



# API HowTo

# EPG Basic Functionalities

The core of the watchmi 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.

## Primetime EPG

The following request returns the full metadata definition for shows occurring today at primetime (20:15-22:15):

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/format/json/primetime>

This modified version of the previous request selects the field to display: locale, thumbnail format, broadcast channel, title, subtitle, short synopsis, airing time, production country, etc.

[http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/format/json/primetime?locale=de&images=min\\_size170x110&field=chanlong&field=chanshort&field=tit&field=stit&field=shsyn&field=time&field=prcdt&field=chr&field=brdcst&field=rcmd&field=tchn&pos=0&limit=1](http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/format/json/primetime?locale=de&images=min_size170x110&field=chanlong&field=chanshort&field=tit&field=stit&field=shsyn&field=time&field=prcdt&field=chr&field=brdcst&field=rcmd&field=tchn&pos=0&limit=1)

## EPG by time

With this request the API will return the IDs of shows starting within the specified time window (start on 21/06/2013 at 10:00:00, maximum start time at 22:59:59):

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/format/json?field=bidonly&start=<date and time>&end=<date and time>>

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

Example:

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/format/json?field=bidonly&start=20131005T100000Z&end=20131005T225959Z>

## Details of a show

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

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/broadcast/<asset ID>/format/json>

**<asset ID>** is the ID provided in the json element bid returned by previous requests.

Example:

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/broadcast/78147355/format/json>

## List of channels

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

<http://hackathon.lab.watchmi.tv/api/example.com/channels/format/json>

## EPG by channel

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

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/channel/39/day/today/format/json>

## Movie Tips of the day

The watchmi API exposes the editorial tips of the day for different categories. In this example the movie tips (tip/31.1) are queried within the given time window:

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/tip/31.1/format/json?field=bidonly&start=<date and time>&end=<date and time>&limit=30>

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

Example:

<http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/tip/31.1/format/json?field=bidonly&start=20131005T050000Z&end=20131005T235959Z&limit=30>

# Theme channels listings

Theme channels contain an editorial selection of broadcast or online contents.

## Theme channel groups

Theme channels are grouped into categories (Kino & TV, Music, Sport, etc.), accessible via this URL:

[http://hackathon.lab.watchmi.tv/api/example.com/chgrps/group/tcatid:/field/channel\\_group\\_name/field/sub\\_groups/field/chid](http://hackathon.lab.watchmi.tv/api/example.com/chgrps/group/tcatid:/field/channel_group_name/field/sub_groups/field/chid)

To get all the channels belonging to the group Kino & TV the group ID is required:

[http://hackathon.lab.watchmi.tv/api/example.com/channels/group/tcatid:4/field/channel\\_info/field/chid/field/channel\\_details/sort/channel\\_name\\_long](http://hackathon.lab.watchmi.tv/api/example.com/channels/group/tcatid:4/field/channel_info/field/chid/field/channel_details/sort/channel_name_long)

## List of assets published by a theme channel

Finally, the list of trailers published by the Bild.de Kino theme channel is returned by the following request:

[http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/group/lgroup:tcid:d4ce9e96-a98b-8c04-751a-d7cbf06c82f1/sort/pubdate/field/default/field/ad\\_streams/images/any](http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/group/lgroup:tcid:d4ce9e96-a98b-8c04-751a-d7cbf06c82f1/sort/pubdate/field/default/field/ad_streams/images/any)

# EPG Personal Recommendations

User-based recommendations are computed in accordance with the user ratings. With every positive and negative rating the watchmi API computes assets that fit the user preferences.

## Create a user

POST <http://hackathon.lab.watchmi.tv/api/example.com/do/user/create>

Returns the ID for the newly-created user and the expiration time

```
{
  "rcmdServiceUserId": "dc33ebd8-dac1-11e2-b172-525400ba6c05",
  "umUserId": "dc5e06e8-dac1-11e2-b172-525400ba6c05",
  "userValidUntil": 1374445693072,
  "userValidUntil_utc": "2013-07-21T22:28:13",
  "log": ""
}
```

## Delete a user

POST [http://hackathon.lab.watchmi.tv/api/example.com/do/user/delete?user.RcmdService\\_User\\_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05](http://hackathon.lab.watchmi.tv/api/example.com/do/user/delete?user.RcmdService_User_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05)

## Rate a program

A user can positively rate a program (like):

[http://hackathon.lab.watchmi.tv/api/example.com/do/vote/up?bid=<asset ID>&user.RcmdService\\_User\\_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05](http://hackathon.lab.watchmi.tv/api/example.com/do/vote/up?bid=<asset ID>&user.RcmdService_User_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05)

Or negatively rate a program (dislike):

[http://hackathon.lab.watchmi.tv/api/example.com/do/vote/down?bid=<asset ID>&user.RcmdService\\_User\\_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05](http://hackathon.lab.watchmi.tv/api/example.com/do/vote/down?bid=<asset ID>&user.RcmdService_User_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05)

**<asset ID>** is the ID provided in the json element bid returned by previous requests.

Example:

[http://hackathon.lab.watchmi.tv/api/example.com/do/vote/up?bid=78558794&user.RcmdService\\_User\\_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05](http://hackathon.lab.watchmi.tv/api/example.com/do/vote/up?bid=78558794&user.RcmdService_User_UUID=dc33ebd8-dac1-11e2-b172-525400ba6c05)

## Get personal recommendations

The following request returns personalized recommendations counting a total of 3 assets within the specified time window, sorting them by decreasing score and specifying a minimum score of 0.5:

```
http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/  
user.RcmdService_User_UUID/dc33ebd8-dac1-11e2-b172-525400ba6c05/group/  
sgroup:scid40/field/score/field/liked/start/<date and time>/end/<date and time>/limit/3/sort/  
score-/minscore/0.5
```

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

Example:

```
http://hackathon.lab.watchmi.tv/api/example.com/broadcasts/  
user.RcmdService_User_UUID/dc33ebd8-dac1-11e2-b172-525400ba6c05/group/  
sgroup:scid40/field/score/field/liked/start/20131005T180000Z/end/20131005T230000Z/  
limit/3/sort/score-/minscore/0.5
```

Some of the possible values for the field “*group*” are:

`sgroup:scid40`

Returns recommendations for the given user

`sgroup:ptid4movie5`

Returns movie recommendations for the given user

`sgroup:ptid4series6`

Returns series recommendations for the given user

`sgroup:ptid4sports6`

Returns recommended sport programs for the given user