

QNAP
File Station Web API
Version: 0.9

Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH QNAP® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN QNAP'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, QNAP ASSUMES NO LIABILITY WHATSOEVER, AND QNAP DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF QNAP PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

QNAP products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

QNAP may make changes to product specifications at any time, without notice. The information here is subject to change without notice. Contact your local QNAP sales office or your distributor to obtain the latest product specifications before placing your product order.

QNAP, QNAP logo, QTS, myQNAPcloud and VioStor are trademarks or registered trademarks of QNAP Systems, Inc. or its subsidiaries. Other names and brands may be claimed as the property of others.

Copyright © 2013, QNAP Systems, Inc. All rights reserved.

Rev	Date	Description
V0.9	29 Oct 2012	Initial draft
V0.91	30 Oct 2012	Add the "open" function
V0.92	22 Nov 2012	Add 10.3 section for "get_file_size" function
V0.93	03 Jan 2013	Modified hidden files option of get_list_file function
V0.94	10 Jan 2013	1.Add Compress section 6 2.Modified 11.1 section for recycle bin share folder list 3.Modified 7 download section which added "compress" parameter
V0.95	17 Jan 2013	1.Add Compress section 6.3 (Get compressing status) 2.Modified section 6.1(Compress files) 3.Add Recycle Bin Recovery section 16
V0.96	21 Jan 2013	Add Media Library section 17
V0.97	25 Jan 2013	1.Modified "Get file list" section 11.2 for video information. 2.Modified "Open" section 10 for transcode format type opening and media player for network page or flv player. 3.Add section 17.3 for media code list.
V0.98	01 Feb 2013	1.Add section 17.4 Get Media Library Files Transcode Status 2.Modified section 17.1. Media Library Transcoding Queue List return values 3.Add more info for "get file list"(section 11.2) which was filtered media type, format, filename

Content

1. Authentication	6
2. Create Folder	7
3. Rename	8
4. Copy	9
5. Move	10
6. Extract	11
6.1 List the Contents of a Zipped File	11
6.2 Extract Files	12
6.3 Cancel Extraction Process	13
7. Compress	14
7.1 Compress Files	14
7.2 Cancel Compress Process	15
7.3 Get Compress status	16
8. Download	17
9. Get Image Thumbnail	18
10. Upload	19
11. Open	20
12. Get List	22
12.1 Get Folder List	22
12.2 Get File List	24
12.3 Get Total Files Size	27
13. Delete	28
14. Get Status	29
15. Set Access Time	30
16. Search	31
17. Share File	33

17.1 Download shared file	33
17.2 Get share link	34
17.3 Send share link	36
17.4 Get shared file list	38
17.5 Get hostname and external IP address	40
17.6 Stop file(s) sharing	41
17.7 Stop all file sharing	42
17.8 Update share link	43
18. Recycle Bin	45
18.1 Get Recycle Bin Tree	45
18.2 Recycle Bin Recovery	46
18.3 Cancel Recycle Bin Recovery	47
18.4 Clean Recycle Bin	48
19. Media Library	49
19.1 Get status of media library transcoding Queue	49
19.2 Add/Delete Media Library Transcoding Queue	51
19.3 Get Media Library Transcode Files List	53
19.4 Get Media Library Files Transcode Status	54
19.5 Delete Transcode	55
19.6 Get Transcoding File Status	56
19.7 Cancel Transcode	57

■ **Overview**

File Station on QTS 4.0 provides a programmable interface allowing the 3rd party integrator/installer to develop application that is highly integrated with File Station. This interface is called "File Station Web API". The API is based on HTTP protocol, including functions like folder and file listing, uploading, copying, renaming, deleting, and..., etc.

1. Authentication

Description:

Authenticate the caller by checking user name and password. If passed, the API will return a session ID (sid) so that the caller can use the sid for further file operations.

Command:

[http://IP:8080/cgi-bin/authLogin.cgi?user=\\${username}&pwd=\\${encode_string}](http://IP:8080/cgi-bin/authLogin.cgi?user=${username}&pwd=${encode_string})

Parameter	Description
<code>\${username}</code>	Login user name
<code>\${encode_string}</code>	Password (Please download "get_sid.js" to get ezEncode function) http://eu1.qnap.com/Storage/SDK/get_sid.js

Note: `encode_string = ezEncode(utf16to8('${real_password}))`

Example:

<http://IP:8080/cgi-bin/authLogin.cgi?user=admin&pwd=YWRtaW4%3D>

(if `real_password` is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

Example of successful authentication:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QDocRoot version="1.0">
<authPassed><![CDATA[1]]></authPassed>
<authSid><![CDATA[ra108opo]]></authSid>           <= here is sid
<isAdmin><![CDATA[1]]></isAdmin>
</QDocRoot>
```

Example of failed authentication:

```
<QDocRoot version="1.0">
<authPassed>0</authPassed>
<errorValue>-1</errorValue>
</QDocRoot>
```

2. Create Folder

Description:

Create a folder in the specified path.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=createdir&sid=\\${sid}&dest_folder=\\${dest_folder}&dest_path=\\${dest_path}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=createdir&sid=${sid}&dest_folder=${dest_folder}&dest_path=${dest_path})

Parameter	Description
<code>\${sid}</code>	Input sid for authentication
<code>\${dest_folder}</code>	Folder name
<code>\${dest_path}</code>	Path of the folder

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=createdir&sid=xxxxxxx&dest_folder=test&dest_path=/Public

Create a folder named "test" in /Public

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

File exists

```
{ "status": 2, "success": "true" }
```

Permission denied

```
{ "status": 4, "success": "true" }
```


3. Rename

Description:

Rename a folder or file in the specified path.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=rename&sid=\\${sid}&path={path}&source_name=\\${source_name}&dest_name=\\${dest_name}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=rename&sid=${sid}&path={path}&source_name=${source_name}&dest_name=${dest_name})

Parameter	Description
<code>\${sid}</code>	Input sid for authentication
<code>{path}</code>	Path of the folder/ file
<code>\${source_name}</code>	Current folder/ file name to be changed
<code>\${dest_name}</code>	New folder/ file name

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=rename&sid=xxxxxxx&path=/Public&source_name=test&dest_name=test1

Rename a folder named "test" in /Public to "test1"

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

File exists

