

Storage (REST API)

[Module description](#) [REST API](#) [XML-RPC API](#)

Retrieving Media Asset List

A query to the media database for a list of media assets. The query returns an XML-document with short descriptions of media assets in the whole database or a certain folder. Graphic templates have the mime_type field with the value 10.

GET

```
http://<IP_ADDR>:7901/storage/<NAME>[/<FOLDER>[/<FOLDER>...]] [?format=val [&datetime=val...][&mimetype=val...]]
```

- **GET** - HTTP GET request.
- **<IP_ADDR>** - server IP address.
- **<NAME>** - media database name.
- **<FOLDER>** - folder name.
- **format** - optional parameter, defines XML output formats:
 - **StoredTree** - XML tree of folders and assets in the media database (default),
 - **mam** - linear asset output,
 - **folderlist** - retrieving the list of folders. Added in version [2.3.90](#).
- **datetime** - optional parameter, filters output by the asset's date of creation. When specifying a value in ISO Date Time format, the server will output assets created after the specified date.
- **mimetype** - optional parameter, filters output by the asset's MIME-type:
 - **Audio**
 - **Video**
 - **AudioVideo**
 - **StillImage**
 - **Text**
 - **Complex**
 - **Tape**
 - **Stream**
 - **Graphics**
 - **Note**.

An example of a server response to a media database query in StoredTree mode:

[| response_storedtree_mediadb.xml](#)

```
<StoredTree query_root="">
  <Folder name="">
    <Clip link_id="0" finalized="true">
      <Attr name="bps">16</Attr>
      <Attr name="channels">2</Attr>
      <Attr name="derived_from">CityViews2 HD</Attr>
      <Attr name="frame_time">0.0400000000</Attr>
      <Attr name="frames">3425</Attr>
      <Attr name="height">1080</Attr>
```

```

    <Attr name="orig_name">CityViews2 HD</Attr>
    <Attr name="orig_type">3</Attr>
    <Attr name="sample_rate">48000</Attr>
    <Attr name="width">1920</Attr>
    <Attr name="first_frame">0</Attr>
    <Attr name="last_frame">3425</Attr>
    <Attr name="title">CityViews3</Attr>
    <Attr name="create_time">2018-03-27T10:26:30</Attr>
    <Attr name="rec_time">2016-12-17T16:15:14</Attr>
  </Clip>
  <Clip link_id="3" finalized="true">
    <Attr name="bps">16</Attr>
    <Attr name="channels">2</Attr>
    <Attr name="derived_from">Hockey</Attr>
    <Attr name="frame_time">0.04000000000</Attr>
    <Attr name="frames">1501</Attr>
    <Attr name="height">1080</Attr>
    <Attr name="orig_name">Hockey</Attr>
    <Attr name="orig_type">3</Attr>
    <Attr name="sample_rate">48000</Attr>
    <Attr name="width">1920</Attr>
    <Attr name="first_frame">0</Attr>
    <Attr name="last_frame">1501</Attr>
    <Attr name="title">Hockey</Attr>
    <Attr name="create_time">2018-03-27T14:13:15</Attr>
    <Attr name="rec_time">2016-12-18T01:14:24</Attr>
  </Clip>
  <Folder name="New Folder"/>
  <Folder name="New Folder 1"/>
</Folder>
</StoredTree>
</xml>

```

An example of a server response to a media database query in the MAM mode:

[| response_mam_mediadb.xml](#)

```

<mam>
  <asset>
    <id>CityViews3</id>
    <dir/>
    <length>00:02:17</length>
    <hires>
      $(video_path)/media/CityViews2_HD-NID0-3/CityViews2_HD.avi
    </hires>
    <proxy>
      $(video_path)/proxy/CityViews2_HD-NID0-19/CityViews2_HD_lrv.avi
    </proxy>
  </asset>
  <asset>

```

```

        <id>Hockey</id>
        <dir/>
        <length>00:01:00.04</length>
        <hires>$(video_path)/media/Hockey-NID3-51/Hockey.avi</hires>
    </asset>
</mam>

