HipServ Web Services API v0.1

Contents

Introduction

This document describes the main Web Services available on a HipServ v2.4 to HipServ users. It does **not** describe Web Services targeting HipServ administrators or branding dependent Web Services.

Prerequisites

To call the Web Services you need:

- good knowledge of HTTP
- a HipServ running v2.4 or higher, configured, with access to the Internet
- a client device with access to the Internet
- the fully qualified domain name or IP address of the Central Server (CS) where the HipServ is registered (CS_FQDN in this document)
- the name as which the HipServ is registered (aka hipname)
- the name of a HipServ user (aka username)
- the password of this user

Authentication

Most HipServ Web Services are only available to authenticated users so logging in is the first thing to do (see Session Services). A session cookie is sent by the HipServ in response to a successful user login. This cookie must be sent with all further requests for the HipServ user to be authenticated. User sessions usually expire after 3 hours of inactivity but this number is branding dependent.

REST Web Services API

Overview

All requests are GET, POST, PUT or DELETE HTTP requests and most of them return XML. The body of POST and PUT requests contain XML as well. It is assumed that all XML will be properly formed and encoded as UTF-8.

Success responses are returned with status code 2xx.

Failure responses are returned with status code 4xx or 5xx. Status codes 401 mean that access is unauthorized. Status codes 403 mean that access is forbidden. Status code 400 is used to report all other errors with a response of the following format (known as hipserv/errors+xml):

```
<?xml version="1.0"?>
<errors>
    <error code="c" msg="m"/>
</errors>
```

Where c is the error code (positive or negative) and m is an error message in English (optional).

Note: in this document, code `400.c` means status code `400` with error code `c`. For example, "Get Server Configuration" can only return a failure with status code 400 and code 4 (I/O Error).

"Flex mode"

To accommodate Flex's weak HTTP support (limited to GET and POST HTTP methods, status code other than 200 considered as an error, no ability to set HTTP request headers, no ability to get the status code or body of an "error" response...), a "Flex mode" was added to the API. It is enabled for all Web Service requests containing a "method" query parameter. In that case, the method in the query string overrides the method in the HTTP request (it must be GET, PUT, DELETE or POST in uppercase). Moreover, in "Flex mode", a 200 status code is always returned. Responses for status codes over 400 in Flex mode look like:

```
<errors>
    <error code="401" msg="Unauthorized" />
</errors>
```

The "Flex mode" is optional.

HipServ XML MIME Types

The following table defines the XML MIME types used in the HipServ REST API. They are describe in great detail in the appendix.

MIME Type	Description

hipserv/session+xml	HipServ session
hipserv/serverconfig+xml	HipServ configuration
hipserv/user+xml	HipServ user
hipserv/mediaSources+xml	HipServ media sources
hipserv/preferences+xml	HipServ preferences
hipserv/gallery+xml	Hipserv gallery
hipserv/directoryContents+xml	HipServ directory and its assets and subdirectories
hipserv/lightDirectoryAssets+xml	Light version of a HipServ directory and its assets
hipserv/assetDetails+xml	Hipserv asset details
hipserv/searchResults+xml	HipServ search results
hipserv/tagList+xml	Hipserv tag list (usually associated with an asset)
hipserv/contacts+xml	HipServ contacts listing
hipserv/contact+xml	HipServ contact
hipserv/themes+xml	HipServ themes listing
hipserv/theme+xml	HipServ theme
hipserv/albumListing+xml	HipServ album listing and its albums
hipserv/albumContents+xml	HipServ album and its assets
hipserv/lightAlbumAssets+xml	Light version of a HipServ album and its assets
hipserv/albumRequest+xml	HipServ album creation/edition request
hipserv/3rdPartyPhotosetListing+xml	HipServ 3rd party photoset listing

Session Web Services

Here are the steps to follow to log in:

- Call the **Get an AUTH Login Code and the URL to the HipServ** Web Service using the fully qualified domain name of the Central Server, the HipServ name, username and password, and get an AUTH login code and the URL to the HipServ.
- Call the **Create a new session for a HipServ user (aka: User login) via the CS** Web Service using the URL to the HipServ and the AUTH login code and get the user URI, server config URI and locale, along with the session cookie to authenticate further requests.

The typical Web Services to call next are:

- Get Server Configuration
- Get HipServ User Information

Get an AUTH Login Code and the URL to the HipServ

Returns the URL to a given HipServ and the AUTH login code to send to that HipServ to login as a given HipServ.

Request

Method	POST
URL	https://CS_FQDN_OR_IP/rest/1.0/sessions/hipserv
URL Example	https://www.myhipserv.com/rest/1.0/sessions/hipserv

Body	`<?xml version="1.0"?>`
	`<session hipserv="`**`HIPSERV`**`" username="`**`USERNAME`**`" password="`**`PASSWORD`**`"`
	`/>`

Details	The request must be made over HTTPS. **CS_FQDN_OR_IP** is the fully qualifying domain name or the IP address of the central server with which the HipServ is registered. **HIPSERV** is the HipServ name. **USERNAME** is the username of the HipServ user. **PASSWORD** is the password of the HipServ user.

Success Response

HTTP Status Code	200 OK
MIME Type	text/xml

Body	`<?xml version="1.0" encoding="UTF-8"?>`
	`<sessionhandler code="`**`CODE`**`" description="`**`DESCRIPTION`**`">`
	`  <session hipserv="`**`HIPSERV`**`" username="`**`USERNAME`**`" auth="`**`AUTH`**`"`

	`url="`**`HIPSERV_URL`**`" />`
	`</sessionhandler>`

Details	**CODE / DESCRIPTION** can be: ■ **0 / success** if the login credentials were verified as correct. ■ **1 / bad credentials** if the HipServ name, username or password are incorrect. No specific information given to thwart account-discovery attacks. ■ **2 / hipserv offline** if the HipServ is down, or otherwise inaccessible from the client's current network location. The <session> element will only be returned in the response if code 0 is returned. **HIPSERV_URL** is the complete URL prefix that the client should use to access the HipServ, and takes into account open/blocked ports, LAN/WAN access, direct/NAT traversal, etc. The **AUTH** is the AUTH login code which must be passed to the CPE.

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot Parse XML	There was a problem attempting to parse the request body as an XML document.
400.5	Invalid Parameter	The <session> element, or one or more of the *hipserv*, *username* or *password* attributes, is missing from the XML input.
500	Database error	There was a database failure looking up the login credentials, or storing an AUTH login key to the database.
503	Secure connection required	Because this action requires transmission of sensitive information (both the password and the AUTH login key), it will not proceed unless performed via SSL.

Create a new session for a HipServ user (aka: User login) via the CS

From an AUTH login code received from the central server, logs a HipServ user in (first logging out the current HipServ user if there is one) and returns the new session.

Request

Method	POST
URL	HIPSERV_URL/api/2.0/rest/sessions

Body	`<?xml version="1.0"?>`
	`<session code="`**`AUTH`**`"/>`

Details	**HIPSERV_URL** is the URL of the HipServ received from the central server. **AUTH** is the AUTH login code received from the central server.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/session+xml

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot Parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
400.22	Invalid Authentication Code	

Invalidate the Current Session (aka Log Out)

Invalidates the current session if the visitor is logged Returns the information of an anonymous visitor in any case.

Request

Method	DELETE
URL	HIPSERV_URL/SESSION_URI
Details	**SESSION_URI** is the URI of the current session, received in the **href** attribute as a result of a user login.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/user+xml
Details	The XML returned is the one of an anonymous visitor since the session was invalidated.

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	

Server Web Services

Get Server Configuration

Returns the configuration of the server.

Request

Method	GET
URL	HIPSERV_URL/api/2.0/rest/server/config

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/serverconfig+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	

Account Web Services

Get HipServ User Information

Returns the information about a given HipServ user. Only this authenticated HipServ user is allowed to call this service.

Request

Method	GET
URL	HIPSERV_URL/USER_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/accounts/users/johndoe
Details	**USER_URI** is the URI of the HipServ user, received in the **userURI** attribute as a result of a user login.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/user+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user.
403	Forbidden	The current authenticated HipServ user is not the user whose information is requested.

Get HipServ User Preferences

Returns the preferences of a given HipServ user. Only this authenticated HipServ user is allowed to call this service.

Request

Method	GET
URL	HIPSERV_URL/USER_URI/prefs
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/accounts/users/johndoe/prefs
Details	**USER_URI** is the URI of the HipServ user, received in the **userURI** attribute as a result of a user login.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/preferences+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user.
403	Forbidden	The current authenticated HipServ user is not the user whose information is requested.

Change HipServ User Preferences

Changes preferences of a given HipServ user. Only this authenticated HipServ user is allowed to call this service.

Request

Method	PUT
URL	HIPSERV_URL/USER_URI/prefs
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/accounts/users/johndoe/prefs

Body	`<?xml version="1.0"?>`
	`<preferences>`
	`  <preference name="NAME1" value="VALUE1" />`
	`  <preference name="NAME2" value="VALUE2" />`
	`  <!- ... ->`
	`</preferences>`

Details	**USER_URI** is the URI of the HipServ user, received in the **userURI** attribute as a result of a user login. **NAME1** is the name of the first preference and **VALUE1** is its value etc.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/preferences+xml
Details	The response only includes the preferences listed in the request.

Failure Responses

Code	Description	Cause
400.3	Cannot Parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user.
403	Forbidden	The current authenticated HipServ user is not the user whose information is requested.

Get HipServ User Gallery

Returns the information needed to display the gallery of a HipServ user, i.e. the list of albums s/he has access to. Only this authenticated HipServ user is allowed to call this service.

Request

Method	GET
URL	HIPSERV_URL/USER_URI/gallery
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/accounts/users/johndoe/gallery
Details	**USER_URI** is the URI of the HipServ user, received in the **userURI** attribute as a result of a user login.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/gallery+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user.
403	Forbidden	The current HipServ user is not the one whose gallery is requested.

Change HipServ User Password

Changes the password of a given HipServ user. Only this authenticated HipServ user is allowed to call this service.