```
{ "status": 2, "success": "true" }
```

Permission denied

4. Copy

Description:

Copy a file/folder from the source to the destination.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=copy&sid=\\${sid}&source_file=\\${source_file}&source_total=\\${source_total}&source_path=\\${source_path}&dest_path=\\${dest_path}&mode=\\${mode}&dup=\\${dup}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=copy&sid=${sid}&source_file=${source_file}&source_total=${source_total}&source_path=${source_path}&dest_path=${dest_path}&mode=${mode}&dup=${dup})

Parameter name	Description
<code>\${source_file}</code>	Name of the copied file/folder
<code>\${source_total}</code>	Total number of copied files/folders
<code>\${source_path}</code>	Source path of the copied file/folder
<code>\${dest_path}</code>	Destination of the copied file/folder
<code>\${mode}</code>	1: skip, 0: overwrite
<code>\${dup}</code>	The duplication file name when copying the same destination with source files/folders.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=copy&sid=xxxx&source_file=test&source_total=1&source_path=/Public&dest_path=/Qdownload&mode=1

Copy a file/folder named "test" from /Public to /Qdownload.

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=copy&sid=xxxx&source_file=test&source_total=1&source_path=/Public&dest_path=/Public&mode=0&dup=copy

Copy a file/folder named "test" from /Public to /Public. (/Public/test → /Public/test-copy(1))

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

Permission denied

```
{ "status": 4, "success": "true" }
```

5. Move

Description:

Move a file/folder from the source to the destination.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=move&sid=\\${sid}&source_file={source_file}&source_total=\\${source_total}&source_path=\\${source_path}&mode=\\${mode}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=move&sid=${sid}&source_file={source_file}&source_total=${source_total}&source_path=${source_path}&mode=${mode})

Parameter name	Description
<code>\${source_file}</code>	Name of the copied file/folder
<code>\${source_total}</code>	Total number of copied files/folders
<code>\${source_path}</code>	Source path of the copied file/folder
<code>\${dest_path}</code>	Destination of the copied file/folder
<code>\${mode}</code>	1: skip, 0: overwrite

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=move&sid=xxxx&source_file=test&source_total=1&source_path=/Qweb&dest_path=/Public&mode=1

Move a file/folder named "test" from /Qweb to /Public. If the same file exists on the destination, do nothing.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

Permission denied

```
{ "status": 4, "success": "true" }
```

6. Extract

6.1 List the Contents of a Zipped File

Description:

List the content of a zipped file.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_extract_list&sid=\\${sid}&path=\\${path}&extract_file=\\${extract_file}&code_page=\\${code_page}&start=\\${start}&limit=\\${limit}&sort=\\${sort}&dir=\\${dir}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_extract_list&sid=${sid}&path=${path}&extract_file=${extract_file}&code_page=${code_page}&start=${start}&limit=${limit}&sort=${sort}&dir=${dir})

Parameter name	Description
<code>\${extract_file}</code>	Path of the extracted file
<code>\${code_page}</code>	Extracting code page (UTF-8 only)
<code>\${start}</code>	Start number of the listed contents
<code>\${limit}</code>	Total number of the listed contents
<code>\${sort}</code>	Sorting type (filename, file_size, compress_size, mt)
<code>\${dir}</code>	List type (ASC, DESC)

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_extract_list&sid=xxxx&path=/&extract_file=/Public/test.zip&code_page=UTF-8&start=0&limit=50&sort=filename&dir=ASC

List the contents of "test.zip" from the first file to the fifth one and sort them by file name and ASC.

Return value:

Success:

```
{ "total": 5, "datas": [ {  
  "filename": "7-zip.chm",  
  "isfolder": 0,  
  "filesize": "88124",  
  "mt": "2009-02-03 12:22:54",  
  "id": 0,  
  "compress_size": "81172" } ]  
}
```

Failure:

```
File doesn't exist  
"status": 5, "success": "true" }
```

6.2 Extract Files

Description:

Extract files from a zipped file to the specified path on NAS. This is an asynchronous API. The caller can use the returned process ID (pid) for further operations.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=extract&sid=\\${sid}&extract_file=\\${extract_file}&dest_path=\\${dest_path}&pwd=\\${pwd}&mode=\\${mode}&overwrite=\\${overwrite}&path_mode=\\${path_mode}&code_page=\\${code_page}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=extract&sid=${sid}&extract_file=${extract_file}&dest_path=${dest_path}&pwd=${pwd}&mode=${mode}&overwrite=${overwrite}&path_mode=${path_mode}&code_page=${code_page})

Parameter name	Description
<code>\${extract_file}</code>	Path of the extracted files
<code>\${code_page}</code>	Extracting code page (UTF-8 only)
<code>\${dest_path}</code>	Destination of the extracted files
<code>\${pwd}</code>	Extraction password (can be null)
<code>\${mode}</code>	Extraction mode (extract_all, extract_part)
<code>\${overwrite}</code>	1: overwrite, 0: skip
<code>\${path_mode}</code>	full: extract file with full path none: don't extract file with full path
<code>\${code_page}</code>	Extracting code page (UTF-8 only)

If mode is extract_part

Parameter name	Description
part_file	Name of the file to be extracted (can be more than one)
part_total	The total number of extracted files

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=extract&sid=xxxx&extract_file=/Public/test.zip&dest_path=/Public&pwd=&mode=extract_all&overwrite=1&path_mode=full&code_page=UTF-8

Extract test.zip to /Public by UTF-8. If the file exists, overwrite the file.

Return value:

Success:

```
{ "status": 1, "pid": 28656 }
```

Failure:

Permission denied

```
{ "status": 4, "success": "true" }
```

6.3 Cancel Extraction Process

Description:

Cancel the extraction process by process ID.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_extract&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_extract&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Pid of the extracting process

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_extract&sid=xxxx&pid=29690

Cancel the extraction process which the pid is 29690.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

```
{ "status": 0, "success": "true" }
```

7. Compress¹

7.1 Compress Files

Description:

Compress files to a specified file on NAS. This is an asynchronous API. The caller can use the returned process ID (pid) for further operations.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=compress&sid=\\${sid}&compress_name=\\${compress_name}&type=\\${type}&level=\\${level}&pwd=\\${pwd}&encrypt=\\${encrypt}&total=\\${total}&compress_file=\\${compress_file}...](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=compress&sid=${sid}&compress_name=${compress_name}&type=${type}&level=${level}&pwd=${pwd}&encrypt=${encrypt}&total=${total}&compress_file=${compress_file}...)

Parameter name	Description
<code>\${extract_file}</code>	Path of the extracted files
<code>\${compress_name}</code>	The compressed name
<code>\${type}</code>	Compressed format type (<i>7z/zip</i>)(default:zip)
<code>\${pwd}</code>	The compressed password (can be null)
<code>\${level}</code>	Compressed level (<i>normal/large/fast</i>) (default:normal)
<code>\${encrypt}</code>	<i>7z:AES256, zip:ZipCrypto/AES256</i> (can be null)
<code>\${path}</code>	The compressed file path
<code>\${total}</code>	The amount of compression files number
<code>\${compress_file}</code>	The compressed file name

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=compress&sid=xxxx&compress_name=test&type=7z&level=large&pwd=123456&encrypt=AES256&path=/Download&total=2&compress_file=4.jpg&compress_file=sample001.jpg

Return value:

Success:

```
{ "status": 1, "pid": 28656 }
```

Failure:

Permission denied

```
{ "status": 4, "success": "true" }
```

¹ * for QTS 4.0 and later

7.2 Cancel Compress Process

Description:

Cancel the compressing process by process ID.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_compress&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_compress&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Pid of the compressing process

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_compress&sid=xxxx&pid=29690

Cancel the compression process which the pid is 29690.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

```
{ "status": 0, "success": "true" }
```


7.3 Get Compress status

Description:

Get the compressing status by process ID.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_compress_status&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_compress_status&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Pid of the compressing process

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_compress_status&sid=xxxx&pid=29690

Get the compression process which the pid is 29690.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Compressing:

```
{ "status": 6, "success": "true" }
```

Failure:

```
{ "status": 0, "success": "true" }
```

8. Download

Description:

Download a file, a folder, or multiple files as a zip file, a compressed file, or just the same as they are.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=download&sid=\\${sid}&isfolder=\\${isfolder}&compress=\\${compress}&source_path=\\${source_path}&source_file=\\${source_file}&source_total=\\${source_total}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=download&sid=${sid}&isfolder=${isfolder}&compress=${compress}&source_path=${source_path}&source_file=${source_file}&source_total=${source_total})

Parameter name	Description
<code>\${isfolder}</code>	The request file is a folder. <i>1</i> : yes, <i>0</i> : no.
<code>\${compress}</code>	If the request file is a folder or files then <i>0/1</i> : zip archive only, <i>2</i> : compress files. else the request file is a single file then <i>0</i> : transfer the original file to client <i>1</i> : zip archive only <i>2</i> : compress files
<code>\${source_path}</code>	Path of the file. Start with the share name.
<code>\${source_file}</code>	File name.
<code>\${source_total}</code>	Total number of files.

Example:

Download a single file.

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=download&sid=xxxx&isfolder=0&compress=0&source_path=/Public&source_file=test.txt&source_total=1

Download multiple files, it will return an archive zip file.

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=download&sid=xxxx&isfolder=0&source_path=/Public&source_file=test.txt&source_file=test2.txt&source_total=2

Return value:

Success:

Return a file for download

Failure:

Return empty contents

9. Get Image Thumbnail

Description:

Get a size-specified thumbnail of an image file.

Command:

<http://IP:8080/cgi-bin/filemanager/>

[utilRequest.cgi?func=\\${func}&sid=\\${sid}&path=\\${source_path}&name=\\${source_file}&size=\\${size}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=${func}&sid=${sid}&path=${source_path}&name=${source_file}&size=${size})

Parameter name	Description
<code>\${func}</code>	<code>get_thumb</code>
<code>\${path}</code>	Path of the file. Start with the share name.
<code>\${name}</code>	File name.
<code>\${size}</code>	80/320/640; default value:320

Example:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi? func=get_thumb&](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_thumb&sid=xxxx&path=/Public&name=sample.jpg&size=80)

[sid=xxxx&path=/Public&name=sample.jpg&size=80](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_thumb&sid=xxxx&path=/Public&name=sample.jpg&size=80)

get Wx80 or 80xH thumbnail of sample.jpg

10. Upload

Description:

Upload a file.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=\\${func}&type=\\${type}&sid=\\${sid}&dest_path=\\${dest_path}&overwrite=\\${overwrite}&progress=\\${progress}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=${func}&type=${type}&sid=${sid}&dest_path=${dest_path}&overwrite=${overwrite}&progress=${progress})

Parameter name	Description
<code>\${func}</code>	<i>upload</i>
<code>\${type}</code>	<i>standard</i>
<code>\${dest_path}</code>	Destination dir path
<code>\${overwrite}</code>	1: overwrite, 0: skip
<code>\${progress}</code>	Destination file path, "/" needs to be replaced with "-"

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=upload&type=standard&sid=xxxx&dest_path=/Public&overwrite=1&progress=-Public-test.zip

Upload test.zip to /Public. If the file exists, overwrite the file.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

Permission denied

```
{ "status": 4, "success": "true" }
```

Quota limit exceeded

```
{ "status": 9, "success": "true" }
```

11. Open

Description:

Open a multimedia file stream for viewing or playing.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi/\\${source_file}?sid=xxxx&func=get_viewer&source_path=\\${source_path}&source_file=\\${source_file}&player=\\${player}&rtt=\\${rtt}format=\\${format}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi/${source_file}?sid=xxxx&func=get_viewer&source_path=${source_path}&source_file=${source_file}&player=${player}&rtt=${rtt}format=${format})

Parameter name	Description
<code>\${func}</code>	<i>open</i>
<code>\${source_path}</code>	Source dir path
<code>\${source_file}</code>	Source file name
<code>\${player}</code>	Use the jwplayer player or not (1/0)
<code>\${rtt}</code>	1:FLV for video or music, 3:MP3 for music
<code>\${format}</code>	Video trnascode format type for opening. <code>\${format}</code> can be : <i>mp4_360 / mp4_720 / flv_720</i>

Example:

http://172.17.20.38:8080/cgi-bin/filemanager/utilRequest.cgi/123.jpg?sid=m3cop5xh&func=get_viewer&source_path=/Download&source_file=123.jpg

Open 123.jpg file.

http://172.17.20.38:8080/cgi-bin/filemanager/utilRequest.cgi/Dream.mp4?sid=m3cop5xh&func=get_viewer&source_path=/Download&source_file=Dream.mp4&format=mp4_360

Open Dearn4.mp4 360p formt type media code.

http://172.17.20.38:8080/cgi-bin/filemanager/utilRequest.cgi/123.avi?sid=m3cop5xh&func=get_viewer&source_path=/Download&source_file=123.avi&player=1

Open 123.avi with jwplayer.

http://172.17.20.38:8080/cgi-bin/filemanager/utilRequest.cgi/123.avi?sid=m3cop5xh&func=get_viewer&source_path=/Download&source_file=123.avi&rtt=1

Open 123.avi with FLV player.

Return value:

Success:

Return a file for viewer

Failure:

Return empty contents

12. Get List

12.1 Get Folder List

Description:

Get folder list in a folder, a shared iso, the shared root folder, or the recycle bin.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=\\${sid}&is_iso=\\${is_iso}&node=\\${node}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=${sid}&is_iso=${is_iso}&node=${node})

Parameter name	Description
<code>\${is_iso}</code>	Is a iso share. <i>1</i> : yes, <i>0</i> : no. Default is <i>0</i> . This value is according to a field "iconCls" in get_tree response. If "iconCls" is "iso", this value is <i>1</i> .
<code>\${node}</code>	Target folder path. Use folder path to get folder list, and use the value with <code>'share_root'</code> to get share list, or use the value with <code>'recycle_root'</code> to get recycle bin share list.

Example:

1. Get share list

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=xxxx&is_iso=0&node=share_root`

2. Get folder list

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=xxxx&is_iso=0&node=/Public`

3. Get folder list (in iso share)

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=xxxx&is_iso=1&node=/folder_name`

4. Get recycle bin list

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=xxxx&node=recycle_root`

Return value:

Success:

```
[
  {
    "text": "Luke",
    "id": "\Luke",
    "cls": "w",
    "draggable": 0,
    "iconCls": "folder",
    "noSupportACL": 0
  },{
    "text": "android-x86-1.6-r2",
    "id": "\android-x86-1.6-r2",
    "cls": "r",
    "draggable": 0,
    "iconCls": "iso",
    "noSupportACL": 0
  }
]
```

Response field:

Field	Description
cls	Folder permission. <i>r</i> : read only, <i>w</i> : read write
iconCls	"iso" for iso share
noSupportACL	<i>0</i> : support ACL, <i>1</i> : not support ACL

Fail:

```
Permission deny
{ "status": 4, "success": "true" }
```


12.2 Get File List

Description:

Retrieve both file and folder list in a specified folder or a shared iso, with filters such as response record count, file type..., etc.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_list&sid=\\${sid}&is_iso=\\${is_iso}&list_mode=\\${list_mode}&path=\\${path}&dir=\\${dir}&limit=\\${limit}&sort=\\${sort}&start=\\${start}&hidden_file=\\${hidden_file}&type=\\${type}&mp4_360=\\${mp4_360}&mp4_720=\\${mp4_360}&flv_720=\\${flv_720}&filename=\\${filename}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_list&sid=${sid}&is_iso=${is_iso}&list_mode=${list_mode}&path=${path}&dir=${dir}&limit=${limit}&sort=${sort}&start=${start}&hidden_file=${hidden_file}&type=${type}&mp4_360=${mp4_360}&mp4_720=${mp4_360}&flv_720=${flv_720}&filename=${filename})

Parameter name	Description
<code>\${is_iso}</code>	Is a iso share. 1: yes, 0: no
<code>\${list_mode}</code>	Value is "all"
<code>\${path}</code>	Folder path
<code>\${dir}</code>	Sorting direction. ASC: Ascending , DESC: Descending
<code>\${limit}</code>	Number of response datas
<code>\${sort}</code>	Sort field (filename/filesize/filetype/mt/privilege/owner/group)
<code>\${start}</code>	Response data start index
<code>\${hidden_file}</code>	List hidden file or not. 0:donnot list hidden files, 1:list files
<code>\${type}</code>	1: MUSIC, 2:VIDEO, 3:PHOTO (1/2/3)
<code>\${mp4_360}</code>	Video format type mp4_360 true or not(1/0)
<code>\${mp4_720}</code>	Video format type mp4_720 true or not(1/0)
<code>\${flv_720}</code>	Video format type flv_720 true or not(1/0)
<code>\${filename}</code>	Search video file name

Example 1:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_list&sid=xxxx&is_iso=0&list_mode=all&path=/Public&dir=ASC&limit=20&sort=filename&start=0

Example 2:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_list&sid=xxxx&path=/Public&dir=ASC&limit=20&sort=natural&start=0&limit=100type=2&flv_720=1&mp4_360=1&filename=Dream.mp4

(list for type is VIDEO(type=2) and format FLV 720(flv_720=1) or format MP4 360(mp4_360=1) and file name is Dream.mp4(filename=Dream.mp4))

Return value:

Success:

```

{
  "total": 1,
  "acl": 7,
  "is_acl_enable": 1,
  "is_winacl_support": 1,
  "is_winacl_enable": 0,
  "rtt_support": 1,
  "page": 1,
  "datas": [
    {
      "filename": "TS-439_20110504-3.5.0.img",
      "isfolder": 0,
      "filesize": "137595818",
      "group": "administrators",
      "owner": "admin",
      "iscompressed": 0,
      "privilege": "770",
      "filetype": 0,
      "mt": "2011\05\06 14:35:20\n",
      "mp4_360": 0,
      "mp4_720": 0,
      "flv_720": 0,
      "trnas": 0,
      "play": 0
    }
  ]
}

```

Response field:

Field	Description
total	Total numer.
acl	ACL permission. 7: read write, 4: read only, 0: deny
is_acl_enable	ACL if enable. 0: not enable, 1: enable
is_winacl_support	WIN ACL if support. 0: not support, 1: support
is_winacl_enable	WIN ACL if enable. 0: not enable, 1: enable
rtt_support	rtt if support. 0: not support, 1: support
page	List media type that the file position page(for media type) (for example 2)
datas	List datas

filename	File name
isfolder	1: folder, 0: file
filesize	File size. unit: byte
filetype	1:MUSIC, 2:VIDEO, 3:PHOTO, 0:undefined
group	File group
owner	File owner
mp4_360	mp4 360p transcoded media ready or not
mp4_720	mp4 720p transcoded media ready or not
flv_720	flv 720 transcoded media ready or not
trans	added transcode or not
play	Is mp4 video format for playing or not(in the "@Transcode" folder)

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

12.3 Get Total Files Size

Description:

Get total files size in a specified path. The size counting includes hidden file and folder.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_file_size&sid=\\${sid}&path=\\${path}&total=\\${total}&name=\\${name0}\\$name=\\${name1}&name=....](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_file_size&sid=${sid}&path=${path}&total=${total}&name=${name0}$name=${name1}&name=....)

Parameter name	Description
<code>\${path}</code>	Folder path
<code>\${total}</code>	The number of file/folder which are calculating the total size
<code>\${name}</code>	file or folder name

Example:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_file_size&sid=XXXX&path=/Download&total=3&name=folder1\\$name=file1&name=file2](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_file_size&sid=XXXX&path=/Download&total=3&name=folder1$name=file1&name=file2)

Return value:

Success:

```
{ "status": 1, "size": "12594446336" }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

13. Delete

Description:

Delete folder(s)/file(s) in a specified path.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=\\${sid}&path=\\${path}&file_total=\\${file_total}&file_name=\\${file_name}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=${sid}&path=${path}&file_total=${file_total}&file_name=${file_name})

Parameter name	Description
<code>\${path}</code>	Folder path.
<code>\${file_total}</code>	Total number of folder/file(s).
<code>\${file_name}</code>	Folder/file name.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=xxxx&path=/Public&file_total=2&file_name=folder name&file_name=file name

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

14. Get Status

Description:

Get status of folder(s)/file(s), such as file size, privilege..., etc.