```

An example of a server response to a media database query in the Folderlist mode:

| [response_folderlist_mediadb.xml](#)

```

<StoredTree query_root="">
  <Folder name="">
    <Folder name="New Folder"/>
    <Folder name="New Folder 1"/>
  </Folder>
</StoredTree>

```

Query examples:

- <http://localhost:7901/storage> — the list of media databases connected to the server and the XML description of their content.
- http://localhost:7901/storage/DB_NAME - general XML structure of the media database.
- http://localhost:7901/storage/DB_NAME/Folder_Name - general XML structure of the Folder_Name folder.
- http://localhost:7901/storage/DB_NAME/?datetime=YYYY-MM-DD - general XML structure of the list of clips added since the specific date (c_date parameter control)

Retrieving Asset Information

A query to retrieve detailed information on the object (including names of parameters for graphic templates). The query returns an XML-document.

```
GET http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>
```

Since version [2.9.69](#), TITLE for all queries can be replaced by a namespace:id pair, where namespace is the key field name, and id is a unique value. For example:

```
GET http://<IP_ADDR>:7901/storage/<NAME>@house_number:H12345
```

Retrieving Asset Data

```
GET http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>
```

An example of server response to a request to the media database for detailed information on an object:

| response_asset_mediadb.xml

```
<Clip folder="clips" hires="$(video_path)/media/CityViews2_HD-
NID0-3/CityViews2_HD.avi" proxy="$(video_path)/proxy/CityViews2_HD-
NID0-19/CityViews2_HD_lrv.avi" archive="" link_id="0" finalized="true">
  <Marker uid="1" pos="1929">
    <Attr name="user"/>
    <Attr name="title"/>
    <Attr name="comment">111222333</Attr>
    <Attr name="track">General</Attr>
    <Attr name="keywords"/>
    <Attr name="rating">0</Attr>
  </Marker>
  <Segment seg_num="1" seg_name="Seg1" start_frame="0"
duration="150">
    <title>Seg1</title>
    <title>333222111</title>
    <track>Segments</track>
    <rating>0</rating>
  </Segment>
  <Attr name="bps">16</Attr>
  <Attr name="channels">2</Attr>
  <Attr name="derived_from">CityViews2 HD</Attr>
  <Attr name="frame_time">0.0400000000</Attr>
  <Attr name="frames">3425</Attr>
  <Attr name="height">1080</Attr>
  <Attr name="orig_name">CityViews2 HD</Attr>
  <Attr name="orig_type">3</Attr>
  <Attr name="sample_rate">48000</Attr>
  <Attr name="width">1920</Attr>
  <Attr name="first_frame">0</Attr>
  <Attr name="last_frame">3425</Attr>
  <Attr name="title">CityViews3</Attr>
  <Attr name="create_time">2018-03-27T10:26:30</Attr>
  <Attr name="rec_time">2016-12-17T16:15:14</Attr>
</Clip>
```

Retrieving XML Metadata Values

A certain XML field value may be retrieved in a following way:

GET

http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>/GETUSERFIELDS[?field=ns:n
ame]

- **GET** - HTTP GET request.
- **<IP_ADDR>** - server IP address.
- **<NAME>** - media database name.
- **<TITLE>** - asset (clip) name.

- **<NS:ID>** - a value pair where NS is the key field name and ID is a unique value set in the field.
- **<COMMAND_NAME>** - executable command.
 - **GETUSERFIELDS** - retrieving XMP fields values. Added in [2.9.69](#) version.
 - param1 - **field** - requested field ID in the ns : name format (set in the metadata schema).

Receiving WebProxy

The request returns media data (MP4 or HLS).

```
GET http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>[/PVW]/<WPN>
```

- **GET** - HTTP GET request.
- **<IP_ADDR>** - server IP address.
- **<NAME>** - media database name.
- **<TITLE>** - asset (clip) name.
- **<NS:ID>** - a value pair where NS is the key field name and ID is a unique value set in the field.
- **[/PVW]** - optional parameter that allows viewing the WP file in a web browser.
- **<WPN>** - web-profile name. Possible values: WP1-WP8. If the requested profile is not configured in the media database settings, you will receive an error message: HTTP ERROR 500. If a WebProxy copy with a specified profile has not been created earlier for this asset, it will be generated in real time. This process may take some time.