Request

Method	PUT
URL	HIPSERV_URL/USER_URI/password
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/accounts/users/johndoe/password

Body	`<?xml version="1.0"?>`
	`<user password="PASSWORD" />`

Details	**USER_URI** is the URI of the HipServ user, received in the **userURI** attribute as a result of a user login. **PASSWORD** is the password which must be 6 or more characters long and include at least one digit.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot Parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
400.-10002	No Response From Central Server	
400.-10034	Invalid Password	
401	Unauthorized	The current visitor is not an authenticated HipServ user.
403	Forbidden	The current authenticated HipServ user is not the user whose information is requested.

Files Web Services

Get Directory Contents

Returns the directory contents for the specified directory. Subdirectories will not be traversed.

Request

Method	GET
URL	HIPSERV_URL/DIRECTORY_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM
Query String Arguments	■ offset (integer >= 0): skip that many children from the beginning of the result set ■ startHref: use the index of the asset with this href as the offset. startHref has precedence over offset. ■ length (integer > 0): max number of children to return ■ sort (name|mimeType|size|mtime|caption|tags): sort by that field ■ order (asc|desc): ascending (default) or descending order ■ folderfilter (true|false): if false, filters out folders ■ mediafilter (image|audio|video|document|all): only return files of that type ■ audiodetails (true|false): if true and mediafilter is audio, add title, album, artist, playtimeSeconds and savedPosition to the attributes of the children assets. Default is false ■ mobile (true|false): if true, the children file elements returned will not contain tags elements and the only attributes they will contain are name, type and href Note: if offset or length is not defined or invalid, all children are returned Note: if sort is not defined or invalid, children are ordered by ascending name Note: if folderfilter is not defined or invalid, folders are returned Note: if mediafilter is not defined or invalid, all media types are returned
Details	**DIRECTORY_URI** is the URI of a HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/directoryContents+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Get Light Directory Assets

Returns a lightweight (faster) version of the directory contents for the specified directory, excluding directories.

Request

Method	GET
URL	HIPSERV_URL/DIRECTORY_URI/lightassets
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM/lightassets
Details	**DIRECTORY_URI** is the URI of a HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/lightDirectoryAssets+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Get Directory Details

Returns details about a folder, without its children

Request

Method	GET

URL	HIPSERV_URL/DIRECTORY_URI/details
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM/details
Details	**DIRECTORY_URI** is the URI of a HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/directoryContents+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Get Directory Last Modified Date/Time

Returns the last modified date/time of a directory

Request

Method	GET
URL	HIPSERV_URL/DIRECTORY_URI/mtime
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM/mtime
Details	**DIRECTORY_URI** is the URI of a HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK

Body	`<?xml version="1.0"?>`
	`<file href="DIRECTORY_URI" mtime="`**`MTIME`**`" />`

Details	**MTIME** is the last modified time of the directory in seconds since the epoch.

Failure Responses

Code	Description	Cause

400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Get Asset Details

Returns details about the specified asset.

Request

Method	GET
URL	HIPSERV_URL/ASSET_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp
Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/assetDetails+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Get Asset or Directory Thumbnail

Returns the thumbnail image for the specified file (asset or directory).

Request

Method	GET
URL	HIPSERV_URL/FILE_URI/thumbnail

URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM /thumbnail
Details	**FILE_URI** is the URI of a HipServ asset or directory, i.e. the href of any file, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK
Details	The response is a JPEG, PNG or GIF image that can fit in a 115x115 square.

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset or directory

Get Asset or Directory Webview

Returns the webview image for the specified file (asset or directory).

Request

Method	GET
URL	HIPSERV_URL/FILE_URI/webview
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM /webview
Details	**FILE_URI** is the URI of a HipServ asset or directory, i.e. the href of any file, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	200 OK
Details	The response is a JPEG, PNG or GIF image that can fit in a 1024x768 rectangle.

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	

401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset or directory

Get Asset Source

Returns the source for the specified asset. Directories do not support this call.

Request

Method	GET
URL	HIPSERV_URL/ASSET_URI/src
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp/src
Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder. Streaming for files smaller than 2GB can be achieved by using the Range HTTP header.

Success Response

HTTP Status Code	200 OK
Details	The response can be of any MIME type.

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset or directory

Search for Files

Returns up to 500 files accessible to the current HipServ user (assets and directories) which caption, tags or filename match all the keywords specified in the query string. The files found through the caption are returned first, then tags then filename. Only a logged in HipServ User is allowed to search for files.

Request

Method	GET
URL	HIPSERV_URL/api/2.0/rest/files/search
Query String Arguments	**kw** - space-separated keyword list to search in tags, captions and filenames
Details	Keywords under 3 characters are ignored.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/searchResults+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
400.18	At least one of the keywords must have 3 characters or more	
401	Unauthorized	The current visitor is not an authenticated HipServ user

Create a directory

Creates a subdirectory in the specified directory and returns its directory contents.

Request

Method	POST
URL	HIPSERV_URL/DIRECTORY_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM

Body	`<?xml version="1.0">`
	`<file name="NAME" type="folder" />`

Details	**DIRECTORY_URI** is the URI of the parent HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information)

Success Response

HTTP Status Code	201 Created
MIME Type	hipserv/directoryContents+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	directory name must not start with a dot directory name must not match the Perl Regular Expression: /[<>&*\?\/\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	A file with the requested name already exists in the requested directory.
400.3	Cannot parse XML	
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Update Directory Name and/or Sharing With DMA

Updates the name and/or Sharing With DMA of a directory. Note: renaming a directory changes its URI and the URI of its assets.

Request

Method	PUT
URL	HIPSERV_URL/DIRECTORY_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM

Body	`<?xml version="1.0"?>`
	`<file sharingWithDMA="true|false"/>`
	`or`
	`<?xml version="1.0"?>`
	`<file name="NAME"/>`
	`or`
	`<?xml version="1.0"?>`
	`<file name="NAME" sharingWithDMA="true|false"/>`

Success Response	**DIRECTORY_URI** is the URI of the HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information).
HTTP Status Code **Details**	200 OK **NAME** is the new filename.
MIME Type	hipserv/directoryContents+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset Note that only the fields which change need to be part of the request and that **inherited** sharingWithDMA cannot changed.

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	directory name must not start with a dot directory name must not match the Perl Regular Expression: /[<>&*\?\/\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	A file with the requested name already exists in the requested directory.
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Update Asset Name and/or Caption

Updates the name and/or caption of an asset. Note: renaming an asset changes its URI.

Request

Method	PUT
URL	HIPSERV_URL/ASSET_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc

Body	`<?xml version="1.0"?>`
	`<file caption="CAPTION"/>`
	
	`or`
	
	`<?xml version="1.0"?>`
	`<file name="NAME"/>`
	
	`or`
	
	`<?xml version="1.0"?>`
	`<file name="NAME" caption="CAPTION"/>`

Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder. **CAPTION** is the new caption and **NAME** is the new filename.

	Note that only the fields which change need to be part of the request.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/assetDetails+xml

Failure Responses

Code	**Description**	**Cause**
400.1	Illegal Filename	asset name must not start with a dot asset name must not match the Perl Regular Expression: /[<>&* \?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	A file with the requested name already exists in the requested asset.
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Rename a Directory (alternate Web Service)

Renames a directory. Note: renaming a directory changes its URI and the URI of its assets.

Request

Method	PUT
URL	HIPSERV_URL/DIRECTORY_URI/name
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM/name

Body	`<?xml version="1.0"?>`
	`<file name="NAME"/>`

Details	**DIRECTORY_URI** is the URI of the HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information).

	NAME is the new filename.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/directoryContents+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	directory name must not start with a dot directory name must not match the Perl Regular Expression: /[<>&*\?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	A file with the requested name already exists in the requested directory.
400.3	Cannot parse XML	
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the directory

Rename an Asset (alternate Web Service)

Renames an asset. Note: renaming an asset changes its URI.

Request

Method	PUT
URL	HIPSERV_URL/ASSET_URI/name
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc/name

Body	`<?xml version="1.0"?>`
	`<file name="NAME"/>`

Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder.

	NAME is the new filename.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/assetDetails+xml

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	asset name must not start with a dot asset name must not match the Perl Regular Expression: /[<>&*\?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	A file with the requested name already exists in the requested asset.
400.3	Cannot parse XML	
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Save Asset Playback Position

Saves the asset playback position for the current visitor (not anonymous).

Request

Method	PUT
URL	HIPSERV_URL/ASSET_URI/savedPosition
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM/savedPosition

Body	`<?xml version="1.0"?>`
	`<file savedPosition="SECONDS"/>`

Details	**ASSET_URI** is the URI of an audio or video HipServ asset, i.e. the href of any file of type audio or video.

	SECONDS is the the playback position to save in seconds.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/assetDetails+xml

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
400.17	Saved position not supported for this media type	
400.29	Service is disabled	The requested asset is an audio asset and the current visitor is a HipServ user who set the ResumeAudioPlayback preference to False.
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset
403	Forbidden	Anonymous visitors cannot save playback positions

Reset Saved Asset Playback Position

Resets the saved asset playback position for the current visitor (not anonymous).

Request

Method	DELETE
URL	HIPSERV_URL/ASSET_URI/savedPosition
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM/savedPosition
Details	**ASSET_URI** is the URI of an audio or video HipServ asset, i.e. the href of any file of type audio or video.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
400.17	Saved position not supported for this media type	
400.29	Service is disabled	The requested asset is an audio asset and the current visitor is a HipServ user who set the ResumeAudioPlayback preference to False.
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset
403	Forbidden	Anonymous visitors cannot save playback positions

Rotate Image Asset

Rotates an image asset by specified number of degrees.

Request

Method	PUT
URL	HIPSERV_URL/ASSET_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc

Body	`<?xml version="1.0"?>`
	`<file rotation="DEGREES"/>`

Details	**ASSET_URI** is the URI of an audio or video HipServ asset, i.e. the href of any file of type audio or video. **DEGREES** is the clockwise rotation angle in degrees (90, 180 or 270).

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/assetDetails+xml

Failure Responses

Code	**Description**	**Cause**

400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Add Tags to an Asset

Adds one or more tags to an asset.

Request

Method	POST
URL	HIPSERV_URL/ASSET_URI/tags
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp/tags
MIME Type	hipserv/tagList+xml
Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder.

Success Response

HTTP Status Code	201 Created
MIME Type	hipserv/tagList+xml containing all tags attached to the asset

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Update all Tags in an Asset

Replaces the tags attached to an asset with new ones.