Command:

<http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat>

[&sid=\\${sid}&path=\\${path}&file_total=\\${file_total}&file_name=\\${file_name}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat&sid=${sid}&path=${path}&file_total=${file_total}&file_name=${file_name})

Parameter name	Description
<code>\${path}</code>	Folder path.
<code>\${file_total}</code>	Total number of folder/file(s).
<code>\${file_name}</code>	Folder/file name.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat&sid=xxxx&path=/Public&file_total=2&file_name=folder name&file_name=file name

Return value:

Success:

```
{ "total": 1, "acl": 1, "is_acl_enable": 0, "is_winacl_support": 1,
  "is_winacl_enable": 1, "datas": [ { "filename": "sample008.jpg", "isfolder": 0, "filesize":
  "51445", "group": "everyone", "owner": "httpdusr", "iscompressed": 0, "privilege":
  "776", "filetype": 0, "mt": "2012\05\10 14:43:30", "epochmt": 1336632210 } ] }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

15. Set Access Time

Description:

Set folder(s)/file(s) modification time.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat &sid=\\${sid}&settime=\\${settime}&mtime=\\${timestamp}&path=\\${path}&file_total=\\${file_total}&file_name=\\${file_name}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat &sid=${sid}&settime=${settime}&mtime=${timestamp}&path=${path}&file_total=${file_total}&file_name=${file_name})

Parameter name	Description
<code>\${path}</code>	Folder path.
<code>\${file_total}</code>	Total number of folder/file(s).
<code>\${file_name}</code>	Folder/file name.
<code>\${settime}</code>	1: set modification time
<code>\${timestamp}</code>	Epoch time (seconds since 1970-01-01 00:00:00 UTC) The modification time will be set current datetime on the server if not specified.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=stat&sid=xxxx&settime=0&mtime=1336632200&path=/Public&file_total=2&file_name=folder_name&file_name=file_name

Return value:

Success:

```
{ "total": 1, "acl": 1, "is_acl_enable": 0, "is_winacl_support": 1, "is_winacl_enable": 1, "datas": [ { "filename": "sample008.jpg", "isfolder": 0, "filesize": "51445", "group": "everyone", "owner": "httpdusr", "iscompressed": 0, "privilege": "776", "filetype": 0, "mt": "2012\05\10 14:43:20", "epochmt": 1336632200 } ] }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

16. Search

Description:

Search file/folder by key word within a specified path.

Command:

```
http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=search&sid=${sid}&keyword=${keyword}&source_path=${source_path}&dir=${dir}&limit=${limit}&sort=${sort}&start=${start}
```

Parameter name	Description
<code>\${is_iso}</code>	Is a iso share. 1: yes, 0: no
<code>\${keyword}</code>	keyword
<code>\${source_path}</code>	Folder path
<code>\${dir}</code>	Sorting direction. ASC: Ascending , DESC: Descending
<code>\${limit}</code>	Number of response data
<code>\${sort}</code>	Sort field (filename/filesize/filetype/mt/privilege/owner/group)
<code>\${start}</code>	Response data start index

Example:

```
http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=search&sid=xxxx&keyword=keyword&source_path=/Public&dir=ASC&limit=20&sort=filename&start=0
```

Return value:

Success:

```
{
  "total": 1,
  "acl": 7,
  "is_acl_enable": 1,
  "datas": [
    {
      "filename": "\\share\\HDA_DATA\\Public\\keyword.txt",
      "isfolder": 0,
      "filesize": "5818",
      "group": "administrators",
      "owner": "admin",
      "iscompressed": 0,
      "privilege": "770",
      "filetype": 0,
      "mt": "2011\\05\\06 14:35:20\\n"
```



```
}
]
}
```

Response field:

Field	Description
total	Total numer.
acl	ACL permission. 7: read write, 4: read only, 0: deny
is_acl_enable	ACL if enable. 0: not enable, 1: enable
datas	List datas
filename	File name. It must to remove "/share/HDX_DATA/" to display or do any actions.
isfolder	1: folder, 0: file
filesize	File size. unit: byte
group	File group
owner	File owner

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

17. Share File

17.1 Download shared file

Description:

Download a shared file by an unique ID (ssid).

Command:

[http://IP:8080/share.cgi?ssid=\\${unique_id}](http://IP:8080/share.cgi?ssid=${unique_id})

Parameter name	Description
<code>\${unique_id}</code>	Use the unique id to download the shared file.

Example:

<http://IP:8080/share.cgi?ssid=xxxxxxxxxxxxxxxxxxxx>

Return value:

Got the shared file from browser !!

Wirte Log:

[WFM2] Download share file /Public/Interface.txt. From IP : 172.17.20.60.

17.2 Get share link

Description:

Create share links of specified files, and retrieve or email the links to someone.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_link&sid=\\${sid}&access_code=\\${access_code}&access_enabled=\\${access_enabled}&network_type=\\${network_type}&download_type=\\${download_type}&ssl=\\${ssl}&valid_duration=\\${valid_duration}&day=\\${day}&hour=\\${hour}&path=\\${path}&file_total=\\${file_total}&file_name=\\${file_name}&hostname=\\${hostname}&addressee=\\${addressee}&subject=\\${subject}&content=\\${content}&include_access_code=\\${include_access_code}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_link&sid=${sid}&access_code=${access_code}&access_enabled=${access_enabled}&network_type=${network_type}&download_type=${download_type}&ssl=${ssl}&valid_duration=${valid_duration}&day=${day}&hour=${hour}&path=${path}&file_total=${file_total}&file_name=${file_name}&hostname=${hostname}&addressee=${addressee}&subject=${subject}&content=${content}&include_access_code=${include_access_code})

Parameter name	Description
<code>\${network_type}</code>	internet: from network, local: from local
<code>\${download_type}</code>	create_download_link: create download link, email_download_link: email download link
<code>\${valid_duration}</code>	specific_time: specific the download time, period_of_time: time period, forever: forver download load
<code>\${hostname}</code>	host IP or domain name
<code>\${day}</code>	shared day time if valid_duration=period_of_time
<code>\${hour}</code>	shared hour time if valid_duration=period_of_time
<code>\${file_total}</code>	shared datetime if valid_duration=specific_time
<code>\${file_name}</code>	0(zero) if valid_duration=forever
<code>\${path}</code>	download file path
<code>\${ssl}</code>	enabled ssl
<code>\${access_enabled}</code>	enable access code
<code>\${access_code}</code>	access code
<code>\${include_access_code}</code>	email contents include access code or not
<code>\${addressee}</code>	get email addressee
<code>\${subject}</code>	get email subject
<code>\${content}</code>	get email content

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_link&sid=ahe84nc5&access_code=GolZ9hiI&access_enabled=true&network_type=internet&download_type=email_download_link&ssl=true&valid_duration=period_of_time&day=07&hour=00&path=/Multimedia/Samples&file_total=1&file_name=sample01.jpg&hostname=172.17.20.62&addressee=admin@qnap.com&subject=To Share with You&content=content&include_access_code=true&

Return value:

Check sharing URL response:

```
{
  "status": 1,
  "expire_time": 1337310416,
  "total": 1,
  "count": 0,
  "limit": 0,
  "misc": "----",
  "links": [ { "filename": "\Multimedia\Annekei - Tsuki\01 - Tsuki.flac",
"link_url": "https://172.17.20.65/share.cgi?ssid=0BPNigh" } ]
}
```

Response field:

Field	Description
status	1.(for UI reflash parameter)(donot reload web page)
expire_time	expire time(seconds)
total	total shared file
creator	creator
count	reserved
limit	reserved
misc	reserved
links	link data contents

17.3 Send share link

Description:

Create share link and email to someone.

