



Content Discovery API HowTo

EPG Basic Functionalities

The core of the Content Discovery REST API consists of its data and the set of features required to correctly display them. This section provides a non-exhaustive overview of common filtering operations for EPG data.

Now/Next View

It's easy to get data for what's on now and what's on next. This request returns metadata that can be used to populate a program guide "now/next" view for three channels (values **37**, **38** and **40** for term **sourceId**)

[http://hack.api.watchmi.tv/asset?beforeAfter={"limit":2,"origin":"currentTime","offset":0}&filter={"term":"sourceId","operator":"in","values":\[37,38,40\]}&annotationAttributes=\["sourceId","assetId","titles"\]&numberOfItems=6](http://hack.api.watchmi.tv/asset?beforeAfter={)

Primetime EPG

The Content Discovery API can be used to build electronic program guides (EPG's) such as the one at watchmi.tv:

<http://www.watchmi.tv/de/tv-programm>

The following request returns the full metadata set for shows occurring on 16 November 2014 at primetime (20:15-22:15):

[http://hack.api.watchmi.tv/asset?filter={"term":"assetType","value":"broadcast"}&annotation=full&windowOfAvailabilityStart=2014-11-16T19:15:00Z&windowOfAvailabilityEnd=2014-11-16T21:15:00Z&numberOfItems=25](http://hack.api.watchmi.tv/asset?filter={)

You can use the **filter** parameter to indicate particular criteria that must be met for shows to be included in the search results:

[http://hack.api.watchmi.tv/asset?filter={"operator":"and","criteria":\[{"term":"assetType","value":"broadcast"},{"operator":"in","term":"sourceId","values":\["610","611"\]}\]}&annotation=short&numberOfItems=25](http://hack.api.watchmi.tv/asset?filter={)

The **annotationAttributes** parameter allows the user to specify the data fields to be returned by the request. In this case, the API returns the broadcast ID, locale-specific title, broadcast channel, airing time and program length (in seconds).

[http://hack.api.watchmi.tv/asset?filter={"term":"assetType","value":"broadcast"}&windowOfAvailabilityStart=2014-11-16T19:15:00Z&windowOfAvailabilityEnd=2014-11-16T21:15:00Z&annotationAttributes=\["assetId","titles.deu","sourceId","availabilityStartTime","availabilityEndTime","durationSeconds"\]&numberOfItems=25](http://hack.api.watchmi.tv/asset?filter={)

EPG by time

With this request the API will return the IDs of shows starting within the specified time window (window begins on 16/11/2014 at 20:15:00, ends on 16/11/2014 at 22:15:00):

[http://hack.api.watchmi.tv/asset?filter={"term":"assetType","value":"broadcast"}&windowOfAvailabilityStart=2014-11-16T19:15:00Z&windowOfAvailabilityEnd=2014-11-16T21:15:00Z&numberOfItems=25](http://hack.api.watchmi.tv/asset?filter={)

date and **time** should be specified in the format yyyy-MM-ddTHH:mm:ssZ (UTC)

Querying metadata for a particular program

When the ID of a show is known, all the metadata fields can be queried.

[http://hack.api.watchmi.tv/asset?assetIds=\["88188807"\]&annotationAttributes=\["assetId","sourceId","titles","availabilityStartTime","availabilityEndTime"\]](http://hack.api.watchmi.tv/asset?assetIds=[)

assetId is the ID provided in the JSON element **assetId** returned by previous requests.

It is possible to query any property listed in the JSON data.

List of Channels

The channels line-up is accessible via the following request:

[http://hack.api.watchmi.tv/documentstore/sources?filter={"term":"assetTypes","value":"broadcast"}&annotationAttributes=\["sourceId","sourceNames.short"\]](http://hack.api.watchmi.tv/documentstore/sources?filter={)

EPG by Channel

To get today's broadcasts of SAT1 (**sourceId=39**) the corresponding channel ID is required:

[http://hack.api.watchmi.tv/asset?filter={"term":"sourceId","value":"39"}&annotation=full&numberOfItems=10](http://hack.api.watchmi.tv/asset?filter={)

Movie Tips of the Day

The Content Discovery API exposes the editorial tips of the day for different categories. In this example the movie tips from TV DIGITAL (structTip.type = "mov" AND structTip.brand = "tvd") are queried within the given time window:

[http://hack.api.watchmi.tv/asset?numberOfItems=10&filter={"operator":"and","criteria":\[{"term":"structTip.type","value":"mov"},{"term":"structTip.brand","value":"tvd"}\]}&relativeWindowOfAvailabilityStart={"origin":"startOfDay","offset":"PT0H"}&relativeWindowOfAvailabilityEnd={"origin":"startOfDay","offset":"PT24H"}&annotationAttributes=\["structTip"\]&annotation=short&numberOfItems=25](http://hack.api.watchmi.tv/asset?numberOfItems=10&filter={)

Theme Channels Listings

Theme Channels contain an editorial selection of broadcast or online contents. For an example of this, see the Theme Channels page on watchmi.tv:

<http://www.watchmi.tv/de/themechannels>

Theme Channel Groups

Theme Channels are grouped into categories (Kino & TV, Music, Sport, etc.). The list of channel groups is accessible via this URL:

[http://hack.api.watchmi.tv/documentstore/sourceGroups?filter={"term":"sourceGroupType","value":"theme"}](http://hack.api.watchmi.tv/documentstore/sourceGroups?filter={)

Example response:

```
[
  {
    "sourceGroupType": "theme",
    "sourceGroupId": "1",
    "sourceGroupNames": {
      "deu": "Nachrichten"
    },
    "sequenceNo": -1,
    "sources": [
      "3a06c406-6126-ba94-157e-fc85034dafa6",
      "c00f6362-1f96-cf44-b17a-979bf76ccc0a",
      "8b242736-fb62-3b74-c57e-2cdbe2470bfb",
      "ae7077de-9897-8684-218e-a723f923d438",
      "de2e664a-2e1f-e6d4-41a6-f858d4b0274b",
      "bfd681c7-975e-0544-35f5-0f900cc79d6e"
    ]
  },
  ...
]
```

The channel identifiers are in the response, contained in the list "sources".

Theme Channels List

To get a list of all theme channels, you can use this URL:

[http://hack.api.watchmi.tv/documentstore/sources?filter={"term":"assetTypes","value":"theme"}&annotationAttributes=\["sourceId","sourceNames.short","media"\]&](http://hack.api.watchmi.tv/documentstore/sources?filter={)

URLs for channel logos that can be presented in lists and other displays can be found in the media attribute – check for media of type "thumbnail".

List of Content Items Published By A Theme Channel

Finally, the list of content published by a theme channel is returned by the following request:

<http://hack.api.watchmi.tv/list/moderated/<source ID>>

where **<source ID>** is the ID of the theme channel we are interested in. For example:

<http://hack.api.watchmi.tv/list/moderated/01d9afae-3c13-c244-fde2-cbb70f9ca7e8?annotation=theme>

Similarity-based Recommendations

You can use a program's **assetId** attribute to request a list of recommendations based on similarity. Pass the **assetId** to the similarity engine in the **seedAsset** query parameter as follows:

[http://hack.api.watchmi.tv/recommendations?context=pdp_similarity_context&seedAsset=<assetId>&adHocContext={\"rules\":{\"then\":{\"global\":{\"numberOfItems\":10,\"annotation\":\"all\"}}}}](http://hack.api.watchmi.tv/recommendations?context=pdp_similarity_context&seedAsset=<assetId>&adHocContext={\)

EPG Personal Recommendations

User-based recommendations are computed in accordance with the subscriber's ratings. With every positive and negative rating the Content Discovery API computes assets that fit the subscriber's preferences.

It is probably easiest to start by cloning (duplicating) our pre-created subscriber preference template. By beginning with a baseline set of preferences, the recommendation system can likely arrive at a robust understanding of the subscriber's preferences more quickly than if it were to begin with an empty profile.

This is intended to work as follows: the first time the user performs an action that indicates a preference, such as rating a program item, the system clones the default subscriber and all its preference profiles. It then applies the preference that the user has just indicated to the relevant profile associated with the subscriber.

Get the subscriber template ID

First you will need to obtain the subscriber template ID on the system in order to clone it:

<http://hack.api.watchmi.tv/subscriber>

The request sends back a response in JSON format, for example:

```
{
  subscriberId: "dd677004-e5df-4f30-9ba8-f4222f64b98f",
  profiles: [
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_sports",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_entertainment",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_series",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_movie",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_kids",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_VARIOUS",
    "dd677004-e5df-4f30-9ba8-f4222f64b98f_info"
  ]
}
```

```
}
```

The value of **subscriberId** is the subscriber template ID, which is used in the next step.

Create a subscriber based on the subscriber template

To create a new subscriber, first generate a new subscriber ID to be used by the system. This can be done using any UUID-generating tool.