Request

Method	PUT
URL	HIPSERV_URL/ASSET_URI/tags
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp/tags
MIME Type	hipserv/tagList+xml
Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/tagList+xml containing all tags attached to the asset

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Delete all the Tags of an Asset

Deletes all the tags of an asset.

Request

Method	DELETE
URL	HIPSERV_URL/ASSET_URI/tags
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp/tags
Details	**ASSET_URI** is the URI of a HipServ asset, i.e. the href of any file of type other than folder.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the asset

Delete a Single File

Delete a singe file (assets and/or directories).

Request

Method	DELETE
URL	HIPSERV_URL/FILE_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM
Details	**FILE_URI** is the URI of the asset or directory to delete.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	Description	Cause
401	Unauthorized	The current visitor is not an authenticated HipServ who has write access to the file to delete

Delete Files

Deletes files (assets and/or directories).

Request

Method	POST
URL	HIPSERV_URL/api/2.0/rest/files/trash

Body	`<?xml version="1.0"?>`
	`<files>`
	`  <file href="URI1" type="TYPE1"/>`
	`  <file href="URI2" type="TYPE2"/>`

	`  ...`
	`</files>`

Details	**URI1**, **URI2** etc. are the URIs of the assets and directories to delete and **TYPE1**, **TYP2**, etc. are their types.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	one or more URIs are invalid
401	Unauthorized	The current visitor is not an authenticated HipServ

Copy Files

Copies files (assets and/or directories) to a target directory.

Request

Method	POST
URL	HIPSERV_URL/DIRECTORY_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM

Body	`<?xml version="1.0"?>`
	`<files>`
	`  <file href="URI1" type="TYPE1"/>`
	`  <file href="URI2" type="TYPE2"/>`
	`  ...`
	`</files>`

Details	**DIRECTORY_URI** is the URI of the target HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information) **URI1**, **URI2** etc. are the URIs of the assets and directories to copy and **TYPE1**, **TYP2**, etc. are their types.

Success Response

HTTP Status Code	201 Created

Body	a **file** root element containing the newly copied files : a hipserv/assetDetails+xml document for each copied asset and a hipserv/directoryContents+xml (without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset) document for each copied directory

Failure Responses

Code	Description	Cause
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	one or more URIs are invalid
400.23	Cannot copy/move to self or to a descendant	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the target directory

Move Files

Moves files (assets and/or directories) to a target directory.

Request

Method	POST
URL	HIPSERV_URL/DIRECTORY_URI/movelist
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM/movelist

Body	`<?xml version="1.0"?>`
	`<files>`
	`  <file href="URI1" type="TYPE1"/>`
	`  <file href="URI2" type="TYPE2"/>`
	`  ...`
	`</files>`

Details	**DIRECTORY_URI** is the URI of the target HipServ directory, i.e. the href of any file of type folder, or the href of a media source (see HipServ User Information) **URI1**, **URI2** etc. are the URIs of the assets and directories to move and **TYPE1**, **TYP2**, etc. are their types.

Success Response

HTTP Status Code	201 Created

Body	a **file** root element containing the newly moved files : a hipserv/assetDetails+xml document for each moved asset and a hipserv/directoryContents+xml (without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset) document for each moved directory

Failure Responses

Code	Description	Cause
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	one or more URIs are invalid
400.23	Cannot copy/move to self or to a descendant	
401	Unauthorized	The current visitor is not an authenticated HipServ user who has access to the target directory

Contacts Web Services

Get all the contacts of a user

Returns all the contacts of a user (who must be logged in).

Request

Method	GET
URL	HIPSERV_URL/CONTACTS_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/contacts/johndoe
Details	**CONTACTS_URI** is available in the **contactsURI** attribute of the "Get HipServ User Information" Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/contacts+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user
404	Not Found	The current visitor is not the authenticated HipServ user whose contacts are requested

Get a Contact

Returns a given contact of a user (who must be logged in)

Request

Method	GET
URL	HIPSERV_URL/CONTACT_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/contacts/johndoe/jenny%40example.com
Details	**CONTACT_URI** is the URI of a contact, they are available in the **href** attributes of the "Get all the contacts of a user" Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/contact+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an authenticated HipServ user
404	Not Found	The current visitor is not the authenticated HipServ user whose contact is requested

Create a contact

Adds a contact to the contact list of a HipServ user

Request

Method	POST
URL	HIPSERV_URL/CONTACTS_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/contacts/johndoe
MIME Type	hipserv/contact+xml without the **href** attribute
Details	**ALBUM_LISTING_URI** is available in the **albumsURI** attribute of the "Get HipServ User Information" Web Service.

Success Response

HTTP Status Code	201 Created
MIME Type	hipserv/contact+xml

Failure Responses

<table>
<tr><th>Code</th><th>Description</th><th>Cause</th></tr>
<tr><td>400.3</td><td>Cannot parse XML</td><td></td></tr>
<tr><td>400.4</td><td>Input/Output Error</td><td></td></tr>
<tr><td>400.6</td><td>Invalid Email</td><td>The email address must match the following Perl regular expression (ignore line breaks):
/^
(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))</td></tr>
</table>

<table>
<tr><td></td><td></td><td>@
((\[(2([0-4]\d|5[0-5])|1?\d{1,2})(\.(2([0-4]\d|5[0-5])|1?\d{1,2})){3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))
$/</td></tr>
<tr><td>400.10</td><td>Duplicate email</td><td>There cannot be two contacts with the same email address in the contact list of a HipServ user</td></tr>
<tr><td>401</td><td>Unauthorized</td><td>The current visitor is not an authenticated HipServ user</td></tr>
<tr><td>404</td><td>Not Found</td><td>The contact list to which the contact is added is not the one of the current visitor</td></tr>
</table>

Edit a contact

Edits a contact. Changing its email address changes its URI.

Request

Method	PUT
URL	HIPSERV_URL/CONTACT_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/contacts/johndoe/jenny%40example.com
MIME Type	hipserv/contact+xml without href attribute. The **email** and/or **DisplayName** attribute must not be empty.
Details	**CONTACT_URI** is the URI of a contact, they are available in the **href** attributes of the "Get all the contacts of a user" Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/contact+xml
Details	the response status code might be 201 although it should be 200.

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	

<table>
<tr><td>400.6</td><td>Invalid Email</td><td>The email address must match the following Perl regular expression (ignore line breaks):
/^
(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))
@
((\[(2([0-4]\d|5[0-5])|1?\d{1,2})(\.(2([0-4]\d|5[0-5])|1?\d{1,2})){3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))
$/</td></tr>
<tr><td>400.49</td><td>Edit contact - User already owns a contact with that e-mail address</td><td>There cannot be two contacts with the same email address in the contact list of a HipServ user</td></tr>
<tr><td>401</td><td>Unauthorized</td><td>The current visitor is not an authenticated HipServ user</td></tr>
<tr><td>404</td><td>Not Found</td><td>The current visitor is not the authenticated HipServ user whose contact is requested</td></tr>
</table>

Delete a Contact

Deletes a contact owned by a HipServ user (who must be logged in).

Request

Method	DELETE
URL	HIPSERV_URL/CONTACT_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/contacts/johndoe/jenny%40example.com
Query String Arguments	**force=true** - remove the contact even if one or more albums owned by the authenticated HipServ user are shared with the contact to remove
Details	**CONTACT_URI** is the URI of a contact, they are available in the **href** attributes of the "Get all the contacts of a user" Web Service.

Success Response

HTTP Status Code	204 No Content
MIME Type	hipserv/contact+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	
400.12	Contact is used in albums	One or more albums owned by the authenticated HipServ user is shared with the contact to remove and '*force=true* is not in the query strings.
401	Unauthorized	The current visitor is not an authenticated HipServ user
404	Not Found	The current visitor is not the authenticated HipServ user whose contact is requested

Themes Web Services

Get All Themes

Returns all themes ordered by rank - only available to authenticated HipServ users.

Request

Method	GET
URL	HIPSERV_URL/THEMES_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/themes
Details	**THEMES_URI** is available in the **themesURI** attribute of the "Get HipServ User Information" Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/themes+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
404	Not Found	The current visitor is not an authenticated HipServ user

Get a Theme

Returns a given theme

Request

Method	GET
URL	HIPSERV_URL/THEME_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/themes/hipserv/travel
Details	**THEME_URI** is the value of a **href** attribute of the response of the "Get All Themes" Web Service.

Success Response

HTTP Status Code	200 OK

MIME Type	hipserv/theme+xml

Failure Responses

Code	Description	Cause
400.4	Input/Output Error	

Albums Web Services

Get Listing of Albums owned by a user

Returns the listing of all albums owned by a user (who must be logged in)

Request

Method	GET
URL	HIPSERV_URL/ALBUM_LISTING_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users/am9obmRvZQ
Details	**ALBUM_LISTING_URI** is available in the **albumsURI** attribute of the "Get HipServ User Information" Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/albumListing+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album listing

Get Album Contents

Returns information about the requested album and its contents.

Request

Method	GET
URL	HIPSERV_URL/ALBUM_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files/am9obmRvZS9TYW1wbGUgQWxidW0
Query String Arguments	▪ offset (integer >= 0): skip that many children from the beginning of the result set ▪ startHref: use the index of the asset with this href as the offset. startHref has

	precedence over offset. ■ length (integer > 0): max number of children to return ■ sort (name|mimeType|size|mtime|caption|tags): sort by that field ■ order (asc|desc): ascending (default) or descending order ■ folderfilter (true|false): if false, filters out folders ■ mediafilter (image|audio|video|document|all): only return files of that type ■ audiodetails (true|false): if true and mediafilter is audio, add title, album, artist, playtimeSeconds and savedPosition to the attributes of the children assets. Default is false ■ mobile (true|false): if true, the children file elements returned will not contain tags elements and the only attributes they will contain are name, type and href Note: if offset or length is not defined or invalid, all children are returned Note: if sort is not defined or invalid, children are ordered by ascending name Note: if folderfilter is not defined or invalid, folders are returned Note: if mediafilter is not defined or invalid, all media types are returned
Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/albumContents+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
403	Forbidden	The current visitor does not have read access to the requested album

Get Light Album Assets

Returns a lightweight (faster) version of the album contents for the specified album

Request

Method	GET
URL	HIPSERV_URL/ALBUM_URI/lightassets

URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /am9obmRvZS9TYW1wbGUgQWxidW0/lightassets
Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/lightAlbumAssets+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
403	Forbidden	The current visitor does not have read access to the requested album

Get Album Details

Returns details about an album, without its children

Request

Method	GET
URL	HIPSERV_URL/ALBUM_URI/details
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/files /am9obmRvZS9TYW1wbGUgQWxidW0/details
Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/albumContents+xml without children elements, mediafilter, folderfilder, nbFilteredAssets or assetsOffset

Failure Responses

Code	**Description**	**Cause**

400.4	Input/Output Error	
403	Forbidden	The current visitor does not have read access to the requested album

Get Album Asset Details/Thumbnail/Webview/Source

Same as Asset Details/Thumbnail/Webview/Source but with album assets URIs and error 403 (Forbidden) can be returned if access to the asset is denied to the current visitor.

