentity
- CheckSocialNameAvailability
- CreateSocialImage
- DeleteExternalIdentity

SocialIdentity

SocialTV *since 2.0*

This entity represents a Social Identity.

Properties

Name	Type	Description
id	Number	The ID of the Social Identity.
socialName	String	The name of the Social Identity.
firstName	String	The first name of the user.
lastName	String	The last name of the user.
cellPhone	String	The number of the cellular phone of the user.
email	String	The e-mail address of the user.
city	String	The city of the user.
birthdate	String	The user's birthdate in format yyyy-mm-dd.
gender	String	The gender of the user.
status	String	The status of the Social Identity.
publishWishlist	Boolean	Whether the profile allows his wish list to be published to other users and social networks.
images	SocialImage	The images of the Social Identity.
externalIdentities	SocialExternalIdentity	The external identities of the sender.

See Also

- GetSocialIdentity

SocialImage

SocialTV since 2.0

This entity represents a reference to either an internal or an external image.

Properties

Name	Type	Description
id	Number	The internal ID of the image.
name	String	The name of the image.
url	String	The reference to the image location.
type	String	*(Deprecated since 2.3)* When 'image_url' was supplied - 'external'. Otherwise - 'internal'.

See Also

- CreateSocialImage
- GetSocialImage
- DeleteSocialImage

VideoContentAvailability

since 2.0

This entity represents a video content availability.

Properties

Name	Type	Description
id	Number	The ID of the content item.
externalContentId	String	The external ID of the content item.
name	String	The name for the video content item.
images	Image[]	The array of images associated with the recommended content item.
prName	String	The name of the Parental Rating for the content item.
prLevel	Number	The level of the Parental Rating for the content item.
availabilities	ServiceAvailability[]	The array of objects containing "videoId" and "serviceId" for each of the services available for this content item.
ContentProperties	CustomProperty[]	The custom properties for the content item.
contentType	String	The type of content (in 2.0 the only possible value is "Program").
externalSeasonId	String	The external ID of the season.
seasonName	String	The name of the season.
contentSeq	Number	The sequential number for the content items in the season.
externalSeriesId	String	The external ID of the series.
seriesName	String	The name of the series.

See Also

- GetVideoSeasonContentList

VideoRecommendation *since 1.2*

This entity provides a recommendation object for a video (on-demand content item).

Properties

Name	Type	Description
id	Number	The ID of the recommendation.
externalContentId	String	The external ID of the content. In case contentType is 'season', this is the external ID of the season.
name	String	The name for the video. In case contentType is 'season', this is the name of the season.
images	Image[]	The array of images associated with the recommended content item.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.
profileRating	Number	The value which the logged-in user rated the content item.
externalRating	float	The external rating for the content item (returned on recommendations from the external rating engine only).
type	String	The engine which generated the recommendation.
prName	String	The name of the Parental Rating for the content item.
prLevel	Number	The level of Parental Rating for the content item.
availabilities	ServiceAvailability[]	The array of objects containing "videoId" and "serviceId" for each of the services available for this recommendation.
genres	Genre[]	The array of genres associated with the recommended content item.
ContentProperties	CustomProperty[]	The custom properties for the content item.

Name	Type	Description
contentType	String	The type of content recommendation (Movie, Program or Season).
externalSeasonId	String	The external ID of the season (only if the type of recommendation is 'Program').
seasonName	String	The name of the season (only if the type of recommendation is 'Program').
contentSeq	Number	The sequential number for the content item in the season (only if the type of recommendation is 'Program').
externalSeriesId	String	The external ID of the series (only if the type of recommendation is 'Season' or 'Program').
seriesName	String	The name of the series (only if the type of recommendation is 'Season' or 'Program').
seriesRating	Number	The rating of the series (only if the type of recommendation is 'Season' or 'Program').
seriesRatersCount	Number	The number of the series raters (only if the type of recommendation is 'Season' or 'Program').
sourceChannelId	String	The external ID of the live channel in which the content was originally broadcasted.
broadcastTime	long	The timestamp in which the content was originally broadcasted.
recommendationReasons	RecommendationReason	Supporting information for the content recommendation.

See Also

- GetVideoRecommendationList

ViewedContent

since 1.0

This entity describes an on demand content item which the profile already viewed.

Properties

Name	Type	Description
id	Number	The ID of the content item.

See Also

- GetViewedContentList

WishListRecommendation

since 2.2

This entity represents an item in the profile's wish list.

Properties

Name	Type	Description
id	Number	The ID of the recommendation.
externalContentId	String	The external ID of the content.
name	String	The name for the video.
images	Image[]	The array of images associated with the recommended content item.
seriesImages	Image[]	The array of images associated with the recommended series. Since 3.1 SR7.
rating	float	The average user rating for the content item.
ratersCount	Number	The number of raters for the user rating.
externalRating	float	The external rating for the content item (returned on recommendations from the external rating engine only).
recommended	Boolean	This is true in the case when the item was added to a wish list based on a COMPASS recommendation, false otherwise.
engine	String	In the case when the item was added to a wish list, based on a COMPASS recommendation, The name of the engine which generated the recommendation.
blend	String	In the case when the item was added to a wish list, based on a COMPASS recommendation, The name of the blend that led to the recommendation.
prName	String	The name of the Parental Rating for the content item.

Name	Type	Description
prLevel	Number	The level of the Parental Rating for the content item.
availabilities	ServiceAvailability[]	The array of objects containing "videoId" and "serviceId" for each of the services available for this recommendation.
genres	Genre[]	The array of genres associated with the recommended content item.
ContentProperties	CustomProperty[]	The custom properties for the content item.
contentType	String	The type of content recommendation (Movie, Program or Season).
externalSeasonId	String	The external ID of the season (only if the type of recommendation is 'Program').
seasonName	String	The name of the season (only if the type of recommendation is 'Program').
contentSeq	Number	The sequential number for the content item in the season (only if the type of recommendation is 'Program').
externalSeriesId	String	The external ID of the series (only if the type of recommendation is 'Season' or 'Program').
seriesName	String	The name of the series (only if the type of recommendation is 'Season' or 'Program').
addToListTime	long	The timestamp when the content item was added to the profile wish list.
sourceChannelId	String	The external ID of the live channel in which the content was originally broadcasted.

See Also

- GetProfileWishList

WishListUsageContent

since 2.1

This entity represents a content item from the profile wish list.

Properties

Name	Type	Description
id	String	The external ID of the content item.

See Also

- GetProfileWishlistUsageData

Actions

This chapter describes the actions exposed by the runtime API.

AcceptInvitation

SocialTV ***since 2.0***

This action accepts an invitation sent by another Social Identity. A message of type 'AcceptInvitation' is issued to the inviter.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the social identity that sent the invitation.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Comments

- The status of the relationship between the parties is changed to 'Active'.

Examples

```
AcceptInvitation?related_social_id=1
```

See Also

- BlockSocialIdentity
- DeleteFriend
- IgnoreInvitation
- InviteFriend
- UnblockSocialIdentity

ActivateSocialIdentity

SocialTV ***since 2.0***

This action updates the status of the logged-in Social Identity to 'Active'.

Returns

Name	Description
Result	

Examples

```
ActivateSocialIdentity
```

See Also

- DeactivateSocialIdentity

AddContentToProfileExcludeList

since 2.1

This action adds the specified content to the content exclusion list of the logged-in profile.

Parameters

Name	Type	Description
content_id	String	The external ID of the on-demand content item that should be added to the content exclusion list.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_CONTENT_ID	In the case when content with the given external content id does not exist.

Examples

```
AddContentToProfileExcludeList?content_id=id1
```

AddProfile

since 1.0

This action adds a new profile to the logged-in household.

Parameters

Name	Type	Description
name	String	The name of the new profile. The name must be unique within the household.
image	String	The location of the profile's image (avatar).
suspended	Boolean	Whether the profile is suspended or not. The usage data is not collected for a suspended profile. Defaults to 'false'.

Name	Type	Description
auto	Boolean	Whether profile creation is triggered automatically by the client or explicitly by the active profile. Defaults to 'false'.
external_id	String	The identifier of the new profile in the external system. Defaults to a string representation of the newly created profile ID.
pin_code	String	The PIN code chosen by the user in order to authenticate the profile.
property	String	The property value is key-value pair that represents the profile custom properties (key and value).
type	String	The type of the new profile. Should be one of the profile types given by the GetProfileTypeList API.

Returns

Name	Description
Result	The result includes the ID of the new profile that was created. See the example below.

Throws

Name	Description
DUPLICATE_KEY	In case two or more keys are equals.
INVALID_PROFILE_ID	In the case when profile with the given external_id already exists.
MAX_PROFILES_EXCEEDED	In case the configured profile limit number per household has exceeded (since 3.1 SR9).
NOT_IN_CUSTOM_PROPERTY	In case the requested key not exist in custom property.
PROFILE_EXISTS	In the case when the name of the profile is the same as the one that already exists in the same household.

Comments

- The profile ID is entity returned within the result: `<Result Status="SUCCESS" Message="Add profile operation finished successfully"><Properties>`**`<Property name="profileId" value="1234"/>`**`</Properties></Result>`

Examples

```
AddProfile?name=John Smith&image=/avatars/Flower.jpg&suspended=false&auto=false
```

```
AddProfile?name=david&property=key1:val1&property=key2:va
```

See Also

- GetProfile
- UpdateProfile
- DeleteProfile

AddToProfileExcludeList

since 2.3

This action adds the specified entity (content or genre) to the exclusion list of the logged-in profile.

Parameters

Name	Type	Description
id	String	The external ID of the genre/content item to be added to the exclusion list.
type	String	The type of the excluded item. Possible values: genre, content.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_CONTENT_ID	In the case when content with the given external content id does not exist.
INVALID_GENRE_ID	In the case when invalid genre ID is given.
INVALID_PARAMETER_VALUE	In the case when the type parameter value is different than the valid values (genre, content).

Examples

```
AddToProfileExcludeList?type=content&id=1234
```

```
AddToProfileExcludeList?type=genre&id=1111
```

See Also

- RemoveFromProfileExcludeList
- ClearProfileExcludeList
- GetProfileExcludeList

AddToWishList *since 2.2*

Adds an item to the profile wish list. Specify the ID of a VOD movie, episode, season or series to add to the wish list.

Parameters

Name	**Type**	**Description**
external_content_id	String	The external ID of the on-demand content item.
recommended	Boolean	This is true if the content item was recommended by COMPASS. False if the event took place elsewhere in the service, for example the VOD catalog. A "true" value is referred to as a "conversion".
engine	String	In the case when a specific engine recommended the content, this field should contain the name of the engine. Valid values are: "most_popular", "user_rating", "user_rating_series", "external_rating", "promotions", "cf", "related_content", "most_recent", "most_popular_series" and "viewer_series".
blend	String	In the case when COMPASS recommended the content item, this field should contain the name of the blend that led to the recommendation. Blend name is case insensitive. If not specified, the default blend is assumed.

Throws

Name	**Description**
INVALID_CONTENT_ID	In the case when content with the given external_content_id does not exist.
INVALID_PARAMETER_VALUE	In the case when the engine name is invalid

Comments

- In the Configuration, set whether to add a representative season of a series to a wish list or to add the specified season. Contact VO Professional Services for information on changing the Configuration.
    - To add a representative season of a series to the wish list, the parameter wishlist-series-instead-of-season in the Configuration must be set to true and in external_content_id specify the external series/season ID. A representative season of the series is added to the wishlist.
    - To add the specified season to the wish list, the parameter wishlist-series-instead-of-season must be set in the Configuration to false (default) and in external_content_id specify the external season ID. In this case a series cannot be added.

Examples

```
AddToWishList?external_content_id=1000&recommended=false
```

```
AddToWishList?external_content_id=1001&recommended=true&engine=cf
```

See Also

- GetProfileWishList
- RemoveFromWishList

BlockSocialIdentity

SocialTV since 2.0

This action ignores an invitation from another Social Identity and blocks it from sending invitations in the future.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the Social Identity to block.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Comments

- The relationship status with the related Social Identity is changed to 'Block Ignore'.