Examples

Retrieving the web proxy version (download by the link):

```
GET http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>/WP1
```

Viewing the web proxy version (the link opens the file in a browser):

The `<video controls="true"></video>` tag is used for displaying.

```
GET http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>/PVW/WP1
```

Deleting Asset

```
DELETE http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>
```

Load XMP Metadata

Detailed description of the function of XMP metadata batch load to MAM via [Metadata Connectors](#). Added in version [2.0.60](#).

```
PUT http://<IP_ADDR>:7901/storage/<NAME>?connector=<CONNECTOR_NAME>
```

Actions with Assets

POST

http://<IP_ADDR>:7901/storage/<NAME>@<TITLE|NS:ID>?command=<COMMAND_NAME>[¶m=val...]

- **POST** - HTTP POST request.
- **<IP_ADDR>** - server IP address.
- **<NAME>** - media database name.
- **<TITLE>** - asset (clip) name.
- **<NS:ID>** - a value pair where NS is the key field name and ID is a unique value set in the field.
- **<COMMAND_NAME>** - executable command.
 - **RESTORE** - restoring HIRES from an archive copy. Added in version [2.0.160](#).
 - **COPYLINK** - creating a new link to the asset.
 - param1 - **title** - new link title.
 - param2 - **folder** - folder for creating a new link.
 - **SETFIELD** - change link field value.
 - param1 - **name** - field name ("type", "clip_name", "comment", "layer", "first_frame", "last_frame", "icon_pos", "user_0" ... "user_7"). Setting values for user_0 (User 1 in the interface) fixed in version [2.9.87](#).
 - **param2** - "val" - new field value.
 - **SETUSERFIELDS** - changing XML fields' values. Added in version [2.6.1](#).
 - param1 - parameter name defines the name of the XML field to be changed. val - comma-separated list of new values.
 -
 - paramN
 - **CREATESEGMENT** - adding a segment marker to asset's metadata.
 - param1 - **seg_num** - number of the created segment.
 - param2 - **start_frame** - segment start (time code in HH:MM:SS[::]FF, HH:MM:SS.ss, or frame number format).
 - param3 - **duration** - segment duration (time code in HH:MM:SS[::]FF, HH:MM:SS.ss, or frame number format).
 - param4 - **title** - segment name (optional).
 - param5 - **keywords** - keywords for search (optional).
 - **CREATESEGMENTATION** - adding a group of segments to asset's metadata. The entire asset is divided to segments of set duration.
 - param1 - **duration** - segment duration (time code in HH:MM:SS[::]FF, HH:MM:SS.ss, or frame number format).
 - param2 - **title** - segment name (optional).
 - **DELETESegment** - deleting the segment marker from asset's metadata.
 - param1 - **seg_num** - number of the segment to be deleted.
 - **CREATEMARKER** - adds or changes the key frame marker in asset's metadata. Added in version [2.6.1](#).
 - param1 - "uid" - unique integer marker identifier used to modify an existing marker. A value of -1 will generate a new identifier while creating a marker.
 - param2 - "frame" - marker position (in frames)
 - param3 - "track" - marker type, possible values:
 - "General",
 - "Secondary Event",
 - "Cue Point",