Note: the path attribute returned by an Album Asset Details Web Service call will be the path of the underlying Asset, and will be empty if the current visitor is not allowed to access this asset via Samba.

Save Album Asset Playback Position / Reset Saved Album Asset Playback Position

Same as Save Asset Playback Position / Reset Saved Asset Playback Position but with album asset URIs, for the current HipServ user or guest (with access to the asset), but **not** for anonymous visitors. Error 403 (Forbidden) is returned if the visitor does not have access to the asset and for anonymous visitors.

Create an Album

Creates an album in the specified album listing and returns the new album. **Note:** when sharing with friends, contacts must already exist

Request

Method	POST
URL	HIPSERV_URL/ALBUM_LISTING_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users/am9obmRvZQ
MIME Type	hipserv/albumRequest+xml
Details	**ALBUM_LISTING_URI** is available in the **albumsURI** attribute of the "Get HipServ User Information" Web Service.

Success Response

HTTP Status Code	201 Created
MIME Type	hipserv/albumContents+xml without children, mediafilter, folderfilder, nbFilteredAssets

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	album name must not start with a dot album name must not match the Perl Regular Expression: /[<>&*\?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	The HipServ user already owns an album with the same name
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.11	Contact does not exist	When creating/editing an album with contacts, the contacts must already exist
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album listing

Update an Album

Updates the properties of a specified album.

Note: when sharing with friends, contacts must already exist

Note: renaming an album changes its URI and the URI of its assets

Request

Method	PUT
URL	HIPSERV_URL/ALBUM_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users /am9obmRvZS9TYW1wbGUgQWxidW0
MIME Type	hipserv/albumRequest+xml
Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/albumContents+xml without children, mediafilter, folderfilder, nbFilteredAssets

Failure Responses

Code	Description	Cause
400.1	Illegal Filename	album name must not start with a dot album name must not match the Perl Regular Expression: /[<>&*\?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	The HipServ user already owns another album with the same name
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.11	Contact does not exist	When creating/editing an album with contacts, the contacts must already exist
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album

Rename an Album (alternate Web Service)

Renames an album. This is an alternate Web Service to perform a rename action only. **Note:** renaming an album changes its URI and the URI of its assets

Request

Method	PUT
URL	HIPSERV_URL/ALBUM_URI/name
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0/name

Body	`<?xml version="1.0"?>`
	`<file name="NAME" />`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service. **NAME** is the new album name.

Success Response

HTTP Status Code	200 OK

MIME Type	hipserv/albumContents+xml without children, mediafilter, folderfilder, nbFilteredAssets

Failure Responses

Code	**Description**	**Cause**
400.1	Illegal Filename	album name must not start with a dot album name must not match the Perl Regular Expression: /[<>&*\?\/\\\\@#\$%\^":]/ i.e must not contain any of the following characters < > & * ? / \ @ # $ % ^ " :
400.2	File Already Exists	The HipServ user already owns another album with the same name
400.3	Cannot parse XML	
400.4	Input/Output Error	
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album

Send a Share Email to all Album Contacts

Sends a Share Email to all Album Contacts.

Request

Method	POST
URL	HIPSERV_URL/ALBUM_URI/email
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0/email

Body	`<?xml version="1.0"?>`
	`<email />`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	**Description**	**Cause**

400.3	Cannot parse XML	
400.4	Input/Output Error	
400.7	share_from_email setting is undefined	The **share_from_email** user preference must be defined to send emails to contacts
400.8	Error sending email	
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album

Add Assets to an Album

Adds the assets listed in the request to a given album, as well as the assets at the root of the directories listed in the request. Subdirectories are not traversed.

Request

Method	POST
URL	HIPSERV_URL/ALBUM_URI
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users /am9obmRvZS9TYW1wbGUgQWxidW0

Body	`<?xml version="1.0"?>`
	`<files>`
	`  <file href="URI1" />`
	`  <file href="URI2" />`
	`  ...`
	`</files>`
	
	`or`
	
	`<?xml version="1.0"?>`
	`<file href="URI1" />`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service. **URI1**, **URI2** etc. are the URIs of the assets and directories to add to the album.

Success Response

HTTP Status Code	201 Created
MIME Type	hipserv/albumContents+xml without children, mediafilter, folderfilder, nbFilteredAssets

Failure Responses

Code	Description	Cause
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	one or more URIs are invalid
400.19	User is not authorized/logged on one or more 3rd party services to which the requested album is linked	The user is not logged on one or more 3rd party services to which the requested album is linked and/or has not allowed the HipServ application to access one ore more of those 3rd party services. The error element contains the following attributes: ▪ **service**: the 3rd party service to log on to ▪ **loginURL**: the 3rd party service login/access URL ▪ **userDisplayName** (optional): display name to log on as on the 3rd party service - can happen if the current user is logged on as another user than the one who linked the album to the 3rd party service The user must be directed to the **loginURL** and allow the HipServ application to access the 3rd party service. Once it's done, he can retry to call this Web Service. Alternatively the album can be unlinked from the 3rd party service(s)
400.20	Misc 3rd Party service error	
400.21	3rd party photoset does not exit	The 3rd party service photoset to which the album is linked does not exist. It should be unlinked.
401	Unauthorized	The current visitor is not the authenticated owner of the requested album

Update an Album Cover Asset

Updates the Cover Asset of an Album

Request

Method	PUT
URL	HIPSERV_URL/ALBUM_URI/cover
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users /am9obmRvZS9TYW1wbGUgQWxidW0/cover

Body	`<?xml version="1.0"?>`
	`<file href="ASSET_URI" />`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service. **ASSET_URI** is the URI of the album asset to use as a cover asset

Success Response

HTTP Status Code	200 OK
Body	`same as the request body`

Failure Responses

Code	**Description**	**Cause**
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	**ASSET_URI** not a valid album asset URI
401	Unauthorized	The current visitor is not an the authenticated HipServ user who owns the requested album

Delete Album or Remove an Asset from an Album

Same as Delete Directory (but with Album URIs) and Delete Asset (but with Album Asset URIs).

Remove Multiple Assets from an Album

Removes listed assets from an album.

Request

Method	POST
URL	HIPSERV_URL/ALBUM_URI/trash
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users /am9obmRvZS9TYW1wbGUgQWxidW0/trash

Body	`<?xml version="1.0"?>`
	`<files>`
	`  <file href="URI1" />`
	`  <file href="URI2" />`
	`  ...`
	`</files>`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service.

	URI1, **URI2** etc. are the URIs of the assets to remove from the album.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	Description	Cause
400.3	Cannot parse XML	
400.4	Input/Output Error	
400.5	Invalid Parameter	one or more URIs are invalid
401	Unauthorized	The current visitor is not the authenticated owner of the requested album

3rd Party Web Services

Supported 3rd party services are:

- `facebook` (Facebook)
- `flickr` (Flickr)

Get Listing of Photosets owned by the current user on a 3rd party service

Returns the listing of all photosets owned by the current user on a given 3rd party service

Request

Method	GET
URL	HIPSERV_URL/api/2.0/rest/3rdparty/SERVICE/photosets
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/3rdparty/facebook/photosets
Details	**SERVICE** is the name of a supported 3rd party service.

Success Response

HTTP Status Code	200 OK
MIME Type	hipserv/3rdPartyPhotosetListing+xml

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
400.19	User is not authorized/logged into the requested 3rd party service.	The user is not logged on the 3rd party service and/or has not allowed the HipServ application to access the 3rd party service. The error element contains the following attributes: ■ **service**: the requested 3rd party service ■ **loginURL**: the 3rd party login/access URL The user must be directed to the **loginURL** and allow the HipServ application to access the 3rd party service. Once it's done, he can retry to call this Web Service.
400.20	Misc 3rd Party service error	
400.29	Service is disabled	The requested 3rd party service is not enabled on the HipServ
401	Unauthorized	The current visitor is not an authenticated HipServ user

Link an album to a 3rd party photoset

Link an album to a flickr/facebook photoset and upload all current photos to the 3rd party service. Once linked, images added to the album will be uploaded to the 3rd party service.

Request

Method	PUT
URL	HIPSERV_URL/ALBUM_URI/3rdparty/SERVICE/photoset
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users /YWRtaW4vQWRtaW4gU2FtcGxlIEFsYnVt/3rdparty/photoset

Body	`<?xml version="1.0"?>`
	`<photoset name="NAME" caption="CAPTION"/>`
	
	`or`
	
	`<?xml version="1.0"?>`
	`<photoset id="ID"/>`

Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service. **SERVICE** is the name of a supported 3rd party service. Provide **NAME** and **CAPTION** (optional) to link to a new photoset. Provide **ID** to link to an existing photoset.

Success Response

HTTP Status Code	200 OK
Body	`same as the request body`

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
400.3	Cannot parse XML	
400.19	User is not authorized/logged into the requested 3rd party service.	The user is not logged on the 3rd party service and/or has not allowed the HipServ application to access the 3rd party service. The error element contains the following attributes: ■ **service**: the requested 3rd party service ■ **loginURL**: the 3rd party login/access URL The user must be directed to the **loginURL** and allow the HipServ application to access the 3rd party service. Once it's done, he can

		retry to call this Web Service.
400.20	Misc 3rd Party service error	
400.29	Service is disabled	The requested 3rd party service is not enabled on the HipServ
401	Unauthorized	The current visitor is not the authenticated owner of the requested album

Unlink an album from a 3rd party photoset

Unlinks an album from a flickr/facebook photoset.