Examples

```
BlockSocialIdentity?related_social_id=1
```

See Also

- AcceptInvitation
- IgnoreInvitation
- InviteFriend
- UnblockSocialIdentity

CheckSocialNameAvailability

SocialTV since 2.0

This action checks if the given social name is occupied. The name availability should be checked before creating a Social Identity.

Parameters

Name	Type	Description
social_name	String	The name to be validated for availability.

Returns

Name	Description
Result	The result indicates availability. If the name is occupied, it returns available=false. See comment below.

Comments

- The availability entity is returned within the result, true if not occupied:

  ```
  <Result Status="SUCCESS" Message="Identity name availability">

  <Properties><Property name="available" value="true"/></Properties>

  </Result>
  ```

Examples

```
CheckSocialNameAvailability?social_name=WilliamSmith
```

See Also

- CreateSocialIdentity

ClearProfileContentExcludeList

since 2.1

This action clears the content exclusion list of the logged-in profile.

Returns

Name	Description
Result	

Examples

```
ClearProfileContentExcludeList
```

ClearProfileExcludeList

since 2.3

This action clears the exclusion list of the logged-in profile.

Parameters

Name	Type	Description
type	String	The type of the excluded item to be cleared. Possible values: all, genre, content.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter value is different than the valid values (all, genre, content).

Examples

```
ClearProfileExcludeList?type=genre
```

See Also

- AddToProfileExcludeList
- RemoveFromProfileExcludeList
- GetProfileExcludeList

CreateExternalIdentity

SocialTV *since 2.0*

This action creates an External Identity for the logged-in Social Identity. This operation associates the Social Identity with the user's profile in an external social network (e.g., Facebook), and saves the users' ID in that network.

Parameters

Name	Type	Description
social_name	String	The name of the Social Identity to associate.
network_type	String	The type of network to which the External Identity is being associated. Values: 'facebook'
external_id	String	The user's ID in the external network.
password	String	The password that is defined in the Social Identity.

Name	Type	Description
property	String	A single pair of property name and value in the format name:value. More than one 'property' parameter can be set.

Returns

Name	Description
Result	The result includes the ID of the Social Identity of the logged-in Social Identity, and the profile ID of the logged in profile. See comment below.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the given network type is not a valid value.
INVALID_SOCIAL_IDENTITY	In the case when the given social name does not exist.

Comments

- This action does not require authentication.
- The social ID and Profile ID entity are returned within the result:

  ```
  <Result Status="SUCCESS" Message="Create Social External Identity
  operation finished successfully">

  <Properties><Property name="socialId" value="1234"/><Property
  name="profileId" value="5678"/></Properties>

  </Result>
  ```
- The password is determined while creating the Social Identity or updating it.

Examples

```
CreateExternalIdentity?        network_type=facebook&external_id=12345&
        password=1111&        social_name=WilliamSmith&
 property=access_token:vlkkXbV2aDX6Hcbl5bBULmg
```

See Also

- CreateSocialIdentity
- DeleteExternalIdentity
- GetExternalIdentities
- UpdateSocialIdentity

CreateHousehold

since 2.4

This action creates a new household in COMPASS and subscribes it to the given services.

Parameters

Name	Type	Description
external_id	String	The external ID of the household in the middleware.
signature	String	The calculated URL signature. Used instead of authentication when calling the API anonymously. See the section: **Calculating URL signature**.
services	String	Comma delimited list of service external IDs or service names for which the household is subscribed. When this parameter is not used, the household will be subscribed to all services defined as "auto subscribe" in the Administration GUI. The external IDs or service names must match the external IDs or service names that were defined in the Administration GUI. The following format should be used: service-name1, service-name2, etc.
packages	String	Comma delimited list of packages IDs for which the household is subscribed.

Returns

Name	Description
Result	The result includes the ID of the new household. See the example below.

Throws

Name	Description
HOUSEHOLD_ALREADY_EXISTS	In the case when the household external_id already exists in COMPASS.
INVALID_PARAMETER_VALUE	In the case when the signature is not correct or one of the service names/external_ids does not exist.

Comments

- The household ID is an entity returned within the result: `<Result Status="SUCCESS" Message="Create household operation finished successfully"> <Properties>`**`<Property name="householdId" value="1234"/>`**`</Properties></Result>`

Examples

```
CreateHousehold?external_id=householdID2&signature=RXlaakpaVH...
```

CreateProfileImage

since 2.3

This action creates an image for the logged-in profile.

Parameters

Name	Type	Description
image_name	String	The name of the image, used to differentiate between the different images of the user.
image_description	String	Free text that describes the image.
image_url	String	The URL where the image is located, in case the image is not stored by the system.

Returns

Name	Description
ProfileImage	

Throws

Name	Description
CONTENT_TYPE_MISSING	The content type is not supplied in the request for binary images.
IMAGE_ALREADY_EXIST	The image with the supplied image_name already exists for the logged-in profile.
INVALID_IMAGE_SIZE	The image size exceeds the maximum system configuration limit.

Comments

- The image name must be unique under the same profile.
- If the picture is an external avatar, image_url should be given. For a picture that is uploaded to be stored in the database, the image binaries should be provided (via HTTP Post).
- For binary images, image type is a mandatory attribute of the HTTP header ("content-type"). The type should be a valid image type defined by the HTTP protocol: for example, "image/jpeg".

Examples

```
CreateProfileImage?image_name=WilliamSmith&image_url=/images/WilliamSmith1.jpg
```

See Also

- DeleteProfileImage
- GetProfileImage
- GetProfileImageList
- UpdateProfileImage

CreateSocialIdentity

***SocialTV** since 2.0*

This action creates a Social Identity for the logged-in profile, if one does not already exist.

Parameters

Name	Type	Description
social_name	String	The unique name for the Social Identity.
first_name	String	The first name of the user.
last_name	String	The last name of the user.
password	String	The password that is used later when creating external identities.
cell_phone	String	The user's phone number.
email	String	The user's e-mail address.
city	String	The user's city.
gender	String	The user's gender. Values: 'male', 'female'.
birthday	Number	The user's birthday in milliseconds. There are no validations on the value except for field type.
publish_wishlist	Boolean	Whether the profile allows his wish list to be published to other users and social networks. Default value: false.

Returns

Name	Description
Result	The result includes the ID of the new Social Identity that was created. See comment below.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the gender parameter contains an invalid value.

Name	Description
SOCIAL_EXTERNAL_IDENTITY_EXIST	If a Social Identity already exists for the profile or the Social Name is already occupied.

Comments

- The social ID entity is returned within the result:

  ```
  <Result Status="SUCCESS" Message="Create Social Identity operation
  finished successfully">

  <Properties><Property name="socialId" value="1234"/></Properties>

  </Result>
  ```
- Before creating a Social Identity, the client should check if the social name is occupied by invoking `CheckSocialNameAvailability`.
- The password is needed when the Social Identity is associated with the user's profile in other social networks, such as Facebook (i.e., invoking `CreateExternalIdentity`). The user must submit this password in order to associate the Social Identity with the external profile.

Examples

```
CreateSocialIdentity?
social_name=WilliamSmith&first_name=William&last_name=Smith&password=yJ82l
```

See Also

- CheckSocialNameAvailability
- CreateExternalIdentity
- CreateSocialImage
- DeleteExternalIdentity
- GetSocialIdentity
- UpdateSocialIdentity

CreateSocialImage

SocialTV since 2.0

This action creates an image for the logged-in Social Identity.

Parameters

Name	Type	Description
image_name	String	The name of the image, used to differentiate between the different images of the user.
image_description	String	Free text that describes the image.
image_url	String	The URL where the image is located, in case it is not stored by the system.

Returns

Name	Description
SocialImage	

Throws

Name	Description
CONTENT_TYPE_MISSING	The content type is not supplied in the request for binary images.
IMAGE_ALREADY_EXIST	The image with the supplied image_name already exists for the logged-in Social Identity.
INVALID_IMAGE_SIZE	The image size exceeds the maximum system configuration limit.

Comments

- The image name must be unique under the same Social Identity.
- If the picture is an external avatar, image_url should be given. For a picture that is uploaded to be stored in the database, the image binaries should be provided (via HTTP Post).
- For binary images, image type is a mandatory attribute of the HTTP header ("content-type"). The type should be a valid image type defined by the HTTP protocol: for example, "image/jpeg".

Examples

```
CreateSocialImage?image_name=WilliamSmith&image_url=/images/WilliamSmith1.jpg
```

See Also

- DeleteSocialImage
- GetSocialImage
- UpdateSocialImage

DeactivateSocialIdentity

***SocialTV* since 2.0**

This action updates the status of the logged-in Social Identity to 'Inactive'.

Returns

Name	Description
Result	

Examples

```
DeactivateSocialIdentity
```

See Also

- ActivateSocialIdentity

DeleteExternalIdentity

***SocialTV** since 2.0*

This action deletes an External Identity for the logged-in Social Identity.

Parameters

Name	Type	Description
network_type	String	The type of network with which the External Identity is associated. Values: 'facebook'.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_EXTERNAL_IDENTITY	In the case when trying to delete an External Identity that does not exist for the logged-in Social Identity.
INVALID_PARAMETER_VALUE	In the case when the given network type is not a valid value.

Examples

```
DeleteExternalIdentity?network_type=facebook
```

See Also

- CreateExternalIdentity
- GetExternalIdentities

DeleteFriend

***SocialTV** since 2.0*

This action deletes a Social Identity from the friends list.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the Social Identity to delete.

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Examples

```
DeleteFriend?related_social_id=1
```

See Also

- AcceptInvitation
- BlockSocialIdentity
- InviteFriend
- IgnoreInvitation
- UnblockSocialIdentity

DeleteProfile

since 1.0

This action deletes a profile from the system. Also deletes the social identity that is associated with the profile.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PROFILE_ID	In the case when the given profile does not exist or does not belong to the logged-in household.

Comments

- All profile data is removed, except for historic data used in order to build the collaborative filtering similarity model and statistical data used for administrative reports.

Examples

```
DeleteProfile
```

See Also

- GetProfile
- AddProfile
- UpdateProfile
- DeleteSocialIdentity

DeleteProfileHistory

since 3.0

This action removes the logged-in profile implicit data (genre and channel lists) without changes the profile status to Suspended so that usage data will be still collected for it. In addition, the profile order history is deleted and the profile rating history is made anonymous by connecting it to a new dummy profile.

Returns

Name	Description
Result	

Examples

```
DeleteProfileHistory
```

See Also

- GetProfileImplicitChannelList
- GetProfileImplicitGenreList
- GetProfileRatingList

DeleteProfileImage

since 2.3

This action deletes a profile image from the system.

Parameters

Name	Type	Description
image_id	Number	The internal ID of the image.
image_name	String	The name of the image.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PROFILE_IMAGE	There is no image with the given ID.

Comments

- One of the parameters, 'image_id' or 'image_name', is mandatory. If both of the parameters will be filled only image_id will be considered.

Examples

```
DeleteProfileImage?image_id=1
```

```
DeleteProfileImage?image_name=small
```

See Also

- CreateProfileImage
- GetProfileImage
- GetProfileImageList
- UpdateProfileImage

DeleteSocialIdentity

SocialTV ***since 2.0***

This action deletes the Social Identity that is associated with the logged-in profile.

Returns

Name	Description
Result	

Comments

- All the External Identities that are related to the Social Identity are also removed.

Examples

```
DeleteSocialIdentity
```

See Also

- CreateSocialIdentity
- GetSocialIdentity

DeleteSocialImage

***SocialTV* since 2.0**

This action deletes an image from the system.

Parameters

Name	Type	Description
image_id	Number	The internal ID of the image.
image_name	String	The name of the image.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IMAGE	There is no image with the given ID.

Comments

- One of the parameters 'image_id' or 'image_name' is mandatory.

Examples

```
DeleteSocialImage?image_id=1
```

```
DeleteSocialImage?image_name=small
```

GetChannelList

since 1.0

This action returns all the channels defined in the system

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Channel[]	

Examples

```
GetChannelList
```

GetContent (deprecated) *since 2.0*

since 2.1 (instead use GetContentList)

This returns content information.