Command:

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=share_file&sid=${sid}&dowdownload_type=${dowdownload_type}&addressee=${addressee}&subject=${subject}&content=${content}&valid_duration=${valid_duration}&day=${day}&hour=${hour}&file_total=${file_total}&access_code=${access_code}&access_enabled=${access_enabled}&hostname=${hostname}&ssl=${ssl}&include_access_code=${include_access_code}&file_name=${file_name}&path=${path}&network_type=${network_type}&link_url=${link_url}&mail_content_date=${mail_content_date}&mail_content_pwd=${mail_content_pwd}&expire_time=${expire_time}`

Parameter name	Description
<code>\${network_type}</code>	internet: from network, local: from local
<code>\${download_type}</code>	create_download_link: create download link, email_download_link: email download link
<code>\${valid_duration}</code>	specific_time: specific the download time, period_of_time: time period, forever: forver download load
<code>\${hostname}</code>	host IP or domain name
<code>\${day}</code>	shared day time if valid_duration=period_of_time
<code>\${hour}</code>	shared hour time if valid_duration=period_of_time
<code>\${file_total}</code>	shared datetime if valid_duration=specific_time
<code>\${file_name}</code>	0(zero) if valid_duration=forever
<code>\${path}</code>	download file path
<code>\${ssl}</code>	enabled ssl
<code>\${access_enabled}</code>	enable access code
<code>\${access_code}</code>	access code
<code>\${include_access_code}</code>	email contents include access code or not
<code>\${addressee}</code>	get email addressee
<code>\${subject}</code>	get email subject
<code>\${content}</code>	get email content
<code>\${link_url}</code>	internet: from network, local: from local
<code>\${mail_content_date}</code>	mail contents for valid date information
<code>\${mail_content_pwd}</code>	mail contents for password information
<code>\${expire_time}</code>	expire time(seconds)

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=share_file&sid=ahe84nc5&access_code=Go1Z9hiI&access_enabled=true&network_type=internet&download_type=email_download_link&ssl=true&valid_duration=period_of_time&day=07&hour=00&path=/Multimedia/Samples&file_total=1&file_name=sample01.jpg&hostname=172.17.20.62&addressee=admin@qnap.com&subject=To Share with you&content=content&include_access_code=true&link_url=https://172.17.20.62/share.cgi?ssid=0qvviuF&mail_content_date=Period of validity: 05/18/2012 09:55&mail_content_pwd=Password: Go1Z9hiI&expire_time=1337306140

Return value:

Check email response:

```
{
  "status": 1,
  "email_error": 1,
}
```

Response field:

Field	Description
status	1.(for UI reflash parameter)(donot reload web page)
email_error	1: error occur.

17.4 Get shared file list

Description:

Get whole shared file list.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_list&sid=\\${sid}&dir=\\${dir}&limit=\\${limit}&sort=\\${sort}&start=\\${start}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_list&sid=${sid}&dir=${dir}&limit=${limit}&sort=${sort}&start=${start})

Parameter name	Description
<code>\${dir}</code>	Sorting direction. ASC: Ascending , DESC: Descending
<code>\${limit}</code>	Number of response data
<code>\${sort}</code>	Sort field (filename/filesize/filetype/mt/privilege/owner/group)
<code>\${start}</code>	Response data start index

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_share_list&sid=0QNAP1234&dir=ASC&limit=20&sort=filename&start=0

Return value:

Success:

```
{
  "max_share_file": 1000
  "total": 1,
  "datas": [
    {
      "ssid": "0QNAP1234",
      "filename": "\\share\\HDA_DATA\\Public\\keyword.txt",
      "filesize": "5818",
      "filetype": 0,
      "download_link": "http://IP:8080/share.cgi?ssid=0QNAP1234",
      "start_time": 1336718447
      "expire_time": 1337323247
      "addressee": "peterlai@qnap.com",
      "creator": "admin",
      "access_code": "222",
      "download_count": 4,
      "download_limit": 0,
      "misc": "----"
    }
  ]
}
```

}

Response field:

Field	Description
max_share_file	Max share file number.
total	Total numer
datas	List datas
ssid	unique primary key
filename	File name. It must to remove "/share/HDX_DATA/" to display or do any actions.
filesize	File size. unit: byte
filetype	reserved
download_link	http://IP:8080/share.cgi?ssid=xxxxxxxxxxxxxxxxxxxx
start_time	sharing start time
expire_time	expire time
addressee	email address
creator	creator
access_code	access code
download_count	download counter
download_limit	download limitation
misc	reserved

17.5 Get hostname and external IP address

Description:

Get hostname and external IP address of the NAS.

Command:

`http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_domain_ip_list&sid=${sid}`

Return Value:

```
{
  "status": 1,
  "mycloudnas_hn": "example.mycloudnas.com",
  "ddns_hn": "example.anydns.com",
  "external_ip": "123.123.123.123",
  "local_ip": "192.168.1.101"
  "host_ip": "172.17.20.62"
}
```

Response field:

Field	Description
status	1.(for UI reflash parameter)(donot reload web page)
mycloudnas_hn	MyCloudNAS name (if the MyCloudNAS was enabled)
ddns_hn	The hostname of the DDNS service (if the DDNS service was enabled)
external_ip	External IP address of the NAS
local_ip	Local (LAN) IP address of the NAS
host_ip	host IP addresss of the NAS

17.6 Stop file(s) sharing

Description:

Stop specified file(s) sharing.

Command:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete_share&sid=\${sid}&file_total=\${file_total}&download_link=\${download_link}&filename=\${file_name}

Parameter name	Description
\${file_total}	number of total files
\${download_link}	link url
\${filename}	shared file

Return Value:

Time expire:

```
{
  "status": 1,
  "success": ture
}
```

Response field:

Field	Description
status	1.(for UI reflash parameter)(donot reload web page)
success	delete success

17.7 Stop all file sharing

Description:

Stop all file sharing.