- “Goto Cue”,
 - “Game Time”,
 - “Icon”,
 - “Picture”,
 - “Profanity Start”,
 - “Profanity End”
 - ...
 - paramN - to set Name/value fields for the marker, send them as «name=value».
- Default fields available for markers:
- “title”,
 - “comment”,
 - “user”,
 - “keywords”
- **DELETEMARKER** - deletes the key frame marker from asset’s metadata. Added in version [2.6.1](#).
 - param1 - “uid” - unique integer identifier of the marker to be deleted.
 - **BV_EXPORT_VERSION** - exports the material version description to BV XML (first configure BV connector on the tab with media database settings). Example call: `http://localhost:7901/storage/Storage_2_MAM@$(title)?command=BV_EXPORT_VERSION`. Added in version [2.8.0](#). Export of versions with an ISARCHIVED attribute is implemented by a simple REST call of BV_EXPORT_VERSION. When exporting via the On Delete flag trigger, note that On Delete is called when the clip is already deleted from the folder and an attempt to access it using a normal title search will result in an “Object not found” error. While the trigger is running, it keeps the clip “open” and the clip is not deleted yet. Thus, you can access such a clip by using the “handle” name space and “Open handle” as an identifier - the variable available in the trigger that is the identifier of the opened file. An example of a REST request: `http://127.0.0.1:7901/storage/Storage_1@$(title)?command=BV_EXPORT_VERSION&ns=handle&id=$(handle)`
 - param1 - “ns” - name space for this ClipID version: “title” by default. Use “handle” and \$ (handle) as id, if a REST request call from trigger is used.
 - param2 - “id” - clip identifier (optional). The “title” id specified in the request path is used by default.
 - **BV_COPY_PROGRAM_FIELDS** - copies XMP fields to the version’s asset from the corresponding program card. Added in version [2.8.23](#).
 - param1 - “ns” - name space for this ClipID version: “title” by default. Use “handle” and \$ (handle) as id, if a REST request call from trigger is used.
 - param2 - “id” - clip identifier (optional). The “title” id specified in the request path is used by default.
 - **BV_MOVE_TO_ASSET** - move the media-file to the folder connected with the Broadview card. Added in version [2.8.40](#).
 - param1 - “asset_id” - ID of the program card loaded from Broadview.
 - param2 - “house_number” - house number assigned to the media-file during transfer (optional).

Generating PNG Thumbnail for Asset

A request to retrieve a PNG thumbnail of the media asset (including parametrised graphics templates) for any specified frame. The request returns a PNG image. Added in version [2.3.63](#).

GET

```
http://<IP_ADDR>:7901/storage/<NAME>@<TITLE>/GETICON[?[pos=FRM][&dur=FRM][&w=PX][&h=PX][&param=NAME:VAL[&param=NAME:VAL]]]
```

- **GET** - HTTP GET request.
- **<IP_ADDR>** - server IP address.
- **<NAME>** - media database name.
- **<TITLE>** - asset (clip) name.
- **pos=FRM** - thumbnail position in frames from the asset start (IN point).
- **dur=FRM** - graphic composition duration in frames. The option is used when a composition has infinite duration, and different animation effects depend on its actual duration.
- **w=PX** - retrieved thumbnail width in pixels (default value equals initial image width).
- **h=PX** - retrieved thumbnail height in pixels (default value equals the initial image height).
- **param=NAME:VAL** - parameters sent to the graphic composition while rendering:
 - **NAME** - graphic element name (e.g. Text Area 0).
 - **VAL** - value sent to the graphic element with the NAME title.

All parameters and values must be encoded to URL Encode if they contain any of the following special characters: ! * ' () ; : @ & = + \$, / ? % # [].

Use the %0a character to create a new line:

```
http://127.0.0.1:7901/storage/Storage_1@g1/GETICON?param=Text:This%20is%20sample%20text%0anew%20line%20here
```

Use the “\\” и “\” escape sequence to pass a «\» or «/» character as a part of a clip title or folder name:

```
http://127.0.0.1:7901/storage/Storage_1@g%5c%2f1/GETICON?param=Text:This%20is%20sample%20text%0anew%20line%20here
http://127.0.0.1:7901/storage/Storage_1/qq%5c%2faa
```

From:

<https://wiki.skylark.tv/> - **wiki.skylark.tv**

Permanent link:

<https://wiki.skylark.tv/api/rest/storage>

Last update: **2023/09/08 10:27**