Parameters

Name	Type	Description
content_id	String	The external ID of the on-demand content item.

Returns

Name	Description
Content	

Examples

```
GetContent?content_id=id1
```

GetContentList *since 2.1*

This action receives a list of contents IDs and returns a list of contents with all availabilities (both video and live).

Parameters

Name	Type	Description
content_id_list	String	A comma-delimited list of content external IDs.
subscription	Boolean	If set to "true", filter by the services and channels to which the household is subscribed. If set to "false", the results are returned from all the services and channels. Default value: "false".
live_catchup_only	Boolean	If set to "true", only live Catch-Up content is returned. The default value is false.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ContentAvailability[]	

Comments

- In the case when a content id in the list does not exist in COMPASS, no error will be returned and the returned list will not contain this content information.

Examples

```
GetContentList?content_id_list=id1,id2,id3
```

GetContentRatingList

since 2.1

This receives a list of content IDs and returns their ratings.

Parameters

Name	Type	Description
content_id_list	String	The comma-delimited list of content external IDs.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ContentRating[]	

Comments

- In the case when a content id in the list does not exist in COMPASS, no error will be returned and the returned list will not contain this content rating.

Examples

```
GetContentRatingList?content_id_list=id1,id2,id3
```

GetDefaultGenreList

since 1.0

In this action the operator can define a subset of genres in the system as default genres. These are used in order to display a reduced list from which the profile can select explicit preferences. This API returns all default genres that are defined.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Genre[]	

Examples

```
GetDefaultGenreList
```

GetExploreOptionsList

since 2.1

This action returns a list of explore options taken from the specified content meta-data, according to the specified type.

Parameters

Name	Type	Description
type	String	The desired exploration option. Valid values are: "genre", "actor", "director".
external_content_id	String	The external ID of the on-demand content item.
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ExploreOption[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter value is different than the valid values, or there is no content with the given external_content_id.

Comments

- The items returned from this API represent options that can be used by GetExploredContentList to get more contents containing this option meta-data.

Examples

```
GetExploreOptionsList?type=genre&external_content_id=1234
```

```
GetExploreOptionsList?type=actor&external_content_id=5678&max_results=20
```

See Also

- GetExploredContentList

GetExploredContentList *since 2.1*

This action returns a list of video recommendations, according to a given meta-data entity ID.

Parameters

Name	Type	Description
entity_type	String	The meta-data type. Valid values are: "genre", "actor", "director" and "cast". If set to "cast", it filters the roles by actor or director.
external_entity_id	String	The external ID of the meta-data entity (as returned in the ExploreOption entity from the GetExploreOptionsList API).
encoding	String	Comma-delimited list of encoding formats or external IDs, supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The format's values must match those recognized by the system, for example: MPEG-2,MPEG-4.
max_pr_level	Number	The maximum level of the parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", it filters by the services to which the household is subscribed. If set to "false", the results are returned to all the services in the middleware. Default value: "true".
filter_viewed_content	Boolean	This defines whether to filter viewed content when returning recommendations. This value overrides the global default value which can be defined via the COMPASS administration GUI.
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value.

Name	Type	Description
services	String	Comma-delimited list of service names or external IDs, according to which the results are filtered. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: service-name1,service-name2, etc
source_channel_id	String	Comma-delimited list of external channel IDs, according to which the engines filter the results.
quality	String	Comma-delimited list of qualities, according to which the results are filtered. For example: HD,SD,3D.
delivery_mode	String	Comma-delimited list of delivery modes, according to which the results are filtered. The following format should be used: download, streaming.
device_models	String	Comma-delimited list of device model external IDs, according to which the engines filter the results (content that supports at least one of the device models in the list will be returned). This parameter overrides the 'delivery_mode' and 'encoding' parameters (which will be ignored).
from	Number	The first index requested.
count	Number	The number of elements requested.
content_type	String	Comma-delimited list of content types, according to which the engines filter the results. If specified, it returns a list of recommendations from a specified content types. Valid values are: "movie" or "season".

Returns

Name	Description
VideoRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the entity_type parameter value is different than the valid values or there is no entity with the given external_entity_id.

Comments

- In the case when the 'device_models' parameter is passed to the action, the 'delivery_mode' and 'encoding' parameters will be ignored, and content items that are not supported by the device model will not be returned.
- Season content will be considered since it supports all the device models that its episodes support.

Examples

```
GetExploredContentList?entity_type=genre&external_entity_id=1
```

```
GetExploredContentList?
entity_type=actor&external_entity_id=5&subscription=false&max_results=20
```

```
GetExploredContentList?
entity_type=genre&external_entity_id=1&source_channel_id=id1,id2
```

```
GetExploredContentList?
entity_type=genre&external_entity_id=1&content_type=movie,program
```

See Also

- GetExploreOptionsList

GetExploredLiveContentList

since 2.1

This action returns a list of live content recommendations, according to a given meta-data entity ID.

Parameters

Name	Type	Description
entity_type	String	The meta-data type. Valid values are: "genre", "actor", "director" and "cast". If set to "cast", the API filters the roles by actor or director.
external_entity_id	String	The external ID of the meta-data entity (as returned in the ExploreOption entity from the GetExploreOptionsList API).
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value.The maximum results limited to 500

Name	Type	Description
channels	String	Comma-delimited list of external channel IDs, according to which the engines filter the results.
subscription	Boolean	If set to "true", the API filters according to the services to which the household is subscribed. If set to "false", the results are returned to all the services in the middleware. Default value: "true".
max_pr_level	Number	The maximum level of the parental rating set in the returned recommended content items.
device_models	String	Comma-delimited list of device model external IDs, according to which the engines filter the results (content that supports at least one of the device models in the list is returned). This parameter overrides the 'delivery_mode' and 'encoding' parameters (which are ignored).
genres	String	Comma-delimited list of genre external IDs, according to which the engines filter the results
start_time	Number	The start time minimum value in milliseconds of returned programs.
end_time	Number	The start time maximum value in milliseconds of returned programs. In the case when the start_time parameter is given, and it should be later than the start_time parameter.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
LiveRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	Used in the case when the entity_type parameter value is different than the valid values or there is no entity with the given external_entity_id.
MISSING_PARAMETER	Used in the case when one of the mandatory parameters is missing .

Examples

```
GetExploredLiveContentList?entity_type=genre&external_entity_id=1
```

```
GetExploredLiveContentList?
entity_type=actor&external_entity_id=5&subscription=false&max_results=20
```

```
GetExploredLiveContentList?
entity_type=genre&external_entity_id=1&channels=id1,id2
```

```
GetExploredLiveContentList?
entity_type=genre&external_entity_id=1&start_time=1292516554000&end_time=1292523754000
```

GetExternalIdentities

***SocialTV** since 2.0*

This action gets a list of External Identities that are associated with the logged-in Social Identity.

Parameters

Name	Type	Description
network_type	String	The type of network with which the External Identity is associated. Values: 'facebook'. If not specified, all external identities are returned.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
SocialExternalIdentity[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the given network type is not a valid value.

Examples

```
GetExternalIdentities?network_type=facebook
```

GetFriendList

***SocialTV* since 2.0**

This action returns a list of Social Identities that have a relationship with the logged-in Social Identity.

Parameters

Name	Type	Description
search_text	String	The prefix of text to be searched in Social Name, First Name and Last Name, among friends.
relation_status_list	String	The comma-delimited list of statuses of relationships between the logged-in Social Identity and related identities. Values: 'ACTIVE', 'BLOCKIGNORE', 'PENDING', 'INVITE'. Default: 'ACTIVE'.
image_name	String	The name of the image to include in the results for all the returned Social Identities. If not supplied, it does not return images in the results. If the image name does not exist for the Social Identity, no images are returned in the results.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ExtendedSocialIdentity[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the given relation status list contains an invalid value.

Comments

- The results are sorted by first name and then by the last name ascending.

Examples

```
GetFriendList?search_text=M&from=0&count=30
```

```
GetFriendList?relation_status_list=active,blockignore
```

```
GetFriendList?image_name=profile
```

GetFrontEndAppsConfig

since 3.1 SR9

This action gets the front end applications central configuration as defined in COMPASS configuration. It is recommended to use the configuration in the JSON format.

Returns

Name	Description
Result	The result includes the front end applications configuration or null if nothing was configured.

Comments

- The configuration is returned within the result: `<Result Status="SUCCESS" Message="{the configuration}"/>`

Examples

```
GetFrontEndAppsConfig
```

GetGenreList

since 1.0

This action returns all the genres defined in the system

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Genre[]	

Examples

```
GetGenreList
```

GetLanguageList

since 1.0

This action returns a list of the languages defined in the system.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Language[]	

Examples

```
GetLanguageList
```

GetLiveRecommendationList

since 1.0

This action returns a list of live recommendations both for the logged-in profile and for anonymous users.

Parameters

Name	Type	Description
type	String	If specified, returns a list of recommendations from a specific engine. Valid values are: "all", "user_preferences", "promotions", "viewer_series" and "related_content". Defaults to "all". For a detailed description of each engine, please refer to the COMPASS Overview guide.
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value which can be defined via the COMPASS Administration GUI.
encoding	String	Comma-delimited list of encoding formats or external IDs supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The format values must match those recognized by the system, for example: MPEG-2,MPEG-4.
max_pr_level	Number	The maximum level of the parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", filter by the channels to which the household is subscribed. If set to "false", the results are returned from all the channels in the middleware. Default value: "true".

Name	Type	Description
channels	String	Comma-delimited list of channel external IDs, according to which the engines filter the results. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: channel-extId1, channel-extId2, etc.
genres	String	Comma-delimited list of genre external IDs, according to which the engines filter the results (contents containing at least one of the genres in the list will be returned).
from	Number	The first index requested.
count	Number	The number of elements requested.
blend	String	Name of the blend according to which the recommendation results from various recommendation engines are blended together. This parameter is only relevant when type parameter is set to "all". Blend name is case-insensitive. If not specified, the default blend is used.
start_time	Number	The start time in milliseconds for a future live recommendations window.
end_time	Number	The end time in milliseconds for a future live recommendations window. This parameter is mandatory In the case when the start_time parameter is given, and it should be later than the start_time parameter.
device_models	String	Comma-delimited list of device model external IDs, according to which the engines filter the results (contents that support at least one of the device models in the list will be returned). This parameter overrides the 'encoding' parameter (which will be ignored).

Name	Type	Description
params	String	The optional parameters which can be supplied to a specific engine directly. This option is relevant only if a specific type is supplied. The following format should be used: param1:value1,param2:value2. See the comments section for valid values.

Returns

Name	Description
LiveRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter contains an invalid value, or the blend parameter contains a non existent blend name.
MISSING_PARAMETER	In the case when one of the mandatory parameters is missing. relevant only for the related_content engine type.

Comments

- When the API is used with no specific engine type, a blend of recommendations is returned. The COMPASS administration GUI allows defining the percentage of recommendations returned by each engine. In the case when different engines recommend similar content, duplicates are removed.
- In some cases there is a discrepancy between the engine name, as it appears in the COMPASS GUI, and in this API's "type" parameter. Therefore, for clarity, the mapping of the names is provided as follows: "user_preferences" = Viewer Preferences, "promotions" = Promotions, "viewer_series" = ViewerSeries, "related_content" = Related Content.
- The API only returns the live content which is being broadcast within a configurable time frame since the recommendations are requested. This availability window is configurable via the COMPASS administration GUI.
- The following filtering rules are used by the engines when the parameter "channels" is supplied: If the parameter "subscription" is set to "true", the engines filter the results by the channels in the parameter "channels", intersected with the channels to which the household is subscribed. If the parameter "subscription" is set to "false", the engines filter the results only by the channels in the parameter "channels".
- The following filtering rules are used by the engines when the parameter "channels" is not supplied: If the parameter "subscription" is set to "true", the engines filter the results by the channels to which the household is subscribed. If the parameter "subscription" is set to "false", the engines filter the results only by the general content availability (i.e., also returns programs on channels to which the household is not subscribed).
- The start_time and end_time parameters are used when requesting live recommendations in a future time window. These two parameters should be given together. If they are not given, the default configurable time window will apply. The configuration parameters defined in the Administration GUI under the "general" section apply also to the future time window.
- Promoted programs or channels can result in recommendations beyond the household's subscription, provided that the operator configured this option in the Administration GUI.
- The first index requested. This parameter is used if pagination of recommendation results is required.
- In the case when the 'device_models' parameter is passed to the action, the 'encoding' parameter will be ignored, and content items that are not supported by the device model will not be returned.
- The related content engine supports the parameters "program_id" and "external_content_id". One of these parameters must be given when calling the engine directly. In the case when both parameters are given, the INVALID_PARAMETER exception is returned. In the case when no parameter is given, the MISSING_PARAMETER exception is returned.
- The related content engine also supports the parameter "order_by". This parameter can get the values "relevance" or "startTime". The default value is "relevance".