Command:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete_share_all&sid=\${sid}

Return Value:

Time expire:

```
{  
  "status": 1,  
  "success": ture  
}
```

Response field:

Field	Description
status	1.(for UI reflash parameter)(donot reload web page)
success	delete success

17.8 Update share link

Description:

Update the attributes of specified share links.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=update_share_link&sid=\\${sid}&downdownload_type=\\${downdownload_type}&valid_duration=\\${valid_duration}&datetime=\\${datetime}&file_total=\\${file_total}&access_code=\\${access_code}&access_enabled=\\${access_enabled}&hostname=\\${hostname}&ssids=\\${ssids}&ssl=\\${ssl}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=update_share_link&sid=${sid}&downdownload_type=${downdownload_type}&valid_duration=${valid_duration}&datetime=${datetime}&file_total=${file_total}&access_code=${access_code}&access_enabled=${access_enabled}&hostname=${hostname}&ssids=${ssids}&ssl=${ssl})

Parameter name	Description
<code>\${download_type}</code>	create_download_link: create download link, email_download_link: email download link
<code>\${valid_duration}</code>	specific_time: specific the download time, period_of_time: time period, forever: forver download load
<code>\${hostname}</code>	host IP or domain name
<code>\${datetime}</code>	expire sharing time
<code>\${file_total}</code>	shared datetime if valid_duration=specific_time
<code>\${ssl}</code>	enabled ssl
<code>\${access_enabled}</code>	enable access code
<code>\${access_code}</code>	access code
<code>\${ssids}</code>	the ssid of shared files

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=update_share_link&sid=e5866gwo&downdownload_type=create_download_link&valid_duration=specific_time&datetime=2012/05/18_15:34&file_total=1&access_code=0AUDPsDc&access_enabled=true&hostname=211.75.189.169&ssids=41YzNil&ssl=false

Return value:

Check update response:

```
{  
  "status": 1,  
  "success": "true",  
}
```

Response field:

Field	Description
-------	-------------

status	1.(for UI reflash parameter)(donot reload web page)
success	ture/false

18. Recycle Bin

18.1 Get Recycle Bin Tree

Description:

Retrieve recycle bin tree list.²

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=\\${sid}&node=recycle_root](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_tree&sid=${sid}&node=recycle_root)

Parameter name	Description
<code>\${node}</code>	Recycle Bin node name (<code>\${node}=recycle_root</code>)

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=copy&sid=xxxx&node=recycle_root

Return value:

Success:

```
[ { "text": "Download", "id": "\share\HDA_DATA\Download", "cls": "w",  
  "draggable": 0 },  
  { "text": "Multimedia", "id": "\share\HDA_DATA\Multimedia", "cls": "w",  
  "draggable": 0 },  
  { "text": "Public", "id": "\share\HDA_DATA\Public", "cls": "w", "draggable":  
  0 } ]
```

Failure:

Recycle Bin option not enable.

```
{ "status": 16, "success": "true" }
```

² * for QTS 4.0 and later

18.2 Recycle Bin Recovery

Description:

Recovery specified files in recycle bin. This is an asynchronous API. The caller can use the returned process ID (pid) for further operations.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=trash_recovery&sid=\\${sid}&source_path=\\${source_path}&source_total=\\${source_total}&mode=\\${mode}&source_file=\\${source_file} ...](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=trash_recovery&sid=${sid}&source_path=${source_path}&source_total=${source_total}&mode=${mode}&source_file=${source_file} ...)

Parameter name	Description
<code>\${source_file}</code>	Name of the copied file/folder
<code>\${source_total}</code>	Total number of the recovery trash files
<code>\${source_path}</code>	Source path of the trash
<code>\${mode}</code>	1: skip, 0: overwrite
<code>\${source_file}</code>	trash file
<code>\${source_file}</code>

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=trash_recovery&sid=xxxx&source_path=/Download/@Recycle/testPath&source_total=2&source_path=/Download/@Recycle/testPath &mode=1&source_file=txt.txt&source_file=1.jpg

Recovery a file/folder named "txt.txt and 1.jpg" from trash folder to / Download/testPath. If the same file exists on the destination, do nothing.

Return value:

Success:

```
{ "status": 1, "pid": 11654 }
```

Failure:

Recovery fail.

```
{ "status": 0, "success": "true" }
```

18.3 Cancel Recycle Bin Recovery

Description:

Cancel recycle bin recovery process by process ID.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel trash_recovery&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_trash_recovery&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Pid of the recycle bin recovery process

Example:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel trash_recovery&sid=xxxx&pid=29690](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_trash_recovery&sid=xxxx&pid=29690)

Cancel the recycle bin recovery process which the pid is 29690.

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Failure:

```
{ "status": 0, "success": "true" }
```


18.4 Clean Recycle Bin

Description:

Empty the recycle bin.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=\\${sid}&path=\\${path}&file_name=\\${file_name}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=${sid}&path=${path}&file_name=${file_name})

Parameter name	Description
<code>\${path}</code>	The path folder name of recycle bin
<code>\${file_name}</code>	Same as the value <code>\${path}</code>

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=xxxx&path=/Download/@Recycle&file_name=/Download/@Recycle

Clean the folder path "/Download/@Recycle".

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```

19. Media Library

19.1 Get status of media library transcoding Queue

Description:

Retrieve the status of media library transcoding queue.

Command:

[http://IP:8080/utilRequest.cgi?sid=\\${sid}&func=video_ml_queue&op=\\${op}&subop=\\${subop}](http://IP:8080/utilRequest.cgi?sid=${sid}&func=video_ml_queue&op=${op}&subop=${subop})

Parameter name	Description
<code>\${op}</code>	1: transcoding queue lists
<code>\${subop}</code>	0: wait status 1: finish status 2: error 3: ongoing 4: all

Example:

http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_ml_queue&op=1&subop=4

Return value:

ERROR:

Error if transcode server not enable:

```
{ "status": 18, "success": "true" }
```

SUCCESS:

```
{ "status": 1, "total": 2, "percentage": 200, "datas": [  
  { "id": "1", "filename": "\Download\test\20121209.mp4", "addtime":  
"89046071967999", "time": 1358735308, "status": 1, "type": 1 },  
  { "id": "2", "filename": "\Download\test\have_or_not.mp4", "addtime":  
"89046072033791", "time": 1358735335, "status": 1, "type": 1 } ] }
```

Response field:

Field	Description
status	1 : success.
total	Data list total number.
percentage	200: means server idle "percentage" value that represents the on going status transcoding file which had transcoded percentage number

id	The index number of data.
filename	The transcoding filename.
addtime	The trascode server add time
time	The media created time.
status	Same as the value of $\${subop}$
type	<i>0:auto, 1>manual</i>

19.2 Add/Delete Media Library Transcoding Queue

Description:

Add or delete files in the media library transcoding queue.