Request

Method	DELETE
URL	HIPSERV_URL/ALBUM_URI/3rdparty/SERVICE/photoset
URL Example	http://johndoe.myhipserv.com/api/2.0/rest/albums/users/YWRtaW4vQWRtaW4gU2FtcGxlIEFsYnVt/3rdparty/photoset
Details	**ALBUM_URI** is the URI of a HipServ album which can be obtained via the album listing or gallery Web Service. **SERVICE** is the name of a supported 3rd party service.

Success Response

HTTP Status Code	204 No Content

Failure Responses

Code	**Description**	**Cause**
400.4	Input/Output Error	
400.29	Service is disabled	The requested 3rd party service is not enabled on the HipServ
401	Unauthorized	The current visitor is not the authenticated owner of the requested album

Non-REST Web Services API

File Upload Services

Upload Files

Standard HTTP upload can be used to upload files to a directory accessible to a HipServ user.

To do so, send a POST request to /filemanager/done_upload on the HipServ with the following parameters in multipart/form-data encryption:

- **np**: target directory URI. Example: /api/2.0/rest/files/ZmFtaWx5
- **ACTION**: `UPLOAD_FILE`
- **file[0]**: first file to upload
- **file[1]**: second file to upload
- etc.

The response will unfortunately always be an HTML document sent with status code 200. The best way to verify that the upload was successful is to request a directory listing.

Sample Request Body:

```
Content-Type: multipart/form-data; boundary=---------------------------3164130335698
Content-Length: 533

-----------------------------3164130335698
Content-Disposition: form-data; name="ACTION"

UPLOAD_FILE
-----------------------------3164130335698
Content-Disposition: form-data; name="np"

/api/2.0/rest/files/ZmFtaWx5
-----------------------------3164130335698
Content-Disposition: form-data; name="file[0]"; filename="foo"
Content-Type: application/octet-stream

bar
-----------------------------3164130335698--
```

Appendix

HipServ XML MIME Types

hipserv/session+xml

A hipserv/session+xml document describes a HipServ user session and contains enough information for the client to get started.

It is made of a single **session** root element with the following attributes:

- **supportURL** - URL of the support page
- **userURI** - URI where to gather information about the current HipServ user
- **serverConfigURI** - URI where to gather information about the server configuration
- **href** - URI of the session; requesting a DELETE method on this URI will log the user out
- **locale** - current locale

```
<?xml version="1.0"?>
<session
    supportURL="http://www.hipserv.com/support/"
    userURI="/api/2.0/rest/accounts/users/johndoe"
    serverConfigURI="/api/2.0/rest/server/config"
    href="/api/2.0/rest/sessions/12345678123456781234567812345678"
    locale="en_US"
/>
```

hipserv/serverconfig+xml

The **hipserv/serverconfig+xml** MIME type defines an XML data structure providing access to the HipServ configuration.

The **configuration** root node contains the following attributes:

- **version** - HipServ version
- **release** - HipServ release
- **key** - Product key issued to the HipServ (only for admin users)
- **portal** - Registered portal name for the HipServ (aka HipName or HipServ Name)
- **apacheURL** - Apache URL (for open source credits)
- **phpURL** - PHP URL (for open source credits)
- **supportURL** - Product support site URL
- **userManualURL** - Product user manual URL
- **mediaPlaybackHelpURL** - Absolute Path to the media player help "?" button on the HipServ
- **eulaURL** - Absolute Path to the End User License Agreement (EULA) on the HipServ
- **loginURL** - Absolute Path to the End User login portal on the HipServ
- **serverNoticesURI** - Absolute Path to the Web Service endpoint to get Server Notices on the HipServ
- **downloadURL** - Absolute Path to the download HTML popup page on the HipServ
- **uploadURL** - Absolute Path to the upload HTML popup page on the HipServ
- **brandingLogoURL** - Absolute Path to the branding logo on the HipServ
- **productName** - branded name for the HipServ

- **quickTourVideoURL** - Flash video URL providing 'QuickTour' product intro (optional)
- **brandingLogoLink** - URL of the link to put on the branding logo (optional)
- **defaultHipName** - the default HipName in LAN Only mode (optional)
- **subscription** - standard|premium (only for admin users when the concept of subscription exists)
- **isRemote** - true|false ; false if the client is in the same local network as the HipServ (only for admin users when the concept of subscription exists)
- **hasNoPrivateFiles** - 'true' if MyComputers and MyLibrary do not exist, missing otherwise
- **hasNoFamilyLibrary** - 'true' if FamilyLibrary does not exist, missing otherwise
- **isOnlyUSBStorage** - 'true' if no hard drive is permanent (i.e. the HipServ is a "plug"), missing otherwise
- **supportsNoCS** - 'true' if the HipServ supports switching between LAN Only and Remote Access
- **noCS** - 'true' if the CPE is "LAN Only" i.e. no remote access i.e. does not communicate with the CS

See below a sample implementation.

```
<?xml version="1.0"?>
<configuration
  version="2.3.2"
  release="2.0.0.261"
  portal="johndoe"
  loginURL="/homebase/signin?i=1234567"
  serverNoticesURI="/api/2.0/rest/server/notices"
  userManualURL="http://support.hipserv.com/help"
  apacheURL="http://www.apache.org"
  phpURL="http://www.php.net"
  supportURL="http://support.hipserv.com/support"
  mediaPlaybackHelpURL="http://support.hipserv.com/support/documentation/codecs"
  eulaURL="/eula.html"
  uploadURL="/filemanager/upload_file"
  downloadURL="/filemanager/download_files"
  brandingLogoURL="/images/Co-Brand-Logos-HipServ.png"
  productName="HipServ"
  brandingLogoLink="http://www.axentra.com"
  key="ABCD-EFGH-IJKL-MNOP"
  subscription="premium"
  isRemote="false" />
```

hipserv/user+xml

The **hipserv/user+xml** MIME type defines an XML data structure providing access to a HipServ user / guest account.

The **user** root node contains the following attributes:

- **username** - Login username (for hipserv user) or email (for guest) (empty for an anonymous visitor)
- **href** - User URI
- **displayName** - user display name
- **isAdmin** - administrator account flag - Boolean flag indicating if the user is an 'Administrator' on the system (true|false)
- **isGuest** - guest account flag - Boolean flag indicating if the user is a 'guest' on the system (true|false)
- **isHipServUser** - HipServ account flag - Boolean flag indicating if the user has a HipServ account (true|false)
- **isFamilyMember** - Family member flag - Boolean flag indicating if the HipServ account has access to the family library (true|false)

- **albumsURI** - user album listing URI (hipserv user only)
- **contactsURI** - user contacts URI (hipserv user only)
- **themesURI** - themes listing URI (hipserv user only)
- **SSOLoginID** - SSO login ID (only if the user logged in via SSO or if the SSO binding was created in the current session)

The **user** root node contains also:

- a single **mediaSources** element hipserv/mediaSources+xml (HipServ user only)
- a single **preferences** element hipserv/preferences+xml (HipServ user only)

See below an example implementation of the **hipserv/user+xml** MIME type:

```
<?xml version="1.0"?>
<user href="/api/2.0/rest/accounts/users/johndoe" username="johndoe"
      displayName="John Doe" isAdmin="true" isGuest="false" isHipServUser="true"
      isFamilyMember="true" albumsURI="/api/2.0/rest/albums/users/am9obmRvZQ"
      contactsURI="/api/2.0/rest/contacts/johndoe" themesURI="/api/2.0/rest/themes">

  <mediaSources>
    <mediaSource id="FamilyLibrary" name="FamilyLibrary"
                 href="/api/2.0/rest/files/ZmFtaWx5"
                 backupHref="/api/2.0/rest/backup/files/ZmFtaWx5"/>
    <mediaSource id="MyLibrary" name="MyLibrary"
                 href="/api/2.0/rest/files/users/johndoe/TXlMaWJyYXJ5"
                 backupHref="/api/2.0/rest/backup/files/users/johndoe/TXlMaWJyYXJ5"/>
    <mediaSource id="MyComputers" name="MyComputers"
                 href="/api/2.0/rest/files/users/johndoe/TXlDb21wdXRlcnM"
                 backupHref="/api/2.0/rest/backup/files/users/johndoe/TXlDb21wdXRlcnM"/>
  </mediaSources>

  <preferences>

    <preference name="locale" desc="Language" value="en_US" weight="1" type="list"
                category="User" categoryID="User" required="True" permission="user"
                default="en_US" validation="" relatedRule="" reloadRequired="False"
                validationMessage="Invalid" editable="True" visible="True" sort=""
                sortOn="" handler="user">
      <value name="en_US" desc="en_US"/>
      <value name="fr_FR" desc="fr_FR"/>
      <value name="es_ES" desc="es_ES"/>
      <value name="nl_NL" desc="nl_NL"/>
      <value name="de_DE" desc="de_DE"/>
      <value name="it_IT" desc="it_IT"/>
    </preference>

    <preference name="share_from_email" desc="Sharing Emails Sender"
                value="john@example.com" weight="2" type="string" category="User"
                categoryID="User" required="False" permission="user" default=""
                validation="^([\w-]+(?:\.[\w-]+)*)@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)
                relatedRule="" reloadRequired="False"
                validationMessage="invalidFormatEmailAddress" editable="True"
                visible="True" sort="" sortOn="" handler="user"/>

  </preferences>

</user>
```

hipserv/mediaSources+xml

The **hipserv/mediaSources+xml** MIME type defines an XML data structure which describes the media sources available to a HipServ user.

The **mediaSources** root element contains:

- 0 or more **mediaSource** elements containing the following attributes:
  - **id** - media source id: MyLibrary|MyComputers|FamilyLibrary|ExternalStorage
  - **name** - media source name (filename on the HipServ)
  - **href** - media source URI (i.e. directory URI)
  - **backupHref** - media source backup URI (only if either USB or Online backup is enabled)

See below an example implementation of the **hipserv/mediaSources+xml** MIME type:

```
<?xml version="1.0"?>
<mediaSources>
  <mediaSource id="FamilyLibrary" name="FamilyLibrary"
               href="/api/2.0/rest/files/ZmFtaWx5"
               backupHref="/api/2.0/rest/backup/files/ZmFtaWx5"/>
  <mediaSource id="MyLibrary" name="MyLibrary"
               href="/api/2.0/rest/files/users/johndoe/TXlMaWJyYXJ5"
               backupHref="/api/2.0/rest/backup/files/users/johndoe/TXlMaWJyYXJ5"/>
  <mediaSource id="MyComputers" name="MyComputers"
               href="/api/2.0/rest/files/users/johndoe/TXlDb21wdXRlcnM"
               backupHref="/api/2.0/rest/backup/files/users/johndoe/TXlDb21wdXRlcnM"/>
</mediaSources>
```

hipserv/preferences+xml

A **hipserv/preferences+xml** defines a list of preferences the current visitor has access to.