Examples

```
GetLiveRecommendationList
```

```
GetLiveRecommendationList?encoding=MPEG-2,MPEG-4&max_results=10
```

```
GetLiveRecommendationList?type=user_preferences
```

```
GetLiveRecommendationList?type=user_preferences&encoding=MPEG-2
```

```
GetLiveRecommendationList?type=promotions
```

```
GetLiveRecommendationList?max_pr_level=1500
```

```
GetLiveRecommendationList?genres=1
```

```
GetLiveRecommendationList?type=promotions&subscription=false
```

```
GetLiveRecommendationList?type=promotions&subscription=true&channels=channel21,channel36
```

```
GetLiveRecommendationList?start_time=1292516554000&end_time=1292523754000
```

```
GetLiveRecommendationList?device_models=model1,model2,model3
```

```
GetLiveRecommendationList?type=related_content&params=program_id:1111
```

```
GetLiveRecommendationList?type=related_content&params=external_content_id:2222,order_by:startTime
```

GetLiveSeriesAvailabilityList *since 2.4*

This action retrieves episodes from a given series that are broadcasted in a specified time range.

Parameters

Name	Type	Description
program_id	String	The program ID of an episode, as returned in the 'programId' parameter of a specific episode recommendation.
season_id	String	The external ID of a season as returned in the 'externalSeasonId' parameter of a specific episode recommendation.
series_id	String	The external ID of a series as returned in the 'externalSeriesId' parameter of a specific episode recommendation.
encoding	String	The comma-delimited list of encoding formats or external IDs supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The formats values must match those recognized by the system, for example: MPEG-2,MPEG-4.

Name	Type	Description
max_pr_level	Number	The maximum level of parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", filter by the channels to which the household is subscribed. If set to "false", the results are returned from all the channels in the middleware. Default value: "true".
channels	String	Comma-delimited list of channel external IDs, according to which the API filters the results. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: channel-extId1, channel-extId2, etc.
live_catchup_only	Boolean	If set to "true", only live Catch-Up episodes are returned. The default value is false.
start_time	Number	The start time in milliseconds for a future episodes broadcast availabilities.
end_time	Number	The end time in milliseconds for a future episodes broadcast availabilities. This parameter is mandatory in the case when the start_time parameter is given, and it should be later than the start_time parameter.
order_by	String	This defines the property by which the returned episode list is sorted. Valid values are: "contentSeq","startTime". "startTime" represents the broadcast time of the episode, "contentSeq" represents the episode sequence. The default value is 'startTime'.

Returns

Name	Description
LiveContentAvailability[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when one of the parameters contains an invalid value.

Comments

- It is possible to provide either program_id of a specific episode, external season_id of a specific season, or external series_id of a specific series. Providing one of these parameters is mandatory. If more than one of these parameters is given, an INVALID_PARAMETER exception is returned.
- If program_id of a specific episode is given, the API returns advanced available episodes from the same series.
- If season_id of a specific season is given, the API returns available episodes of that season and onwards.
- If series_id of a specific series is given, the API returns all available episodes from that series.
- In the case when the start_time and end_time are not given, the API will return episodes which are broadcast within a configurable time frame since the API was called. This availability window is configurable via the COMPASS administration GUI.
- The following filtering rules are used by the API when the parameter "channels" is supplied: If the parameter "subscription" is set to "true", the API filters the results by the channels in the parameter "channels", intersected with the channels to which the household is subscribed. If the parameter "subscription" is set to "false", the API filters the results only by the channels in the parameter "channels".
- The following filtering rules are used by the API when the parameter "channels" is not supplied: If the parameter "subscription" is set to "true", the API filters the results by the channels to which the household is subscribed. If the parameter "subscription" is set to "false", the API filters the results only by general broadcast availability (i.e., also returns programs on channels to which the household is not subscribed).

Examples

```
GetLiveSeriesAvailabilityList?
program_id=external_id1&start_time=1292516554000&end_time=1292523754000
 &order_by=contentSeq
```

GetProfile

since 1.0

This action returns the active profile.

Returns

Name	Description
Profile	

Examples

```
GetProfile
```

GetProfileChannelList

since 1.0

This action returns the list of channels explicitly chosen by the logged-in profile.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Channel[]	

Examples

```
GetProfileChannelList
```

GetProfileContentExcludeList *since 2.1*

This action returns a list of content that was excluded by the logged-in profile.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ExcludedContent[]	

Examples

```
GetProfileContentExcludeList
```

GetProfileExcludeList *since 2.3*

This action returns a list of items that were excluded by the logged-in profile.

Parameters

Name	Type	Description
type	String	The type of the excluded items to return. Possible values: all, genre, content.

Returns

Name	Description
ExcludedData[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter value is different than the valid values (all, genre, content).

Examples

```
GetProfileExcludeList?type=all
```

```
GetProfileExcludeList?type=content
```

See Also

- AddToProfileExcludeList
- RemoveFromProfileExcludeList
- ClearProfileExcludeList

GetProfileGenreList

since 1.0

This action returns the list of genres explicitly chosen by the logged-in profile.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Genre[]	

Examples

```
GetProfileGenreList
```

GetProfileImage

since 2.3

This action returns an actual binary profile image according to the image type.

Parameters

Name	Type	Description
image_id	String	The ID of the image to return.

Returns

Name	Description
ProfileImage	

Throws

Name	Description
INVALID_PROFILE_IMAGE	In the case when the given image ID is not found.

Comments

- An image is returned.

Examples

```
GetProfileImage?image_id=1
```

See Also

- CreateProfileImage
- DeleteProfileImage
- GetProfileImageList
- UpdateProfileImage

GetProfileImageList

since 2.3

This action returns a list of all images created for the logged-in profile.

Returns

Name	Description
ProfileImage[]	

Examples

```
GetProfileImageList
```

See Also

- CreateProfileImage
- DeleteProfileImage
- GetProfileImage
- UpdateProfileImage

GetProfileImplicitChannelList

since 1.1

This action returns the list of implicit channels for the logged-in profile.

Parameters

Name	Type	Description
order_by	String	This defines the property by which the returned list is sorted. Valid values are: "number", "weight". "weight" represents the channel importance for the logged-in profile, based on the collected usage data. Defaults to "number".
from	Number	The first index requested.

Name	Type	Description
count	Number	The number of elements requested.

Returns

Name	Description
Channel[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the order_by parameter contains an invalid value.

Examples

```
GetProfileImplicitChannelList?order_by=number
```

GetProfileImplicitGenreList

since 1.1

This action returns the list of implicit genres for the logged-in profile.

Parameters

Name	Type	Description
order_by	String	This defines the property by which the returned list is sorted. Valid values are: "name", "weight". "weight" represents the genre importance for the logged-in profile, based on the collected usage data. Defaults to "name".
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Genre[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the order_by parameter contains an invalid value.

Examples

```
GetProfileImplicitGenreList?order_by=name
```

GetProfileList

since 1.0

This action returns the list of profiles associated with the logged-in household. The returned list is sorted by created order such that the primary profile will always be listed first.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
Profile[]	

Examples

```
GetProfileList
```

GetProfileOrderList

since 1.1

This action returns the list of content items ordered by the logged-in profile. The returned list is ordered from the most recent order to the earliest one.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ProfileOrder[]	

Examples

```
GetProfileOrderList
```

GetProfileRatingList

since 1.1

This action returns the list of ratings made by the logged-in profile. The returned list is ordered from the most recent rating to the earliest one.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ProfileRating[]	

Examples

```
GetProfileRatingList
```

GetProfileTypeList

since 3.1 SR9

This action returns all the profile types defined in the system

Returns

Name	Description
ProfileType[]	

Examples

```
GetProfileTypeList
```

GetProfileWishList

since 2.2

This action returns a list of video items from the profile wish list including movies, episodes, and seasons.

- To get an available representative season of each series in the wish list, set the parameter wishlist-series-instead-of-season in the Configuration to true.
- To get the available seasons from the wish list, set the parameter wishlist-series-instead-of-season in the Configuration to false (default).

Parameters

Name	Type	Description
profile_id	String	The external ID of the desired profile. This parameter is used when trying to access the wish list of a user other than the logged-in user. The API will return the wish list only when the profile allowed his wish list to be published, otherwise an exception will be raised.
order_by	String	The values: alphabetic, add_to_wish_list_time, content_type, availability_end_time. Default value: "alphabetic".
order	String	The values: asc, dec. Default value: "asc".

Name	Type	Description
content_type	String	Comma-delimited list of content types, according to which the API filter the results. If specified, it returns a list of recommendations from the specified content types. Valid values are: "movie", "season" or "program".
device_models	String	The comma-delimited list of device model external IDs, according to which the engines filter the results. (contents that support at least one of the device models in the list will be returned).
from	Number	The first index requested.
count	Number	The number of elements requested.
signature	String	The calculated URL signature. Used instead of authentication when calling the API anonymously. See the section: **Calculating URL signature**.

Returns

Name	Description
WishListRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the order_by, order or content_type parameters contain an invalid value.
INVALID_PROFILE_ID	In the case when the given profile_id does not exist.
PROFILE_WISHLIST_NOT_PUBLIC	In the case when profile_id is given and the profile did not allow his wish list to be published.

Examples

```
GetProfileWishList?order_by=add_to_wish_list_time&order=asc
```

GetProfileWishlistUsageData

since 2.1

This action returns the content wish list usage data of the logged-in profile. This API should be used in the case when the wish list is managed externally to COMPASS. If the wish list is managed in COMPASS, use GetProfileWishList instead.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
WishListUsageContent[]	

Examples

```
GetProfileWishlistUsageData
```

GetReceivedSocialMessages

SocialTV since 2.0

This action returns a list of messages that were sent to the logged-in Social Identity. The archived messages do not contain images.

Parameters

Name	Type	Description
message_status_list	String	This comma-delimited list of receiver message statuses for returned messages. Values: 'active', 'hidden', 'read'. Default: 'active'.
image_name	String	The name of the image to include in the results for all the returned Social Identities. If not supplied, do not return images in the results. If the image name does not exist for the Social Identity, no images are returned in the results.
message_type_list	String	The comma-delimited list of message types for returned messages. Values: 'invitation', 'acceptinvitation', 'message', 'videorecommendation', 'liverecommendation', 'vodgift'. Default: all types.
property	String	A single pair of property name and value in the format name:value. This is used to filter messages according to the specified property.
network_type	String	The type of network with which the External Identity of the sender is associated. Values: 'facebook'

Name	Type	Description
sender_id_list	String	The comma-delimited list of Social Identity IDs from whom messages were sent to the logged-in Social Identity.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ReceivedSocialMessage[]	The returned entity name is SocialMessage.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the message_status_list, message_type_list or network_type parameters contain an invalid value.

Comments

- The number of returned messages is limited to 500.

Examples

```
GetReceivedSocialMessages?message_status_list=active,read
```

```
GetReceivedSocialMessages?image_name=profile&from=0&count=20
```

```
GetReceivedSocialMessages?
message_type_list=videorecommendation,liverecommendation
```

```
GetReceivedSocialMessages?network_type=facebook
```

```
GetReceivedSocialMessages?
message_type_list=videorecommendation&property=external_content_id:5647&property=external_co
```

```
GetReceivedSocialMessages?
message_type_list=liverecommendation&property==external_content_id:4567&property=program_id:
```

```
GetReceivedSocialMessages?message_type_list=vodgift
```

GetReceivedSocialMessagesArchived

***SocialTV** since 2.1*

This action returns a list of archived messages that were sent to the Social Identity. The archived messages do not contain images.