Command:

[http://IP:8080/utilRequest.cgi?sid=\\${sid}&func=video_ml_queue&op=\\${op}&subop=\\${subop}&path=\\${path}&total=\\${total}&filename=\\${filename}](http://IP:8080/utilRequest.cgi?sid=${sid}&func=video_ml_queue&op=${op}&subop=${subop}&path=${path}&total=${total}&filename=${filename})

Parameter name	Description
<code>\${op}</code>	3:add transcoding queue 5:delete transcoding queue
<code>\${subop}</code>	<code>\${op}=3</code> (add trnascoding queue) : 0: append(default) 1: preemptive <code>\${op}=5</code> (delete transcoding queue) : 0:default 1: delete all finished tasks 5: delete all tasks except finished
<code>\${mask}</code>	7:default (transcode 240,360,720 format) media transcode 1:240 format transcode only 2:360 format transcode only 4:720 format transcode only
<code>\${total}</code>	The amount of filenames.
<code>\${path}</code>	The opration path name.
<code>\${filename}</code>	filename
...	...

Example:

Add:

`http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_ml_queue&op=3&path=/Download/test&total=2&filename=Dream1.mp4&filename=test.mp4`

Return value:

SUCCESS:

```
{ "status": 1, "result": "2" }
```

Response field:

Field	Description
-------	-------------

status	1 : success.
result	Add transcoding success file numbers.

Delete:

http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_ml_queue&op=5&path=/Download
/test&total=1&filename=Dream1.mp4

Return value:

SUCCESS:

```
{ "status": 1, "result": "1" }
```

Response field:

Field	Description
status	1 : success.
result	Delete transcoding success file numbers.

Delete all:

http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_ml_queue&op=5&subop=1

Return value:

SUCCESS:

```
{ "status": 1, "result": "true" }
```

Response field:

Field	Description
status	1 : success.
result	true : Delete completed.

19.3 Get Media Library Transcode Files List

Description:

Get media library transcode files list.

Command:

[http://IP:8080/utilRequest.cgi?sid=\\${sid}&func=video_list&source_path=\\${source_path}&source_total=\\${source_total}&source_file=\\${source_file}...](http://IP:8080/utilRequest.cgi?sid=${sid}&func=video_list&source_path=${source_path}&source_total=${source_total}&source_file=${source_file}...)

Parameter name	Description
<code>\${source_path}</code>	Source file path
<code>\${source_total}</code>	The source file total number for <code>\${source_file}</code>
<code>\${source_file}</code>	Source file name
....	

Example:

`http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_list&&source_path=/Download/tes
t&source_total=2&source_file=20121209.mp4&source_file=have_or_not.mp4"`

Return value:

SUCCESS:

```
{ "status": 1, "total": 2, "datas": [  
  { "filename": "20121209.mp4", "mp4_360": 1, "flv_720": 1, "mp4_720": 1 },  
  { "filename": "have_or_not.mp4", "mp4_360": 1, "flv_720": 1, "mp4_720":  
1 } ] }
```

FAIL: File does not exist.

```
{ "status": 5, "success": "true" }
```

Response field:

Field	Description
status	1 : success.
total	Data list total number.
filename	The source file name.
mp4_360	MPEG4 360p format media code ready or not (1/0).
mp4_720	MPEG4 720p format media code ready or not (1/0).
flv_720	FLV 720p format media code ready or not (1/0).

19.4 Get Media Library Files Transcode Status

Description:

Get media library files transcode status.

Command:

[http://IP:8080/utilRequest.cgi?sid=\\${sid}&func=video_ml_status&path=\\${path}&total=\\${total}&filename=\\${filename}....](http://IP:8080/utilRequest.cgi?sid=${sid}&func=video_ml_status&path=${path}&total=${total}&filename=${filename}....)

Parameter name	Description
<code>\${path}</code>	Source file path
<code>\${total}</code>	The source file total number for <code>\${source_file}</code>
<code>\${filename}</code>	Source file name
....	

Example:

`http://IP:8080/utilRequest.cgi?sid=XXXXXXXX&func=video_ml_status&&path=/Download&total=2&filename=3.mp4&filename=4.mp4"`

Return value:

SUCCESS:

```
{ "status": 1, "total": 2, "percentage": 200, "datas": [
  { "id": "21", "filename": "\Download\3.mp4", "addtime": "89109212172287",
"time": 1359698855, "status": 1, "type": 1 },
  { "id": "22", "filename": "\Download\4.mp4", "addtime": "89109212369151",
"time": 1359698934, "status": 1, "type": 1 } ] }
```

FAIL: File does not exist.

```
{ "status": 5, "success": "true" }
```

Response field:

Same as 17.1

19.5 Delete Transcode

Description:

Delete the media transcode or image files.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete_transcode&sid=\\${sid}&path=\\${path}&file_total=\\${file_total}&file_name=\\${file_name}&mode={mode}&keep_trans=\\${keep_trans}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete_transcode&sid=${sid}&path=${path}&file_total=${file_total}&file_name=${file_name}&mode={mode}&keep_trans=${keep_trans})

Parameter name	Description
<code>\${path}</code>	Folder path.
<code>\${file_total}</code>	Total number of folder/file(s).
<code>\${file_name}</code>	Folder/file name.
<code>\${mode}</code>	1 : delete image files 2 : delete transcode video files 3 : delete image and transcode video files.
<code>\${keep_trans}</code>	0 : delete transcode files in "@Transcode" folder 1 : donnot delete transcode files in "@Transcode" folder

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=delete&sid=xxxx&path=/Public&file_total=2&file_name=folder name&file_name=file name&mode=3&keep_trans=0

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Fail:

Permission deny

```
{ "status": 4, "success": "true" }
```


19.6 Get Transcoding File Status

Description:

Get transcode file status.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_video_qstatus&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_video_qstatus&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Video transcoding process id.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=get_video_qstatus&sid=xxxx&pid=2450

Return value:

Success:

```
{ "status": 1, "filename": "aa.mp4", "running": 1 }
```

```
{ "status": 1, "success": "true" }
```

19.7 Cancel Transcode

Description:

Cancel transcoding process.

Command:

[http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_transcode&sid=\\${sid}&pid=\\${pid}](http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_transcode&sid=${sid}&pid=${pid})

Parameter name	Description
<code>\${pid}</code>	Video transcoding process id.

Example:

http://IP:8080/cgi-bin/filemanager/utilRequest.cgi?func=cancel_transcode&sid=xxxx&pid=245
0

Return value:

Success:

```
{ "status": 1, "success": "true" }
```

Fail:

```
{ "status": 0, "success": "true" }
```