Its **preferences** root element contains 0 or more **preference** elements (one per preference) containing the following attributes:

- **name** - preference name/id
- **desc** - description of the preference in plain English (for information only)
- **value** - preference value if defined, default value otherwise
- **weight** - integer used to sort preferences of the same category (lower numbers go first)
- **type** - preference type; they are:
  - **label** - just a label, not a real preference
  - **bool** - boolean : `True` or `False`
  - **string** - single line string
  - **list** - the supported value of the preference are listed in **value** elements (see below)
  - **bigString** - multiline string
  - **date** - date (yyyy-mm-dd)
  - **time** - time in military time (hh:mm)
  - **relatedListParent** - yet to document
  - **relatedListChild** - yet to document
  - **button** - button, to "click" it, set the value of the preference to `buttonClicked`
- **category** - preference category in plain English (for information only)
- **categoryID** - preference category id; preferences should be grouped by category for edition. User preferences are all in the `User` category.
- **required** - if true or True, the preference value cannot be an empty string. This attribute can also contain a condition such as `dhcp=False` which mean the preference is required if and only if the value of the dhcp

preference is False.

- **permission** - who is allowed to access the preference: `user`: HipServ users, `admin`: admin users, `setup`: local visitors during setup , empty string: anybody
- **default** - default value
- **validation** - validation criterion, can be safely ignored for non-string/bigString preferences. Can be either:
  - #csv#REGEX - the value must be split on commas and each piece must match the REGEX regular expression
  - REGEX - the value must match the REGEX regular expression
- **relatedRule** - condition to meet for the preference to be displayed; for example `dhcp=True` means "show this preference if and only if the value of the dhcp preference is True".
- **reloadRequired** - if true or True, the user should re-log in once the preference is saved
- **validationMessage** - legacy, not used
- **editable** - legacy, not used
- **visible** - if False or false, the preference should not be displayed
- **sort** - if True or true, the supported values for the preference should be sorted in the UI
- **sortOn** - if set to `name`, sort the supported values by localized name
- **handler** - `user` for user preferences, `server` for server preferences

preference elements of **type** `string`, `bigString` or `relatedListChild` also contain 1 or more **value** elements with (one per supported value) with the following attributes:

- name: the supported preference value
- desc: a description of the value (for information only)

Sample:

```
<preferences>

  <preference name="share_from_email" desc="Sharing Emails Sender" value="john@example.com"
              weight="2" type="string" category="User" categoryID="User" required="False"
              permission="user" default=""
              validation="^([\w-]+(?:\.[\w-]+)*)@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$"
              relatedRule="" reloadRequired="False" validationMessage="invalidFormatEmailAddress"
              editable="True" visible="True" sort="" sortOn="" handler="user"/>

  <preference name="landing_page_bg_position" desc="Landing Page Background Position" value="center"
              weight="4" type="list" category="User" categoryID="User" required="True" permission="user"
              default="center" validation="" relatedRule="" reloadRequired="False"
              validationMessage="Invalid" editable="True" visible="True" sort="" sortOn=""
              handler="user">
    <value name="center" desc="backgroundposition.center"/>
    <value name="fit" desc="backgroundposition.fit"/>
  </preference>

  <preference name="ResumeAudioPlayback" desc="" value="False" weight="10" type="bool" category="User"
              categoryID="User" required="false" permission="user" default="False" validation=""
              relatedRule="" reloadRequired="False" validationMessage="Invalid" editable="True"
              visible="true" sort="" sortOn="" handler="user">
    <value name="True" desc="True"/>
    <value name="False" desc="False"/>
  </preference>

  <preference name="do_not_prompt_clear" desc="Reset all prompts" value="null" weight="12"
              type="button" category="User" categoryID="User" required="False" permission="user"
              default="" validation="" relatedRule="" reloadRequired="False"
              validationMessage="Invalid" editable="True" visible="True" sort="" sortOn=""
              handler="user"/>

<preferences>
```

User preferences applicable to a mobile UI

Name	Description	Values	Default Value
locale	user locale	branding dependent, usually de_DE, en_US, es_ES, fr_FR, it_IT and nl_NL	branding dependent, usually en_US
file_exists_action	Action to perform when moving/copying /uploading a file to a directory containing a file with the same name	■ `new`: keep the new file, overwrite the old one ■ `rename-new`: rename the new file to avoid overwriting the old one ■ `old`: keep the old file, do not overwrite it	`rename-new`
landing_page_bg_clear	button to clear the landing page image	`buttonClicked`	n/a
landing_page_bg_position	landing page image scaling method	■ `center`: scale the image so that it fills the screen as much as possible without cropping it, maintaining the image ratio ■ `fit`: scale the image so that it fills the whole screen, cropping it if necessary, maintaining the image ratio	`center`
landing_page_bg_url	Landing page background image URI	The URI of an image asset accessible by the user	(empty string)
new_album_sharing_dma	Share new albums with media players by default	`True` or `False`	`False`
new_album_sharing_family	Share new albums with other family accounts by default	`True` or `False`	`False>`
ResumeAudioPlayback	Resume Audio Playback where it stopped? i.e. allow visitors to save the position of the playback of audio files	`True` or `False`	`False`
share_from_email	Sharing Emails Sender	a valid email address	(empty string)

hipserv/gallery+xml

A **hipserv/gallery+xml** document is an XML document describing the albums available (read access) to a visitor (HipServ user, guest or anonymous visitor).

Its **gallery** root element contains 0 or more **file** elements with the following attributes:

- **name** - album name
- **type** - "folder"
- **href** - album URI
- **ownerFullName** - display name of the album owner
- **nbFiles** - approximate number of files in the album
- **coverURI** - URI of the asset to use for the album cover, from the DB if defined, otherwise the first image, video, music or document asset in the album (in this order) (missing if the album is empty)
- **canDownload** - true|false: can gallery visitors download files from this albums?
- **theme** - URI of the theme of the album
- **themeId** - ID of the theme of the album

Sample:

```
<?xml version="1.0"?>
<gallery>
  <file name="Admin Sample Album" type="folder"
        href="/api/2.0/rest/albums/users/YWRtaW4vQWRtaW4gU2FtcGxlIEFsYnVt" ownerFullName="Admin"
        coverURI="/api/2.0/rest/albums/users/YWRtaW4vQWRtaW4gU2FtcGxlIEFsYnVtL3BhcmlzMDEuanBn"
        canDownload="true" theme="/api/2.0/rest/themes/hipserv/birthday"
        themeId="hipservBirthday" nbFiles="4"/>
  <file name="Sample Album" type="folder"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0" ownerFullName="John Doe"
        coverURI="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vcGFyaXMwMS5qcGc"
        canDownload="true" theme="/api/2.0/rest/themes/hipserv/travel"
        themeId="hipservTravel" nbFiles="4"/>
</gallery>
```

hipserv/directoryContents+xml

The hipserv/directoryContents+xml MIME type defines an XML data structure providing access to the contents of a HipServ directory. All assets and subdirectories in the directory are provided, subdirectories are not traversed.

The **file** root node containts the following attributes:

- **name** - folder name
- **href** - folder URI
- **parentHref** - URI of the parent folder (or empty if the folder is a media source)
- **path** - the Samba path of the folder
- **type** - always `folder`
- **mtime** - last modification time in seconds since the epoch
- **nbFolders** - the number of folders in the folder
- **nbFiles** - the number of files in the folder

- **sharingWithDMA** - `true` if the folder is shared with DMA, `inherited` if the folder is shared with DMA because one of its ancestors is, `false` otherwise (optional, default `false`)
- **deviceHref** - URI of the ejectable USB storage device where the folder resides (optional)
- **readOnly** - (true|false), true for folders that cannot be deleted, copied, renamed or moved, and where no file or folder can be created/added/copied/moved (so far, only ExternalStorage) (optional, default `false`)
- **isCoreMount** - (true|false), true for folders that are core mountpoints, i.e. can only be ejected by admins (so far, MyLibrary, FamilyLibrary, MyComputers on plugs where those folders are on the eSATA drive) (optional, default `false`)
- **mediafilter** - media type filter applied to the current directory content (image|audio|video|document|all)
- **folderfilter** - (true|false) if false, folders are filtered out
- **nbFilteredAssets** - total number of children files and folder returned in the document, after media type and folder filtering and paging
- **assetsOffset** - the offset from the beginning the directory listing

The **file** root node also contains zero or more **file** elements (one for each filtered/paged child file or folder) with the following attributes:

- **name** - name of file or directory
- **href** - file or directory URI
- **mtime** - last modification time in seconds since the epoch
- **type** - the folder or file type: folder|image|audio|video|document
- and for folders:
    - **nbFolders** - the number of folders in the folder
    - **nbFiles** - the number of files in the folder
    - **sharingWithDMA** - `true` if the folder is shared with DMA, `inherited` if the folder is shared with DMA because one of its ancestors is, `false` otherwise (optional, default `false`)
    - **deviceHref** - URI of the ejectable USB storage device where the folder resides (optional)
    - **readOnly** - (true|false), true for folders that cannot be deleted, copied, renamed or moved, and where no file or folder can be created/added/copied/moved (so far, only ExternalStorage) (optional, default `false`)
    - **isCoreMount** - (true|false), true for folders that are core mountpoints, i.e. can only be ejected by admins (so far, MyLibrary, FamilyLibrary, MyComputers on plugs where those folders are on the eSATA drive) (optional, default `false`)
- and for assets: (if available)
    - **mimeType** - Asset MIME type; audio/mpeg, image/jpeg etc.
    - **caption** - caption
    - **size** - file size in bytes
    - **tags** - `tags` element containing a `tag` element for each tag (in a text node)

See below an example implementation of hipserv/directoryContents+xml MIME type:

```
<?xml version="1.0" ?>

<file name="Samples" href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM" type="folder"
      parentHref="/api/2.0/rest/files/ZmFtaWx5" path="FamilyLibrary\Samples"
      nbFolders="1" nbFiles="4" mtime="1285158728" mediafilter="all"
      folderfilter="true" nbFilteredAssets="5">

  <file name="Sample Folder" type="folder"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvU2FtcGxlIEZvbGRlcg"
        sharingWithDMA="true" nbFolders="0" nbFiles="0" mtime="1285158852"/>

  <file name="06 - Dissolved Girl.mp3" type="audio"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM"
        mimeType="audio/mpeg" size="5873792" mtime="1285158715"/>

  <file name="Paris.avi" type="video"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp"
        mimeType="video/x-msvideo" size="16540732" mtime="1285158726">
    <tags>
      <tag>paris</tag>
    </tags>
  </file>

  <file name="paris01.jpg" type="image"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc"
        mimeType="image/jpeg" caption="Paris 2010" size="415297" mtime="1285158722">
    <tags>
      <tag>notre dame</tag>
      <tag>paris</tag>
    </tags>
  </file>

  <file name="USER-GUIDE.pdf" type="document"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvVVNFUi1HVUlERS5wZGY"
        mimeType="application/pdf" size="481239" mtime="1285158728"/>

</file>
```

hipserv/lightDirectoryAssets+xml

A hipserv/lightDirectoryAssets+xml document is a lightweight description of a HipServ directory, with the directory's URI and the list of all assets with their URI, name and type (subdirectories are excluded).