Parameters

Name	Type	Description
message_status_list	String	The comma-delimited list of receiver message statuses for returned messages. Values: 'active', 'hidden', 'read'. Default: 'active'.
message_type_list	String	The comma-delimited list of message types for returned messages. Values: 'invitation', 'acceptinvitation', 'message', 'videorecommendation', 'liverecommendation', 'vodgift'. Default: all types.
network_type	String	The type of network with which the External Identity of the sender is associated. Values: 'facebook'.
sender_id_list	String	The comma-delimited list of Social Identity IDs from which messages were sent to the logged-in Social Identity.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ReceivedSocialMessage[]	The returned entity name is SocialMessage.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the message_status_list, message_type_list or network_type parameters contain an invalid value.

Comments

- The number of returned messages is limited to 500.

Examples

```
GetReceivedSocialMessagesArchived?message_status_list=active,read
```

```
GetReceivedSocialMessagesArchived?message_type_list=videorecommendation,liverecommendation
```

```
GetReceivedSocialMessagesArchived?network_type=facebook
```

GetRecommendationListByPopularContributor *since 2.2*

This returns a list of video recommendations containing popular contributors that participate in the specified content item.

Parameters

Name	Type	Description
external_content_id	String	The external ID of the on-demand content item.
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value which can be defined via the COMPASS administration GUI.
encoding	String	The comma-delimited list of encoding formats or external ID supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The format's values must match those recognized by the system, for example: MPEG-2,MPEG-4.
max_pr_level	Number	The maximum level of the parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", it filters by the services to which the household is subscribed. If set to "false", the results are returned to all the services in the middleware. Default value: "true".
services	String	The comma-delimited list of service names or external IDs, according to which the engines filter the results. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: service-name1,service-name2, etc.
genres	String	The comma-delimited list of genre external IDs, according to which the engines filter the results (contents containing at least one of the genre's in the list will be returned).
from	Number	The first index requested.
count	Number	The number of elements requested.

Name	Type	Description
filter_genres	Boolean	Whether to filter out recommendations by genres that are defined in the system-wide configuration by time slices. This option is relevant only if the genres parameter is not supplied. This value overrides the global default value.
filter_viewed_content	Boolean	This defines whether to filter viewed content when returning recommendations. This value overrides the global default value which can be defined via the COMPASS administration GUI.
source_channel_id	String	Comma-delimited list of external channel IDs, according to which the engines filter the results.
quality	String	Comma-delimited list of qualities, according to which the engines filter the results. For example: HD,SD,3D.
delivery_mode	String	The comma-delimited list of delivery modes, according to which the engines filter the results. The following format should be used: download, streaming.
device_models	String	Comma-delimited list of device model external IDs, according to which the engines filter the results (contents that support at least one of the device models in the list will be returned). This parameter overrides the 'delivery_mode' and 'encoding' parameters (which will be ignored).

Returns

Name	Description
PopularContributorRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when there is no content with the given external_content_id.

Comments

- The returned recommendations are sorted in descending order by the number of common popular contributors in the original content item and the recommended ones.
- In the case when there are not enough recommendations to return, additional recommendations of most popular contents are returned. These recommendations appear at the end of the list.
- The following filtering rules are used by the server when the parameter "services" is supplied: If the parameter "subscription" is set to "true", the server filters the results by the services in the parameter "services", intersected with the services to which the household is subscribed; if the parameter "subscription" is set to "false", the server filters the results only by the services in the parameter "services".
- The following filtering rules are used by the server when the parameter "services" is not supplied: If the parameter "subscription" is set to "true", the server filters the results by the services to which the household is subscribed. If the parameter "subscription" is set to "false", the server filters the results only by general content availability (i.e., also returns content items on services to which the household is not subscribed).
- Contents that are only available in services marked as un-recommended in the administration GUI will not be recommended.
- In the case when the 'device_models' parameter is passed to the action, 'delivery_mode' and 'encoding' parameters will be ignored, and content items that are not supported by the device model will not be returned.
- The season content will be considered since it supports all the device models that its episodes support.

Examples

```
GetRecommendationListByPopularContributor?
external_content_id=1234&encoding=MPEG-2,MPEG-4&max_results=10
```

GetSentSocialMessages

***SocialTV** since 2.0*

This action returns a list of messages that were sent by the logged-in Social Identity.

Parameters

Name	Type	Description
message_status_list	String	The comma-delimited list of receiver message statuses for returned messages. Values: 'active', 'hidden', 'read'. Default: 'active'.
image_name	String	The name of the image to include in the results for all the returned Social Identities. If it is not supplied, do not return images in the results. If the image name does not exist for the Social Identity, no images are returned in the results.

Name	Type	Description
message_type_list	String	The comma-delimited list of message types for returned messages. Values: 'invitation', 'acceptinvitation', 'message', 'videorecommendation', 'liverecommendation', 'vodgift'. Default: all types.
property	String	A single pair of property names and values in the format name:value. This is used to filter messages according to the specified property.
network_type	String	The type of network to which the External Identity of the sender is associated. Values: 'facebook'
receiver_id_list	String	The comma-delimited list of Social Identity IDs to which messages were sent by the logged-in Social Identity.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
SentSocialMessage[]	The returned entity name is SocialMessage.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the message_status_list, message_type_list or network_type parameters contain an invalid value.

Comments

- The number of returned messages is limited to 500.

Examples

```
GetSentSocialMessages?message_status_list=active,read
```

```
GetSentSocialMessages?image_name=profile
```

```
GetSentSocialMessages?network_type=facebook&from=20&count=20
```

```
GetSentSocialMessages?message_type_list=videorecommendation,liverecommendation
```

```
GetSentSocialMessages?
message_type_list=videorecommendation&property=external_content_id:5647&property=external_co
```

```
GetSentSocialMessages?
message_type_list=liverecommendation&property=external_content_id:4567&property=program_id:7
```

```
GetSentSocialMessages?message_type_list=vodgift
```

GetSentSocialMessagesArchived

***SocialTV** since 2.1*

This action returned a list of archived messages that were sent by the Social Identity. The archived messages do not contain images.

Parameters

Name	Type	Description
message_status_list	String	The comma-delimited list of receiver message statuses for returned messages. Values: 'active', 'hidden', 'read'. Default: 'active'.
message_type_list	String	The comma-delimited list of message types for returned messages. Values: 'invitation', 'acceptinvitation', 'message', 'videorecommendation', 'liverecommendation', 'vodgift'. Default: all types.
network_type	String	The type of network to which the External Identity of the sender is associated. Values: 'facebook'
receiver_id_list	String	The comma-delimited list of Social Identity IDs to which messages were sent by the logged-in Social Identity.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
SentSocialMessage[]	The returned entity name is SocialMessage.

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the message_status_list, message_type_list or network_type parameters contain an invalid value.

Comments

- The number of returned messages is limited to 500.

Examples

```
GetSentSocialMessagesArchived?message_status_list=active,read
```

```
GetSentSocialMessagesArchived?message_type_list=videorecommendation,liverecommendation
```

```
GetSentSocialMessagesArchived?network_type=facebook
```

GetSeriesSeasonList

since 2.3 SR2

This action returns a list of seasons that are part of the given VOD series, ordered by season sequence number.

Parameters

Name	Type	Description
series_id	String	The series external ID as returned in the recommendation.

Returns

Name	Description
Season[]	

Throws

Name	Description
INVALID_SERIES_ID	In the case when there is no series with the given series_id.

Examples

```
GetSeriesSeasonList?series_id=id1
```

GetSocialIdByExtId

SocialTV since 2.0

This action returns the Social ID by the External ID.

Parameters

Name	Type	Description
network_type	String	The type of network with which the External Identity is associated. Values: 'facebook'.
external_id	String	The user's ID in the external network.
signature	String	The calculated URL signature. See the section: **Calculating URL signature**.

Returns

Name	Description
Result	The result includes the ID of the Social Identity and Profile ID. See the comment below.

Throws

Name	Description
INVALID_EXTERNAL_IDENTITY	In the case when no Social Identity is associated with the supplied External ID and Network Type.
INVALID_PARAMETER_VALUE	In the case when the network_type parameter contains an invalid value, or the signature cannot be validated.

Comments

- This action does not require authentication.
- The Social ID entity is returned within the result:

```
<Result Status="SUCCESS" Message="Social identity found">

<Properties><Property name="socialId" value="1234"/><Property
name="profileId" value="1234"/></Properties>

</Result>
```

Examples

```
GetSocialIdByExtId?network_type=facebook&external_id=1&signature=RXlaakpaVH...
```

GetSocialIdentity

SocialTV since 2.0

This action returns the Social Identity details for the logged-in profile.

Returns

Name	Description
SocialIdentity	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	No Social Identity is associated with the logged-in profile.

Examples

```
GetSocialIdentity
```

GetSocialImage

***SocialTV** since 2.0*

This action returns an actual binary image according to the image type.

Parameters

Name	Type	Description
image_id	String	The ID of the image to return.

Returns

Name	Description
SocialImage	

Throws

Name	Description
INVALID_SOCIAL_IMAGE	In the case when the given image ID is not found.

Comments

- An image is returned.

Examples

```
GetSocialImage?image_id=1
```

See Also

- CreateSocialImage
- DeleteSocialImage
- UpdateSocialImage

GetVideoRecommendationList

since 1.0

This action returns a list of video recommendations both for the logged-in profile and for anonymous users.

Parameters

Name	Type	Description
type	String	If specified, it returns a list of recommendations from a specific engine. Valid values are: "all", "most_popular", "user_rating", "user_rating_series", "external_rating", "promotions", "cf", "related_content", "most_recent", "most_popular_series" and "viewer_series". It defaults to "all". For a detailed description of each engine, please refer to the COMPASS Overview guide.
max_results	Number	The maximum results to be returned by the API. This value overrides the global default value which can be defined via the COMPASS Administration GUI.
random	Boolean	This defines whether the recommendations are returned shuffled or not. Each engine can shuffle the recommendations internally. Random = True (default) shuffles the results of each engine separately and the blend results are shuffled too. Random = False returns the results for each engine and for the blend in the same order with every request (until the engine data is updated). The list of results is updated periodically according to the engines' scheduling.
encoding	String	Comma-delimited list of encoding formats or external IDs supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The format values must match those recognized by the system, for example: MPEG-2, MPEG-4.
params	String	The optional parameters which can be supplied to a specific engine directly. This option is relevant only if a specific type is supplied. The following format should be used: param1:value1,param2:value2. See the comments section for valid values.

Name	**Type**	**Description**
max_pr_level	Number	The maximum level of the parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", it filters by the services to which the household is subscribed. If set to "false", the results are returned to all the services in the middleware. Default value: "true".
services	String	Comma-delimited list of service names or external IDs, according to which the engines filter the results. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: service-name1, service-name2, etc.
genres	String	Comma-delimited list of genre external IDs, according to which the engines filter the results (contents containing at least one of the genres in the list will be returned).
from	Number	The first index requested.
count	Number	The number of elements requested.
filter_genres	Boolean	Whether to filter out recommendations by genres that are defined in the system-wide configurations by time slices. This option is relevant only if the genres parameters are not supplied. This value overrides the global default value.
filter_viewed_content	Boolean	Defines whether to filter viewed content when returning recommendations. This value overrides the global default value which can be defined via the COMPASS Administration GUI.

