



CloudMe Minimal API - 2013-03-04

**login:** [] (SOAP)  
Used to login to the system and returns loads of information, most importantly the user id, homefolder id and session token (https and digest required).

**copyDocument:** ["srcFolderId", "srcDocumentId", "srcDocument", "dstFolderId", "dstDocument"] (SOAP)  
Copies a file from one folder to another and will be named dstDocument afterwards, so that must be supplied. Supplying either id or name is enough to locate file in source folder.

**copyStream:** ["srcFolderId", "srcDocumentId", "srcDocument", "srcStreamId", "dstFolderId", "dstDocument", "dstDocumentId", "dstStreamId", "newName"] (SOAP)  
Copies the content of one stream onto another stream, not supplying streamIds or those with index one will copy content of the source file onto the other file overwriting previous content.

**createDocument:** ["folderId", "acl", "overwrite", "entry"] (SOAP)  
Creates an XML or plain-text document, the entry <atom:title> needs to be specified (otherwise the document will have no name).

**deleteDocument:** ["folderId", "documentId", "document", "streamId"] (SOAP)  
Deletes a file if streamId is not provided or is different than 1 because then it deletes the stream instead, this has not been tested though. Supplying either id or name is enough to locate file in source folder.

**deleteFolder:** ["folderId", "childFolderId"] (SOAP)  
Used to delete a folder and all its content.

**getFolderXML:** ["folderId"] (SOAP)  
GET http://{host}/v1/folders/"folderId" (REST)  
Lists a folder and all sub-folders of that folder unless they're sub-folders of another root folder.

**loadMetadata:** ["folderId", "documentId", "document"] (SOAP)  
Get information about a file, supplying either id or name is sufficient to locate file.

**moveDocument:** ["srcFolderId", "srcDocumentId", "srcDocument", "dstFolderId", "dstDocument"] (SOAP)  
Moves a file from one folder to another and will be named dstDocument afterwards, so that must be supplied. Supplying either id or name is enough to locate file in source folder.

**moveFolder:** ["srcFolderId", "srcChildFolderId", "dstFolderId", "name"] (SOAP)  
Moves the childFolder to a new folder where it will be named name, not supplying a name generates a random hex-name.

**newFolder:** ["folderId", "childFolder", "type", "listview", "xlink", "acl", "custom", "content"] (SOAP)  
Used to create a new folder in folderId with the name childFolder.

**queryFolder:** ["folderId", "query", "offset", "count", "filter", "order", "locale", "numeric", "ascending"] (SOAP)  
Used to query the database about files, returns 0 to count amount of results that matches the query and spills out all information about them.

**renameDocument:** ["folderId", "documentId", "document", "streamId", "newName"] (SOAP)  
Renames a file to newName if streamId is not provided or is different than 1 because then it renames the stream instead, this has not been tested though. Supplying either id or name is enough to locate file in source folder.

**saveMetadata:** ["folderId", "documentId", "document", "content"] (SOAP)  
Sets a files information to content, overwrites previous information-chunk so nothing from the previous information will be available unless provided in the new information-chunk.

**Upload File:** (REST)  
PUT http://{host}/v1/documents/"folderId"  
Uploads the attached file to the folder folderId.

**Get Webshares:** (REST)  
GET http://{host}/v1/users/"userId"/webshares?"parameters"  
parameters: offset, count, order, desc, name, visibility, type  
Lists brief information about all of a user's webshares, can be filtered using the parameters.

**Get Webshare:** (REST)  
GET http://{host}/v1/users/"userId"/webshares/"webshareId"  
Lists full information about a webshare, including folder(s) and/or file(s) in it.

**Create Webshare:** (REST)  
POST http://{host}/v1/users/"userId"/webshares  
Adds folder(s) and/or file(s) among the webshares

**Change Webshare:** (REST)  
POST http://{host}/v1/users/"userId"/webshares/"webshareId"  
Used to change one, all or some of the following attributes on a webshare:  
name, visibility, password, type.

**Delete Webshare:** (REST)  
DELETE http://{host}/v1/users/"userId"/webshares/"webshareId"  
Deletes the webshare.

CloudMe provides 100 API calls, but these are the most commonly used.