Sample:

```
<file href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM">
  <file name="06 - Dissolved Girl.mp3" type="audio"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM"/>
  <file name="Paris.avi" type="video"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp"/>
  <file name="paris01.jpg" type="image"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc"/>
  <file name="USER-GUIDE.pdf" type="document"
        href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvVVNFUi1HVUlERS5wZGY"/>
</file>
```

hipserv/assetDetails+xml

The hipserv/assetDetails+xml MIME type defines an XML data structure providing access to the details of a HipServ asset.

The root **file** element contains the following attributes:

- **name** - asset filename
- **href** - asset URI
- **parentHref** - URI of the parent folder
- **path** - Samba path to the asset (empty if the asset is not accessible via samba for the current visitor)
- **type** - type of asset: images|audio|video|document
- **mimeType** - MIME type of the asset for example image/jpeg
- **size** - number of bytes the asset occupies on the file system
- **mtime** - last modified time stamp in seconds since the epoch
- **caption** - descriptive end user supplied caption (if defined)
- type-specific metadata:
    - for image assets:
        - **height** - image height
        - **width** - image width
        - **timeTaken** - time taken (if defined)
    - for audio assets (each if defined):
        - **bitrate** - audio bitrate
        - **playtimeSeconds** - play time in seconds
        - **artist** - artist
        - **composer** - composer
        - **album** - album name
        - **year** - album year
        - **title** - track title
        - **savedPosition** - saved playback position in milliseconds
    - for video assets (each if defined):
        - **width** - video width
        - **height** - video height
        - **codec** - video codec
        - **bitrate** - video bitrate
        - **playtimeSeconds** - play time in seconds
        - **savedPosition** - saved playback position in milliseconds
    - for document assets: none

The root **file** element can also contain a single **tags** element containing one or more **tag** element, each containing a search tag associated with the asset in a text node.

See below an example implementation of hipserv/assetDetails+xml MIME type for each of the supported asset types:

Image Asset

```
<?xml version="1.0" ?>
<file name="paris01.jpg"
      href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc"
      type="image" parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM"
      path="FamilyLibrary\Samples\paris01.jpg" mimeType="image/jpeg"
      size="415297" mtime="1285158722" caption="Paris 2010"
      width="1024" height="641">
  <tags>
    <tag>notre dame</tag>
    <tag>paris</tag>
  </tags>
</file>
```

Audio Asset

```
<?xml version="1.0" ?>
<file name="06 - Dissolved Girl.mp3"
      href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM"
      type="audio" parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM"
      path="FamilyLibrary\Samples\06 - Dissolved Girl.mp3" mimeType="audio/mpeg"
      size="5873792" mtime="1285158715" bitrate="128" playtimeSeconds="366"
      artist="Massive Attack" album="Mezzanine" year="1998" title="Dissolved Girl"/>
```

Video Asset

```
<?xml version="1.0" ?>
<file name="Paris.avi"
      href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp"
      type="video" parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM"
      path="FamilyLibrary\Samples\Paris.avi" mimeType="video/x-msvideo"
      size="16540732" mtime="1285158726" width="720" height="400"
      codec="mpeg4" bitrate="291" playtimeSeconds="454">
  <tags>
    <tag>paris</tag>
  </tags>
</file>
```

Document Asset

```
<?xml version="1.0" ?>
<file name="USER-GUIDE.pdf"
    href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvVVNFUi1HVUlERS5wZGY"
    type="document" parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM"
    path="FamilyLibrary\Samples\USER-GUIDE.pdf" mimeType="application/pdf"
    size="481239" mtime="1285158728"/>
```

hipserv/searchResults+xml

The hipserv/searchResults+xml MIME type defines the XML data structure of search results.

The **searchResults** root element contains the following attributes:

- **keywords** - space-separated list of unique keywords of 3 characters or more
- **nbResults** - number of results returned (capped)
- **moreResults** - `true` if more files matching the keywords exist but were not returned because the number of results is capped

The **searchResults** root element also contains one **file** element per search result containing the following attributes:

- **name** - name of file or directory
- **href** - file or directory URI
- **mtime** - last modification time in seconds since the epoch
- **type** - the folder or file type: folder|image|audio|video|document
- **parentHref** - URI of the parent folder (empty if the file is a media source)
- **matchedFields** - comma-separated list of fields matching the searched keywords. Fields are "tags", "caption", "name".
- and for folders:
    - **nbFolders** - the number of folders in the folder
    - **nbFiles** - the number of files in the folder
    - **sharingWithDMA** - `true` if the folder is shared with DMA, `inherited` if the folder is shared with DMA because one of its ancestors is, `false` otherwise (optional, default `false`)
    - **deviceHref** - URI of the ejectable USB storage device where the folder resides (optional)
    - **readOnly** - (true|false), true for folders that cannot be deleted, copied, renamed or moved, and where no file or folder can be created/added/copied/moved (so far, only ExternalStorage) (optional, default `false`)
    - **isCoreMount** - (true|false), true for folders that are core mountpoints, i.e. can only be ejected by admins (so far, MyLibrary, FamilyLibrary, MyComputers on plugs where those folders are on the eSATA drive) (optional, default `false`)
- and for assets: (if available)
    - **mimeType** - Asset MIME type; audio/mpeg, image/jpeg etc.
    - **caption** - caption
    - **size** - file size in bytes
    - **tags** - `tags` element containing a `tag` element for each tag (in a text node)

See below an example implementation of hipserv/searchResults+xml MIME type:

```
<?xml version="1.0" ?>
<searchResults keywords="paris" nbResults="2">
  <file name="Paris.avi" href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvUGFyaXMuYXZp"
        type="video" mimeType="video/x-msvideo" size="16540732" mtime="1285158726"
        matchedFields="tags,name" parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM">
    <tags>
      <tag>paris</tag>
    </tags>
  </file>
  <file name="paris01.jpg" href="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXMvcGFyaXMwMS5qcGc"
        type="image" mimeType="image/jpeg" caption="Paris 2010" size="415297"
        mtime="1285158722" matchedFields="tags,caption,name"
        parentHref="/api/2.0/rest/files/ZmFtaWx5L1NhbXBsZXM">
    <tags>
      <tag>notre dame</tag>
      <tag>paris</tag>
    </tags>
  </file>
</searchResults>
```

hipserv/tagList+xml

A hipserv/tagList+xml document describes a list of tags associated with an asset.

Sample document:

```
<?xml version="1.0"?>
<tags>
  <tag>Vacation</tag>
  <tag>Beach</tag>
  <tag>Picnic</tag>
</tags>
```

hipserv/contacts+xml

A hipserv/contact+xml document describes a list of contacts, with their href, email and DisplayName.

Sample document:

```
<contacts>
  <contact href="/api/2.0/rest/contacts/johndoe/jenny%40example.com"
          email="jenny@example.com"  DisplayName="Jenny Doe"  />
  <contact href="/api/2.0/rest/contacts/johndoe/junior%40example.com"
          email="junior@example.com" DisplayName="Junior Doe" />
</contacts>
```

hipserv/contact+xml

A hipserv/contact+xml document describes a contact, with its href, email and DisplayName.

Sample document:

```
<contact href="/api/2.0/rest/contacts/johndoe/jenny%40example.com"
         email="jenny@example.com" DisplayName="Jenny Doe" />
```

hipserv/themes+xml

A hipserv/themes+xml document describes the list of themes available on the HipServ and follows the following template, i.e. a list of themes with href, id, localized names and the background-color style only:

```
<?xml version="1.0"?>
<themes>
  <theme rank="1" href="/api/2.0/rest/themes/hipserv/travel" id="hipservTravel">
    <names>
      <name locale="en_US" value="Travel"/>
      <name locale="de_DE" value="Reise"/>
      <name locale="es_ES" value="Viaje"/>
      <name locale="fr_FR" value="Voyage"/>
      <name locale="it_IT" value="Viaggio"/>
      <name locale="nl_NL" value="Reizen"/>
    </names>
    <styles>
      <style property="background-color" value="#7eb3d2"/>
    </styles>
  </theme>
  <theme rank="2" href="/api/2.0/rest/themes/hipserv/party" id="hipservParty">
    <names>
      <name locale="en_US" value="Party"/>
      <name locale="de_DE" value="Party"/>
      <name locale="es_ES" value="Fiesta"/>
      <name locale="fr_FR" value="Fête"/>
      <name locale="it_IT" value="Festa"/>
      <name locale="nl_NL" value="Feest"/>
    </names>
    <styles>
      <style property="background-color" value="#0d312f"/>
    </styles>
  </theme>
</themes>
```

hipserv/theme+xml

A hipserv/theme+xml document describes a theme and follows the following template:

```
<?xml version="1.0"?>
<theme rank="2" href="/api/2.0/rest/themes/hipserv/travel" id="hipservTravel">
  <names>
    <name locale="en_US" value="Travel"/>
    <name locale="de_DE" value="Reise"/>
    <name locale="es_ES" value="Viaje"/>
    <name locale="fr_FR" value="Voyage"/>
    <name locale="it_IT" value="Viaggio"/>
    <name locale="nl_NL" value="Reizen"/>
  </names>
  <styles>
    <!-- background color -->
    <style property="background-color"      value="#7eb3d2"/>
    <!-- font family -->
    <style property="font-family"           value="Georgia,Arial"/>
    <!-- navbar text color -->
    <style property="navbar-color"          value="#1e507b"/>
    <!-- navbar text rollover color -->
    <style property="navbar-color-rollover" value="#3495ac"/>
    <!-- default text color -->
    <style property="color"                 value="#ffffff"/>
    <!-- album name text color -->
    <style property="album-name-color"      value="#1e507b"/>
    <!-- media type text color -->
    <style property="media-type-color"      value="#ffffff"/>
    <!-- ruler color -->
    <style property="ruler-color"           value="#7b7a6e"/>
  </styles>
</theme>
```