Name	Type	Description
blend	String	The name of a blend according to which the recommendation results from various recommendation engines are blended together. This parameter is only relevant when the type parameter is set to "all". If content_type is specified, then the blend parameter must also be provided. The blend name is case-insensitive. If not specified, the default blend is used.
source_channel_id	String	Comma-delimited list of external channel IDs, according to which the engines filter the results.
quality	String	Comma-delimited list of qualities, according to which the engines filter the results. For example: HD, SD, 3D.
delivery_mode	String	Comma-delimited list of delivery modes, according to which the engines filter the results. The following format should be used: download, streaming.
device_models	String	Comma-delimited list of device model external IDs, according to which the engines filter the results (contents that support at least one of the device models in the list will be returned). This parameter overrides the 'delivery_mode' and 'encoding' parameters (which will be ignored).
content_type	String	Comma-delimited list of content types, according to which the engines filter the results. If content_type is specified, then the blend parameter must also be specified. If specified, it returns a list of recommendations from the specified content types. If "type" parameter is provided, the requested content type should be specified as part of the request "params". [Valid values: movie, season, program]. The values are case-insensitive.
categories	String	Comma-delimited list of categories external IDs, according to which the engines filter the results

Name	Type	Description
group_categories	String	Defines whether to parse the recommendation results grouped by the categories filtered. Valid values are: "0" or "1".
min_production_year	Number	The minimum production year for the returned contents. Contents which are older than this parameter value will be filtered out.
max_production_year	Number	The maximum production year for the returned contents. Contents which are newer than this parameter value will be filtered out.

Returns

Name	Description
VideoRecommendation[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter contains an invalid value, or the blend parameter contains a non-existing blend name.
NOT_IN_CATALOG	In case the request contains external_content_id that not exist in the catalog. The HTTP header response status will be 404.

Comments

- When the API is used with no specific engine type, a blend of recommendations is returned. The COMPASS administration GUI allows defining the percentage of recommendations returned by each engine. In the case when different engines recommend similar content, duplicates are removed.
- In some cases there is a discrepancy between the engine name as it appears in the COMPASS GUI and in this API's "type" parameter. Therefore, for clarity, the mapping of the names is provided as follows: "most_popular" = Most Popular, "user_rating" = Viewer Rating, "external_rating" = External Rating, "promotions" = Promotions, "cf" = Collaborative Filtering, "related_content" = Related Content, "most_recent" = Most Recent, "most_popular_series" = Most Popular Series, "user_rating_series" = Viewer Rating Series, "viewer_series" = Viewer Series.
- The following filtering rules are used by the engines when the parameter "services" is supplied: If the parameter "subscription" is set to "true", the engines filter the results by the services in the parameter "services", intersected with the services to which the household is subscribed. If the parameter "subscription" is set to "false", the engines filter the results only by the services in the parameter "services".
- The following filtering rules are used by the engines when the parameter "services" is not supplied: If the parameter "subscription" is set to "true", the engines filter the results by the services to which the household is subscribed. If the parameter "subscription" is set to "false", the engines filter the results only by general content availability (i.e., also returns content items on services to which the household is not subscribed).
- A promotion for video content may refer to a service that is beyond the household's subscription, provided that the operator configured this option in the administration GUI.
- Content that is only available in services marked as un-recommended in the administration GUI will not be recommended.
- The most popular engine supports the parameter "period" and "content_type", for which recommendations are requested. Valid values for "period" are: "all, "week" and "month". It defaults to "all". Valid values for "content_type" are: "all", "movies", "programs". It defaults to "movies". When the engine Most Popular is called from the blend, then only "Movie" content type is supported.
- The Collaborative Filtering engine and the Related Content engine both support the parameters "video_id" and "external_content_id". "video_id" is pipe-delimited list of the ID's of the video for which similar items are to be returned (matching the video ID of the service availability). "external_content_id" is pipe-delimited list of the external content ID's of the content item for which similar items are to be returned (matching the content item external ID). In the case when the parameters are not given, the engines provide recommendations, based on the personal viewing history of the profile. In the case when the blend is defined as an item-to-item blend in the administration GUI, one of these parameters must be given. In the case when both parameters are given, the INVALID_PARAMETER_VALUE is returned.
- The related content engine also supports the parameter "extensive". This is a Boolean value determining whether to return an interesting or a basic list. It defaults to "false".
- The Most Recent engine supports the parameter "group" - the name of one of the groups that is defined in the engine configuration. If this parameter is supplied, the recommendation list includes only content items that are linked to services in the specified group.
- The first index requested. This parameter is used if pagination of recommendation results is required. This parameter is relevant mainly when Random = False, since the order of recommendations is then fixed.
- In the case when the 'device_models' parameter is passed to the action, 'delivery_mode' and 'encoding' parameters will be ignored, and content items that are not supported by the device model will not be returned.
- Season content will be considered since it supports all the device models that its episodes support.

Examples

```
GetVideoRecommendationList
```

```
GetVideoRecommendationList?random=true&encoding=MPEG-2,MPEG-4&max_results=10
```

```
GetVideoRecommendationList?type=most_popular
```

```
GetVideoRecommendationList?type=most_popular&encoding=MPEG-2,MPEG-4
```

```
GetVideoRecommendationList?type=most_popular&params=period:all
```

```
GetVideoRecommendationList?type=most_popular&params=period:week
```

```
GetVideoRecommendationList?type=most_popular&params=period:month
```

```
GetVideoRecommendationList?type=most_popular&params=period:all,content_type:all
```

```
GetVideoRecommendationList?type=most_popular&params=period:week,content_type:all
```

```
GetVideoRecommendationList?type=most_popular&params=period:all,content_type:Movie
```

```
GetVideoRecommendationList?type=most_popular&params=period:all,content_type:Program
```

```
GetVideoRecommendationList?type=cf&params=video_id:1111
```

```
GetVideoRecommendationList?type=most_recent&params=group:VOD
```

```
GetVideoRecommendationList?type=related_content&params=video_id:1111,extensive:true
```

```
GetVideoRecommendationList?max_pr_level=1500
```

```
GetVideoRecommendationList?genres=1,2
```

```
GetVideoRecommendationList?type=user_rating&subscription=false
```

```
GetVideoRecommendationList?type=user_rating&subscription=true&services=service3,service5
```

```
GetVideoRecommendationList?blend=blend4
```

```
GetVideoRecommendationList?source_channel_id=id1,id2
```

```
GetVideoRecommendationList?device_models=model1,model2,model3
```

```
GetVideoRecommendationList?type=promotions&min_production_year=2010&max_production_year=2013
```

```
GetVideoRecommendationList?content_type=Program,Movie
```

```
GetVideoRecommendationList?type=related_content&params=external_content_id:815807|1726019
```

```
GetVideoRecommendationList?type=cf&params=video_id:2256|4139
```

GetVideoSeasonContentList

since 2.0

This action returns a list of video content item availabilities for a given season.

Parameters

Name	Type	Description
season_id	String	The season external ID as returned in the recommendation.
encoding	String	The comma-delimited list of encoding formats or external IDs supported by the client. The external ID must match the external ID that is defined in the Administration GUI. The formats values must match those recognized by the system, for example: MPEG-2,MPEG-4.
max_pr_level	Number	The maximum level of parental rating allowed in the returned recommended content items.
subscription	Boolean	If set to "true", it filters by the services to which the household is subscribed. If set to "false", the results are from all the services in the middleware. Default value: "true".
filter_viewed_content	Boolean	This defines whether to filter viewed content when returning recommendations. This value overrides the global default value which can be defined via the COMPASS administration GUI.
services	String	The comma-delimited list of service names or external IDs, according to which the engines filter the results. The external ID must match the external ID that was defined in the Administration GUI. The following format should be used: service-name1,service-name2, etc.
genres	String	The comma-delimited list of genre external IDs, according to which the engines filter the results (contents containing at least one of the genres in the list will be returned).
quality	String	The comma-delimited list of qualities, according to which the engines filter the results. For example: HD,SD,3D.

Name	Type	Description
delivery_mode	String	The comma-delimited list of delivery modes, according to which the engines filter the results. The following formats should be used: download, streaming.
device_models	String	The comma-delimited list of device model external IDs, according to which the engines filter the results (contents that support at least one of the device models in the list will be returned). This parameter overrides the 'delivery_mode' and 'encoding' parameters (which will be ignored).
categories	String	Comma-delimited list of categories names, according to which the engines filter the results
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
VideoContentAvailability[]	

Throws

Name	Description
INVALID_CONTENT_ID	In the case when there is no season with the given season_id.

Comments

- In the case when the 'device_models' parameter is passed to the action, the 'delivery_mode' and 'encoding' parameters will be ignored, and content items not supported by the device model will not be returned.
- Season content will be considered since it supports all the device models that its episodes support.

Examples

```
GetVideoSeasonContentList?season_id=id1
```

```
GetVideoSeasonContentList?season_id=id1&encoding=MPEG-2,MPEG-4
```

```
GetVideoSeasonContentList?season_id=id1&max_pr_level=1500
```

```
GetVideoSeasonContentList?season_id=id1&genres=1,6
```

```
GetVideoSeasonContentList?season_id=id1&subscription=true&from=0&count=5
```

```
GetVideoSeasonContentList?season_id=id1&subscription=false&services=service3,service5
```

GetViewedContentList *since 1.0*

This action returns a list of content that was viewed by the logged-in profile.

Parameters

Name	Type	Description
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
ViewedContent[]	

Comments

- The API only returns the content viewed within a configurable time frame. This time frame is configured (in months) via the COMPASS administration GUI.

Examples

```
GetViewedContentList
```

IgnoreInvitation *SocialTV since 2.0*

This action ignores the invitation from another Social Identity.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the Social Identity that sent the invitation.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Comments

- The relationship status with the related Social Identity is changed to 'Ignore'.

Examples

```
IgnoreInvitation?related_social_id=1
```

See Also

- AcceptInvitation
- BlockSocialIdentity
- InviteFriend
- UnblockSocialIdentity

InviteFriend

***SocialTV** since 2.0*

This action sends an invitation to another Social Identity to become friends. A relationship between the parties is created. A message of type 'Invitation' is issued from the inviter to the invitee.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the Social Identity to invite.
message	String	The free text message to include in the invitation.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Comments

- The relationship status with the related Social Identity is changed to 'Pending'.
- The relationship status between the related Social Identity and the inviter is changed to 'Invite'.

Examples

```
InviteFriend?related_social_id=1&message=Become my friend!
```

See Also

- AcceptInvitation
- BlockSocialIdentity
- DeleteFriend
- IgnoreInvitation
- UnblockSocialIdentity

IsAlive

since 3.1

A "keep alive" action that is responsible to return an indication whether COMPASS is up and running. Since 3.1 SR16.

Examples

```
IsAlive
```

Login

since 1.0

This action logs in to COMPASS.

Parameters

Name	Type	Description
household_id	String	The external ID of the household.
profile_id	String	The external ID of the desired profile. In the case when such an ID is not in use, the internal ID should be used.
pin_code	String	The PIN code used to authenticate the profile. This is relevant only in the case when the profile_id parameter is given.
signature	String	The calculated URL signature. The wrong value raises an exception. See section: **Calculating URL signature**.
network_type	String	The type of network with which the Social External Identity is associated (e.g., 'facebook').

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_CREDENTIALS	In the case when the arguments supplied do not match a valid household and profile.

Name	Description
INVALID_PIN_CODE	In the case when the given profile has a pin code and the given pin_code parameter is either empty or contains a different value.

Comments

- The 'household_id' parameter is mandatory when logging in from clients.
- External applications (such as Facebook applications) can perform a login operation using 'profile_id', 'signature' and 'network_type'. The signature is calculated using a secret key that was defined for that network type, and should be provided to the application developer. See the section: **Calculating URL signature**.
- The 'pin_code' parameter is mandatory, if a pin code was defined for the profile when logging using 'household_id' and 'profile_id'.
- When logging in using 'profile_id', 'signature' and 'network_type', the pin code is overlooked, even if one was assigned to the profile.
- It is possible to extend the login mechanism on a project basis, using plugins.

Examples

```
Login?household_id=1
```

```
Login?household_id=1&profile_id=1&pin_code=1234
```

```
Login?profile_id=1&network_type=facebook&signature=SENzdEhSMS9CRWF1T1Mzaj...
```

See Also

- Logout

Logout

since 1.0

This action logs out from COMPASS.

Returns

Name	Description
Result	

Examples

```
Logout
```

See Also

- Login

OptInProfile

since 1.1

This action changes the logged-in profile status to Active so that the usage data can be collected for the profile.