Then pass **newSubscriberID** to the system in a POST request along with the ID of the subscriber to be cloned, which was obtained from the previous step and is referred to below as **templateSubscriberID**

POST to <http://hack.api.watchmi.tv/subscriber/<newSubscriberID>>

```
{  
  "template": "<templateSubscriberID>"  
}
```

The system responds with the subscriber ID of the newly-created subscriber, along with a list of profiles associated with the subscriber.

For example:

```
{  
  "subscriberId": "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e",  
  "profiles": [  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_sports",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_entertainment",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_series",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_movie",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_kids",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_VARIOUS",  
    "ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_info"  
  ]  
}
```

The trailing string in the profile name identifies the type of content for which the profile is intended to be used to generate recommendations: ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_series for TV series, ef6ac01b-e2cf-4fa4-7ba8-f42c2f64a47e_dd677004-e5df-4f30-9ba8-f4222f64b98f_movie for movies, et cetera...

Delete a user

Send HTTP DELETE request to

<http://hack.api.watchmi.tv/subscriber/<subscriberId>>

Rate a program

A user can positively rate a program (like):

POST to <http://hack.api.watchmi.tv/profile/<profile ID>/scorer?errorTolerant=true&recordingSuggestion=false>

```
{
  "assetId": "<asset ID>",
  "rating": 1,
  "subscriber": "<subscriber ID>"
}
```

Or negatively rate a program (dislike):

POST to <http://hack.api.watchmi.tv/profile/<profile ID>/scorer?errorTolerant=true&recordingSuggestion=false>

```
{
  "assetId": "<asset ID>",
  "rating": -1,
  "subscriber": "<subscriber ID>"
}
```

<asset ID> is the ID provided in the JSON property named **assetId** which was returned by the previous request.

Get personal recommendations

Generate a set of personalized recommendations for a subscriber by sending an API request with the subscriber ID and profile ID with which to generate the recommendations. It is possible to specify a time window for the recommendations by using the **availabilityStartTime** and **availabilityEndTime** parameters.

[http://hack.api.watchmi.tv/recommendations?context=pdp_preference_conext&subscriber=<subscriberId>&prefProfile=<profileId>&availabilityStartTime=<windowStartTime>&availabilityEndTime=<windowEndTime>&adHocContext={\"rules\":{\"then\":{\"preferenceEngines\":{\"name\": \"PrefTV\"}},\"global\":{\"numberOfItems\":10,\"annotation\": \"daily picks\"}}}}](http://hack.api.watchmi.tv/recommendations?context=pdp_preference_conext&subscriber=<subscriberId>&prefProfile=<profileId>&availabilityStartTime=<windowStartTime>&availabilityEndTime=<windowEndTime>&adHocContext={\)

For example:

[http://hack.api.watchmi.tv/recommendations?context=pdp_preference_context&subscriber=294d2490-65f2-11e4-9803-0800200c9a66&prefProfile=294d2490-65f2-11e4-9803-0800200c9a66_dd677004-e5df-4f30-9ba8-f4222f64b98f_series&availabilityStartTime=2014-11-07T19:15:00Z&availabilityEndTime=2014-11-07T21:15:00Z&adHocContext={\"rules\":{\"then\":{\"preferenceEngines\":{\"name\": \"PrefTV\"}},\"global\":{\"numberOfItems\":10,\"annotation\": \"daily picks\"}}}}](http://hack.api.watchmi.tv/recommendations?context=pdp_preference_context&subscriber=294d2490-65f2-11e4-9803-0800200c9a66&prefProfile=294d2490-65f2-11e4-9803-0800200c9a66_dd677004-e5df-4f30-9ba8-f4222f64b98f_series&availabilityStartTime=2014-11-07T19:15:00Z&availabilityEndTime=2014-11-07T21:15:00Z&adHocContext={\)

<date and time> should be specified in the format yyyyMMdd'T'HHmmSS'Z' (UTC)

Trailers

The Content Discovery API provides movie trailer search functionality to facilitate content discovery.

Search for a trailer by name

Trailers can be searched by the name of the movie or program. For example, "Sherlock Holmes":

[http://hack.api.watchmi.tv/search?assetTypes=\["trailer"\]&annotationAttributes=\["titles","media"\]&query=Sherlock+Holmes](http://hack.api.watchmi.tv/search?assetTypes=[)

Broadcasts can have trailers assigned which you can look up.

If a broadcast has trailers assigned / matched, their IDs will be listed inside the JSON property **relatedAssets.trailer**

For example:

```
asset: {
  titles: {
    deu: "Captain Phillips",
    eng: "Captain Phillips"
  },
  relatedAssets: {
    trailer: [
      "46887_29353_25578_12735",
      "46890_29353_25578_12735"
    ]
  },
  ...
}
```