The styles are those used by the HTML mobile UI. They can be used by other UIs or be ignored.

hipserv/albumListing+xml

A **hipserv/albumListing+xml** document is an XML document with a file root element with the following attributes:

- **name** - listing name
- **href** - listing URI
- **type** - `folder`
- **parentHref** 0 "" (empty string)
- **nbFolders** - number of albums in the listing
- **nbFiles** - 0
- **mediafilter** - `all`
- **folderfilter** - `true`
- **nbFilteredAssets** - same as nbFolders

The root element contains 0 or more file elements, each describing an album, they contain the following attributes:

- **name** - album name
- **type** - `folder`
- **href** - album URI
- **canDownload** - true|false: can gallery visitors download files from this albums?
- **theme** - theme URI
- **themeId** - theme ID
- **caption** - album caption (optional)
- **nbFiles** - approximate number of assets in the album

Each album can contain a single sharing element with 1 to 4 share elements, each with a different "with" attribute. Attributes of the share elements are:

- with: with whom the album is shared: dma|public|family|contacts
- until: until when the album is shared: YYYY-MM-DD (optional, not for dma)
- withPassword: do contacts require to type a password to see the album (optional, "true" if present, only for contacts)

<share with="contacts"> elements must contain 1 or more contact elements, each with the href, email address and DisplayName of the contact in the contact attribute.

Sample document:

```
<?xml version="1.0"?>
<file name="johndoe"
      href="/api/2.0/rest/albums/users/am9obmRvZQ"
      type="folder" parentHref="" nbFolders="1"
      nbFiles="0" mediafilter="all"
      folderfilter="true" nbFilteredAssets="1">

  <file name="Sample Album" type="folder"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0"
        canDownload="true" theme="/api/2.0/rest/themes/hipserv/travel"
        themeId="hipservTravel" caption="Here is the sample album!"
        nbFiles="5">

    <sharing>
      <share with="dma"/>
      <share with="public"/>

      <share with="contacts" until="2010-10-22" withPassword="true">
        <contact email="jenny@example.com"
                 href="/api/2.0/rest/contacts/johndoe/jenny%40example.com"
                 DisplayName="Jenny Doe"/>
        <contact email="junior@example.com"
                 href="/api/2.0/rest/contacts/johndoe/junior%40example.com"
                 DisplayName="Junior Doe"/>
      </share>

      <share with="family"/>
    </sharing>

  </file>

</file>
```

hipserv/albumContents+xml

A **hipserv/albumContents+xml** document is an XML document describing an album and its contents.

Its **file** root element contains the following attributes:

- **name** - album name
- **href** - album URI
- **parentHref** - URI of the parent folder (album listing)
- **type** - always `folder`
- **mtime** - last modification time in seconds since the epoch
- **nbFolders** - number of folders in the album (always 0)
- **nbFiles** - number of files in the album
- **sharingWithDMA** - `true` if the album is shared with DMA, `false` otherwise (optional, default `false`)
- **caption** - album caption/description/paragraph (optional)
- **theme** - theme URI
- **themeId** - theme Id
- **ownerFullName** - album owner full name
- **canDownload** - true|false: can gallery visitors download files from this albums?
- **publicShareURL** - URL of the album in the public gallery. Ignore it for non-public albums
- **RSS20URL** - URL of the RSS feed for the album (only for public albums and only if RSS support is enabled)
- **coverURI** - URI of the asset to use for the album cover, from the DB if defined, otherwise the first image, video, music or document asset in the album (in this order) (missing if the album is empty)

- **mediafilter** - media type filter applied to the current directory content (image|audio|video|document|all)
- **folderfilter** - (true|false) if false, folders are filtered out
- **nbFilteredAssets** - total number of children files and folder returned in the document, after media type and folder filtering and paging
- **assetsOffset** - the offset from the beginning the directory listing

If the current visitor is the owner of the album, the root file element can contain a single sharing element with 1 to 4 share elements, each with a different "with" attribute. Attributes of the share elements are:

- with: with whom the album is shared: dma|public|family|contacts
- until: until when the album is shared: YYYY-MM-DD (optional, not for dma)
- withPassword: do contacts require to type a password to see the album (optional, "true" if present, only for contacts)

<share with="contacts"> elements must contain 1 or more contact elements, each with the href, email address and DisplayName of the contact in the contact attribute.

If the album is linked with any 3rd party services (i.e., Flickr or Facebook), an externalServices element will be present containing the list of services that are linked (see sample).

The root file element can contain 0 or more file elements following the same pattern as those in hipserv/directoryContents+xml.

```
<?xml version="1.0"?>
<file name="Sample Album"
      href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0"
      type="folder" parentHref="/api/2.0/rest/albums/users/am9obmRvZQ"
      nbFolders="0" nbFiles="4" caption="Here is the sample album!"
      theme="/api/2.0/rest/themes/hipserv/travel"
      themeId="hipservTravel" ownerFullName="John Doe"
      canDownload="true"
      publicShareURL="http://www.myhipserv.com/share/johndoe/share/johndoe/Sample%20Album/"
      coverURI="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vcGFyaXMwMS5qcGc"
      RSS20URL="http://www.myhipserv.com/feeds/johndoe/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0
      mediafilter="all" folderfilter="false" nbFilteredAssets="4" assetsOffset="0">

  <sharing>
    <share with="dma"/>
    <share with="public"/>
    <share with="contacts" until="2010-10-22" withPassword="true">
      <contact email="jenny@example.com"
               href="/api/2.0/rest/contacts/johndoe/jenny%40example.com"
               DisplayName="Jenny Doe"/>
      <contact email="junior@example.com"
               href="/api/2.0/rest/contacts/johndoe/junior%40example.com"
               DisplayName="Junior Doe"/>
    </share>
    <share with="family"/>
  </sharing>

  <externalServices>
    <service name="facebook"/>
    <service name="flickr"/>
  </externalServices>

  <file name="06 - Dissolved Girl.mp3" type="audio"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vMDYgLSBEaXNzb2x2ZWQgR2lybC5tcDM"
        mimeType="audio/mpeg" size="5873792" mtime="1285158715"/>

  <file name="Paris.avi" type="video"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vUGFyaXMuYXZp"
        mimeType="video/x-msvideo" size="16540732" mtime="1285158726">
    <tags>
      <tag>paris</tag>
    </tags>
  </file>

  <file name="paris01.jpg" type="image"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vcGFyaXMwMS5qcGc"
        mimeType="image/jpeg" caption="Paris 2010" size="415297" mtime="1285158722">
    <tags>
      <tag>notre dame</tag>
      <tag>paris</tag>
    </tags>
  </file>

  <file name="USER-GUIDE.pdf" type="document"
        href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vVVNFUi1HVUlERS5wZGY"
        mimeType="application/pdf" size="481239" mtime="1285158728"/>

</file>
```

hipserv/lightAlbumAssets+xml

A hipserv/lightAlbumAssets+xml document is a light description of a HipServ album, with the album's URI and theme and the list of all assets with their URI, name and type.

Sample:

```
<file href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0" theme="/api/2.0/rest/themes/hipserv/trav
  <file name="06 - Dissolved Girl.mp3" type="audio" href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWx
  <file name="Paris.avi" type="video" href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vUGFyaXMuY
  <file name="paris01.jpg" type="image" href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vcGFyaXM
  <file name="USER-GUIDE.pdf" type="document" href="/api/2.0/rest/albums/users/am9obmRvZS9TYW1wbGUgQWxidW0vV
</file>
```

hipserv/albumRequest+xml

A **hipserv/albumRequest+xml** document is an XML document with a **file** root element with the following attributes:

- **name** - album name
- **type** - "folder"
- **caption** - album caption/description/paragraph (optional)
- **theme** - theme URI
- **canDownload** - true|false: can gallery visitors download files from this albums?

The root **file** element can contain a single **sharing** element with 1 to 4 **share** elements, each with a different **with** attribute. Attributes of the **share** elements are:

- **with** - with whom the album is shared: dma|public|family|contacts
- **until** - until when the album is shared: YYYY-MM-DD (optional, not for dma)
- **withPassword** - do contacts require to type a password to see the album (optional, "true" if present, only for contacts)

<share with="contacts"> elements must contain 1 or more **contact** elements, each with the URI of the contact in the **href** attribute.

```
<?xml version="1.0"?>
<file name="Sample Album" type="folder"
      caption="Here is the sample album!"
      theme="/api/2.0/rest/themes/hipserv/travel" canDownload="true">
  <sharing>
    <share with="dma"/>
    <share with="public"/>
    <share with="contacts" until="2010-10-22" withPassword="true">
      <contact href="/api/2.0/rest/contacts/johndoe/jenny%40example.com"/>
      <contact href="/api/2.0/rest/contacts/johndoe/junior%40example.com"/>
    </share>
    <share with="family"/>
  </sharing>
</file>
```

hipserv/3rdPartyPhotosetListing+xml

A **hipserv/3rdPartyPhotosetListing+xml** document lists the photosets available to the current authenticated HipServ User.

Its **photosets** root element contains 1 or more **photoset** elements (one per photoset), each with the following attributes:

- **id** - photoset id

- **name** - photoset name
- **caption** - photoset caption

```
<?xml version="1.0"?>
<photosets>
  <photoset id="0" name="Default Photostream" caption="Default Photostream"/>
  <photoset id="72157625010626410" name="Sample" caption="This is a sample album."/>
</photosets>
```

Retrieved from "http://devwiki.axentra.com/index.php/HS_2.0:_Web_Services_API"