Returns

Name	Description
Result	

Examples

```
OptInProfile
```

OptOutProfile

since 1.1

This action removes the logged-in profile implicit data (genre and channel lists) and changes the profile status to Suspended so that usage data is not collected for it.
In addition, the profile order history is deleted and the profile rating history is made anonymous by connecting it to a new dummy profile.

Returns

Name	Description
Result	

Examples

```
OptOutProfile
```

See Also

- GetProfileImplicitChannelList
- GetProfileImplicitGenreList
- GetProfileRatingList

RegisterEvent

since 1.0

This action registers a usage event in COMPASS.

Parameters

Name	Type	Description
type	String	The type of event. Valid values are: "order", "rate", "zap", "play", "preview", "reminder", "record", "record_series", "launch", "content_end", "update_wish_list", "impression" , "click" ,"redirect_to_watch"
program_id	String	In the case of live content, this is the external ID of the program to which the event refers. When the "type" parameter is set to "impression", this parameter can appear several times to support a list of programs.

Name	**Type**	**Description**
external_content_id	String	The external ID for the on-demand content item. When the "type" parameter is set to "impression", this parameter can appear several times to support a list of contents. Note: This ID should uniquely identify the content item regardless of the offer or distribution in the service.
external_series_id	String	The external ID of the item series, in case of series recommendation.
recommended	Boolean	This is true if the content item was recommended by COMPASS. This is false if the event took place elsewhere in the service, for example in the VOD catalog. A "true" value is referred to as a "conversion".
engine	String	In the case when a specific engine recommended the content, this field should contain the name of the engine. Valid values for video content are: "most_popular", "user_rating", "user_rating_series", "external_rating", "promotions", "cf", "related_content", "most_recent", "most_popular_series", "viewer_series" and "social_friend_recommendation". Valid values for live content are: "user_preferences", "promotions", "related_content" and "social_friend_recommendation".
shuffle_num	Number	The number of shuffles performed by the profile until "conversion".
is_paid	Boolean	Whether the household paid explicitly for consumption of a specific content item. This defaults to "false".
price	Float	The amount the user paid for the content. This parameter is relevant when the "type" parameter is set to "order".
rating	Integer	The value is between 1 and 5, representing the profile's rating for the selected content item. Supported for video content only.

Name	Type	Description
blend	String	In the case when COMPASS recommended the content item, this field should contain the name of the blend that led to the recommendation. The blend name is case-insensitive. If not specified, the default blend is assumed.
add_to_list	Boolean	This indicates whether the update wish list event is "add" to wish list or "remove" from wish list. A "true" value refers to adding content to the profile wish list. A "false" value refers to removing content from the profile wish list.
add_to_series_record	Boolean	This indicates whether the record series event is adding a series recording or cancelling a series recording. A "true" value refers to adding a series recording. A "false" value refers to cancelling a series recording.
device_model	String	The external ID of the device model from which the event was sent.
text	String	Free text adding project-specific data on the event. For instance: the UI screen from which the event was sent. The maximum length of the text is 10 characters. The text should be URL encoded.

Returns

Name	Description
Result	This differs according to the profile. • Active profile: `<Result Status="SUCCESS" Message="Event of type 'order' registered successfully"/>` • Suspended profile: `<Result Status="FAILURE" Message="No event registered for suspended profile"/>` • Anonymous user: `<Result Status="SUCCESS" Message="Event of type 'order' registered successfully"/>`

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the "type" parameter or "engine" parameter contain an invalid value.

Comments

- Events are processed asynchronously. The resulting impact on the engines' recommendations is not immediate.
- The following are the parameters that are applicable for each event type. Mandatory parameters are marked in bold.
    - order: **external_content_id**, recommended, shuffle_num, is_paid, engine, blend, price, device_model, text
    - content_end: **external_content_id**, device_model, text
    - rate: **external_content_id**, **rating**, recommended, engine, blend, device_model, text
    - update_wish_list: **external_content_id**, recommended, shuffle_num, engine, blend, **add_to_list**, device_model, text
    - preview: **external_content_id**, recommended, shuffle_num, engine, blend, device_model, text
    - zap: **program_id**, recommended, shuffle_num, engine, blend, device_model, text
    - reminder: **program_id**, recommended, shuffle_num, engine, blend, device_model, text
    - record: **program_id**, recommended, shuffle_num, engine, blend, device_model, text
    - play: **program_id**, recommended, shuffle_num, engine, blend, device_model, text
    - record_series: **program_id**, recommended, shuffle_num, engine, blend, **add_to_series_record**, device_model, text
    - impression: **program_id** or **external_content_id**, engine, blend, device_model, text
    - click: **program_id** or **external_content_id** or **external_series_id**, recommended, engine, blend, device_model, text
- Events are not collected for suspended profiles, unless "anonymize usage data for suspended users" is enabled in the COMPASS administration GUI. In this case, events will be registered as if received from an anonymous user.
- In record series event, the program_id parameter refers to a single episode of the recorded season.
- In some cases there is a discrepancy between the engine name as it appears in the COMPASS GUI and in this API's "engine" parameter. Therefore, for clarity, the mapping of the names is provided as follows: "most_popular" = Most Popular, "user_rating" = Viewer Rating, "external_rating" = External Rating, "promotions" = Promotions, "cf" = Collaborative Filtering, "related_content" = Related Content, "most_recent" = Most Recent, "most_popular_series" = Most Popular Series, "user_rating_series" = Viewer Rating Series, "viewer_series" = Viewer Series, "user_preferences" = Viewer Preferences. "social_friend_recommendation" is for content recommendations received from friends using Social TV. This is used to enable monitoring Social TV effectiveness.
- As of COMPASS 2.4, all events can be sent in POST method.
  This allows sending several events in one call, where each
  event is on its own line. For example: POST RegisterEvent
  type=impression&external_content_id=1&program_id=2&program_id=3
  type=click&external_content_id=1

Examples

```
RegisterEvent?type=order&external_content_id=4345
```

```
RegisterEvent?
type=order&external_content_id=1234&recommended=true&engine=most_popular&shuffle_num=2
```

```
RegisterEvent?
type=order&external_content_id=1235&recommended=true&blend=blend2&is_paid=true&price=5.23
```

```
RegisterEvent?
type=zap&program_id=5678&recommended=true&engine=user_preferences
```

```
RegisterEvent?type=rate&external_content_id=1122&rating=5
```

```
RegisterEvent?type=update_wish_list&external_content_id=1234&add_to_list=true
```

```
RegisterEvent?type=record_series&program_id=1234&add_to_series_record=true
```

```
RegisterEvent?type=launch
```

```
RegisterEvent?
type=impression&external_content_id=1&external_content_id=2&blend=blend1
```

RemoveContentFromProfileExcludeList

since 2.1

This action removes the specified content from the content exclusion list of the logged-in profile.

Parameters

Name	Type	Description
content_id	String	The external ID of the on-demand content item to be removed from the content exclusion list.

Returns

Name	Description
Result	

Comments

- In the case when the given content_id does not exist in the profile exclude list, SUCCESS is returned.

Examples

```
RemoveContentFromProfileExcludeList?content_id=id1
```

RemoveFromProfileExcludeList

since 2.3

This action removes the specified item from the exclusion list of the logged-in profile.

Parameters

Name	Type	Description
id	String	The external ID of the genre/content item to be removed from the exclusion list.
type	String	The type of excluded item to be removed. Possible values: genre, content.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the type parameter value is different than the valid values (genre, content).

Comments

- In the case when the given id does not exist in the profile exclude list, SUCCESS is returned.

Examples

```
RemoveFromProfileExcludeList?type=genre&id=1234
```

See Also

- AddToProfileExcludeList
- ClearProfileExcludeList
- GetProfileExcludeList

RemoveFromWishList

since 2.2

Removes content items from the profile wish list. Specify the external ID of a VOD movie, episode, or season to remove from the wish list. If the parameter wishlist-series-instead-of-season is set to true in the Configuration, the representative season of the series is removed from the wish list. By default watched content items are removed from the wish list automatically. To keep content items on the wish list even after they are watched, set the parameter remove-from-wishlist-on-order in the Configuration to false. In this case content items that were watched are not automatically removed from the wish list and can be removed manually using this API.

Parameters

Name	Type	Description
external_content_id	String	A semicolon separated list of the IDs of items to be removed from the wish list, including VOD movies, episodes, and seasons.

Throws

Name	Description
INVALID_CONTENT_ID	In the case when content with the given external_content_id does not exist.

Examples

```
RemoveFromWishList?external_content_id=123;124
```

See Also

- GetProfileWishList
- AddToWishList

RemoveProfilePinCode

since 2.0

This action removes the logged-in profile PIN code.

Parameters

Name	Type	Description
current_pin_code	String	The current PIN code of the profile.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PIN_CODE	In the case when the given profile has a PIN code and the given current_pin_code parameter contains a different value.

Examples

```
RemoveProfilePinCode?current_pin_code=1234
```

See Also

- UpdateProfilePinCode
- ResetProfilePinCode

ResetProfilePinCode

since 2.0

This action resets a profile PIN code to a predefined default value.

Parameters

Name	Type	Description
profile_id	String	The external ID of the desired profile.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PROFILE_ID	In the case when the given profile does not exist or does not belong to the logged-in household.

Examples

```
ResetProfilePinCode?profile_id=1
```

See Also

- UpdateProfilePinCode
- RemoveProfilePinCode

SearchSocialIdByExtId

SocialTV *since 2.0*

This action searches for Social Identities by their External ID.

Parameters

Name	Type	Description
network_type	String	The type of network with which the External Identity is associated. Values: 'facebook'.
external_id_list	String	The comma-delimited list of user IDs in the external network. The maximum value in the list: 500.
image_name	String	The name of image to include in the results for all the returned Social Identities. If not supplied, do not return images in the results. If the image name does not exist for the Social Identity, no images are returned in the results.
from	Number	The first index requested.

Name	Type	Description
count	Number	The number of elements requested.

Returns

Name	Description
SearchSocialIdentity[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the given network type is not a valid value.

Examples

```
SearchSocialIdByExtId?network_type=facebook&external_id_list=id1,id2
```

```
SearchSocialIdByExtId?network_type=facebook&external_id_list=id1,id2&image_id=profile
```

SearchSocialIdentities

***SocialTV** since 2.0*

This action returns a list of Social Identities that match the search criteria.

Parameters

Name	Type	Description
search_text	String	The text to search in the following fields: first name, last name, social name. The 'starts with' pattern is used, i.e., only fields which start with the specified text will be returned. An OR operator is used between these fields.
cell_phone	String	The cellular phone. This is combined with other search fields with the OR operator.
city	String	City. This is combined with other search fields with the AND operator.
image_name	String	The name of image to include in the results for all the returned Social Identities. If not supplied, do not return images in the results. If the image name does not exist for the Social Identity, no images are returned in the results.

Name	Type	Description
birthday_notification	Number	Birthday offset in days. Return all identities which birthday is before or after the given number of days. This is combined with other search fields with the AND operator. If birthday_notification = 0 then the search will be done on the birthday date itself. If birthday_notification = 1 then the search will be done one day before the user's birthday.
network_type	String	The type of network to which the signature is relevant. This parameter is mandatory in case signature is provided. Values: 'facebook'.
signature	String	The calculated URL signature. Used instead of authentication when calling the API anonymously. See the section: **Calculating URL signature**.
from	Number	The first index requested.
count	Number	The number of elements requested.

Returns

Name	Description
SocialIdentity[]	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the network_type parameter contains an invalid value, or the signature cannot be validated.

Comments

- The number of returned Social Identities is limited to 500.

Examples

```
SearchSocialIdentities?search_text=a
```

```
SearchSocialIdentities?search_text=a&image_name=profile
```

```
SearchSocialIdentities?birthday_notification=1&signature=...
```

SendRecommendations *since 2.0*

This action send recommendations on content to friends in the Social Network. The recommendations are translated into a `SocialMessage`, and viewed as any other message.

Parameters

Name	Type	Description
message	String	Free text whose maximum size is determined by the system configuration. The default is 140 characters.
receiver_social_id_list	String	A comma-delimited list of friend's Social IDs. If not supplied, the message is sent to all the friends.
external_content_id	String	The external ID of the content to which the recommendation refers.
program_id	String	The external ID of the program to which the recommendation refers (in case of live recommendation).
video_id	String	The external ID representing the specific offer for the on-demand content (availability ID).
source	String	The source of the message.
properties	String	*(Deprecated since 2.1)* Use 'property' parameters instead.
property	String	A single pair of property names and values in the format name:value. More than one 'property' parameter can be used.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PARAMETER_VALUE	One of parameters value is incorrect.
INVALID_SOCIAL_IDENTITY	At least one of the receiver Social Identities do not exist.

Comments

- One of the parameters: 'program_id', 'external_content_id' or 'video_id' is mandatory.
- The parameters 'program_id', 'external_content_id' or 'video_id' are registered as properties on the message, where the name of the parameter is the key of the property, and the value is the property value.
- When using the parameter 'program_id', the social message is generated with type 'liverecommendation'. When using the parameters 'video_id' or 'external_content_id', the social message is generated with type 'videorecommendation'.
- The system does not validate that the recommended content exists or is available in the middleware. Furthermore, by the time that the receiver views the recommendation, the content might not be available for consumption.
- The 'source' is a free text parameter that can be used to indicate the sender application and so on.

Examples

```
SendRecommendations?program_id=1
```

```
SendRecommendations?video_id=1&property=key1:val1&property=key2:val2
```

```
SendRecommendations?receiver_social_id_list=1,2&external_content_id=1
```

See Also

- GetReceivedSocialMessages
- SendSocialMessage

SendSocialMessage

SocialTV since 2.0

This action sends a Social Message to a list of friends.

Parameters

Name	Type	Description
message_type	String	The type of message to create. Values: 'message', 'videorecommendation', 'liverecommendation', 'vodgift'.
message	String	The free text. The maximum size is determined by the system configuration. The default is 140 characters.
related_social_id_list	String	A comma-delimited list of friend Social IDs.
all_friends	String	If true, the message is sent to all the active friends (ignoring the 'related_social_id_list' parameter).

Name	Type	Description
network_type_list	String	A comma separated social external network type. Values: 'facebook'. If set then the message will be published to the given external network(s) using the appropriate plug-in.
properties	String	*(Deprecated since 2.1)* Use the 'property' parameters instead.
property	String	A single pair of property names and values in the format name:value. More than one 'property' parameter can be used.
source	String	The source of the message - free text.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PARAMETER_VALUE	One of parameters value is incorrect.
INVALID_SOCIAL_IDENTITY	At least one of the related Social Identities do not exist.

Comments

- A message can be sent to friends with relationship status 'Active' only.
- When sending messages of type 'videorecommendation' or 'liverecommendation', recommendation information should be supplied in a property parameter. The name of the property should be 'external_content_id' (external content ID of the content) or 'video_id' (middleware external ID) for video recommendation, and 'program_id' (middleware external ID) for a program recommendation. The value of the property should be the external ID of the recommended content, according to the recommendation type. Additional properties can be set for recommendation: 'content_name' (content name), 'content_url' (URL link to content), 'image_name' (name of the image), 'description' (description of the content).
- When sending messages of type 'vodgift', VOD gift information should be supplied in a property parameter. The name of the property should be 'vod_gift_id' and the value should be the external middleware VOD gift ID. Additional properties can be set for VOD gift: 'external_content_id' (external content ID of the content), 'content_name' (content name), 'image_name' (name of the image). Message of type 'vodgift' can be sent only to one active friend.
- The 'source' is a free text parameter that can be used to indicate the sender application and so on.

Examples

```
SendSocialMessage?message_type=message&message=hello
```

```
SendSocialMessage?message_type=message&related_social_id_list=1,2
 &message=hello&property=name1:value1
```

```
SendSocialMessage?message_type=vodgift&related_social_id_list=5&message=you
 have a gift&property=vod_gift_id:5682&property=external_content_id:6875
 &property=content_name:the green hornet&property=image_name:large
```

```
SendSocialMessage?
network_type_list=facebook&message_type=message&message=hello world
```

```
SendSocialMessage?
network_type_list=facebook&message_type=message&message=hi&all_friends=true
```

See Also

- GetReceivedSocialMessages

UnblockSocialIdentity

***SocialTV** since 2.0*

This action unblocks a previously blocked Social Identity.

Parameters

Name	Type	Description
related_social_id	Number	The ID of the Social Identity to unblock.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_SOCIAL_IDENTITY	The related Social Identity does not exist.

Comments

- The relationship status with the related Social Identity is changed to 'Ignore'.

Examples

```
UnblockSocialIdentity?related_social_id=1
```

See Also

- AcceptInvitation
- BlockSocialIdentity
- IgnoreInvitation
- InviteFriend

UpdateProfile

since 1.0

This action updates one of the profiles of the logged-in household.

Parameters

Name	Type	Description
name	String	The new name for the profile. The name must be unique per household.
image	String	The location of the profile's image (avatar).
suspended	Boolean	Whether the profile is suspended or not. The usage data is not collected for a suspended profile.
property	String	The property value is key-value pair that represents the profile custom properties (key and value).
type	String	The type of the profile. Should be one of the profile types given by the GetProfileTypeList API.

Returns

Name	Description
Result	

Throws

Name	Description
DUPLICATE_KEY	In case two or more keys are equals.
INVALID_PROFILE_ID	In the case when the given profile does not exist or does not belong to the logged-in household.
NOT_IN_CUSTOM_PROPERTY	In case the requested key not exist in custom property. * @exception {PROFILE_EXISTS} In the case when the new name for the profile already exists in the household.
PROFILE_EXISTS	In the case when the new name for the profile already exists in the household.
PROPERTY_KEY_NOT_IN_PROFILE	In case the request deleted key not exist for the profile

Comments

- Once the profile is updated, it is no longer defined as 'automatic'.
- In order to delete profile custom property use the word null as the value for the deleted key (see example).

Examples

```
UpdateProfile?name=William Smith&image=/avatars/Dog.jpg&suspended=false
```

```
UpdateProfile?name=david&property=key1:&property=key2:val2
```

```
UpdateProfile?name=david&property=key1:&property=key2:null
```

See Also

- GetProfileList
- AddProfile
- DeleteProfile

UpdateProfileChannelList

since 1.0

This action updates the list of favorite channels for the logged-in profile.

Parameters

Name	Type	Description
channels	String	The comma-delimited list of the profile's favorite channels. The list completely replaces the old preferences. In the case when the parameter is not supplied, all existing favorite channels are removed.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_CHANNEL_ID	In the case when invalid channel ID is given, or if the household is not subscribed to the given channel.

Examples

```
UpdateProfileChannelList?channels=1,2,3,4
```

```
UpdateProfileChannelList
```

See Also

- GetProfileChannelList

UpdateProfileGenreList

since 1.0

This action updates the list of favorite genres for the logged-in profile.

Parameters

Name	Type	Description
genres	String	The comma-delimited list of the profile's favorite genres. The list completely replaces the old preferences. In the case when the parameter is not supplied, all existing favorite genres are removed.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_GENRE_ID	In the case when invalid genre ID is given.

Examples

```
UpdateProfileGenreList?genres=1,2,3,4
```

```
UpdateProfileGenreList
```

See Also

- GetProfileGenreList

UpdateProfileImage

since 2.3

This action updates a Profile Image for the logged-in profile.

Parameters

Name	Type	Description
image_id	Number	The ID of the image to update.
image_name	String	The name of the image, used to differentiate between the different images of the user.
image_description	String	Free text that describes the image.
image_url	String	The URL where the image is located, in case it is not stored by the system.

Returns

Name	Description
ProfileImage	

Throws

Name	Description
CONTENT_TYPE_MISSING	The content type is not supplied in the request for binary images.
IMAGE_ALREADY_EXIST	The image with the supplied image_name already exists for the logged-in profile.
INVALID_IMAGE_SIZE	The image size exceeds the maximum system configuration limit.

Comments

- If the picture is an external avatar, image_url should be given. For a picture that is uploaded to be stored in the database, the image binaries should be provided (via HTTP Post).
- For binary images, image type is a mandatory attribute of the HTTP header ("content-type"). The type should be a valid image type defined by the HTTP protocol: for example, "image/jpeg".

Examples

```
UpdateProfileImage?image_name=WilliamSmith&image_url=/images/WilliamSmith1.jpg
```

See Also

- CreateProfileImage
- DeleteProfileImage
- GetProfileImage
- GetProfileImageList

UpdateProfilePinCode

since 2.0

This action updates the logged-in profile PIN code.

Parameters

Name	Type	Description
current_pin_code	String	The current PIN code of the profile. This is mandatory in the case when the logged-in profile has a PIN code.
new_pin_code	String	The new PIN code for the profile.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PIN_CODE	In the case when the logged-in profile has a PIN code and the given current_pin_code parameter is empty or contains a different value.

Examples

```
UpdateProfilePinCode?new_pin_code=5347
```

```
UpdateProfilePinCode?current_pin_code=1234&new_pin_code=5347
```

See Also

- ResetProfilePinCode
- RemoveProfilePinCode

UpdateSocialIdentity

SocialTV since 2.0

This action updates the Social Identity for the logged-in profile.

Parameters

Name	Type	Description
first_name	String	The first name of the user.
last_name	String	The last name of the user.
password	String	The password used when creating external identities.
cell_phone	String	The user's phone number.
email	String	The user's e-mail address.
city	String	The user's city.
gender	String	The user's gender. Values: 'male', 'female'.
birthday	Number	The user's birthday (date in milliseconds). There are no validations on the value except for field type.
publish_wishlist	Boolean	Whether the profile allows his wish list to be published to other users and social networks. Default value: false.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the gender parameter contains an invalid value.

Comments

- One of the parameters is mandatory for an update operation.

Examples

```
UpdateSocialIdentity?first_name=William&last_name=Smith
```

See Also

- CreateSocialIdentity
- GetSocialIdentity

UpdateSocialImage

***SocialTV** since 2.0*

This action updates a Social Image for the logged-in Social Identity.

Parameters

Name	Type	Description
image_id	Number	The ID of the image to update.
image_name	String	The name of the image, used to differentiate between the different images of the user.
image_description	String	Free text that describes the image.
image_url	String	The URL where the image is located, in case it is not stored by the system.

Returns

Name	Description
SocialImage	

Throws

Name	Description
CONTENT_TYPE_MISSING	The content type is not supplied in the request for binary images.
IMAGE_ALREADY_EXIST	The image with the supplied image_name already exists for the logged-in Social Identity.
INVALID_IMAGE_SIZE	The image size exceeds the maximum system configuration limit.

Comments

- If the picture is an external avatar, image_url should be given. For a picture that is uploaded to be stored in the database, the image binaries should be provided (via HTTP Post).
- For binary images, image type is a mandatory attribute of the HTTP header ("content-type"). The type should be a valid image type defined by the HTTP protocol: for example, "image/jpeg".

Examples

```
UpdateSocialImage?image_name=WilliamSmith&image_url=/images/WilliamSmith1.jpg
```

See Also

- CreateSocialImage
- DeleteSocialImage
- GetSocialImage

UpdateSocialMessageStatus

***SocialTV** since 2.0*

This action updates the status of a Social Message.

Parameters

Name	Type	Description
message_id_list	String	The list of comma-delimited message IDs.
receiver_status	String	The message status of the receiver. Values: deleted, active, hidden, read.
sender_status	String	The message status of the sender. Values: deleted, active, hidden, read.

Returns

Name	Description
Result	

Throws

Name	Description
INVALID_PARAMETER_VALUE	In the case when the given receiver_status or sender_status parameters contain an invalid value.
INVALID_SOCIAL_MESSAGE	The social Message with the supplied ID was not sent by the logged-in Social Identity when updating 'sender_status', or Social Message with the supplied ID was not received by the logged-in Social Identity when updating 'receiver_status'.

Comments

- Either **receiver_status** or **sender_status** are mandatory. If the message is sent to the logged-in Social Identity, receiver_status is needed. If the message is sent by the logged-in Social Identity, sender_status is needed.

Examples

```
UpdateSocialMessageStatus?message_id_list=1,2&receiver_status=read
```