



FileCloud Server

Version 20.1

API Guide

Copyright Notice

© 2020 CodeLathe Technologies, Inc. All rights reserved.

No reproduction without written permission.

While all reasonable care has been taken in the preparation of this document, no liability is accepted by the authors, CodeLathe Technologies, Inc., for any errors, omissions or misstatements it may contain, or for any loss or damage, howsoever occasioned, to any person relying on any statement or omission in this document.

Any questions regarding this document should be forwarded to:

CodeLathe Technologies, Inc.
13785 Research Blvd, Suite 125
Austin TX 78750, USA
Phone: U.S: +1 (888) 571-6480
Fax: +1 (866) 824-9584
Email: support@codelathe.com

1. FileCloud Developer Guide	4
1.1 PHP Sample for FileCloud API	6
1.2 PHP Sample To Upload File With Custom Form	8
1.3 API Changes	10
1.3.1 API Change History	11
1.3.2 CSRF Token API changes	18
1.3.3 Policy Related API Changes	19
1.4 FileCloud API - "Hello World" Exercise	21
1.5 FileCloud API - Authentication Exercise	24
1.6 FileCloud API - Logging Exercise	25
1.7 FileCloud API - Requirements	26
1.8 FileCloud API - XML Responses	27

FileCloud Developer Guide



FileCloud HTTP API allows clients such as web browsers and mobile devices to programmatically connect and access FileCloud Instances running on your personal computer or any other device.

- FileCloud runs on Apache server on both Windows and Linux Platforms.
- This server handles incoming HTTP calls, transforms it into internal System Messages and sends it to the appropriate internal services.

HTTP is the fundamental protocol of the World Wide Web. HTTP is a connectionless request response protocol, meaning there is no concept of a persistent connection between a series of requests.

REQUEST AND RESPONSE

An HTTP request is a message sent from the client to the server. The server sends back a response. The request and response might contain content called the body. In addition, the response always contains a numeric response code, which lets us know if the request was successful. It gives more detailed information about what exactly happened (e.g. the cause of failure).

METHODS

HTTP supports several request methods, which help the server know how to handle the request. The HTTP methods for our purposes are:

GET: Used to retrieve a resource (such as a web page or image) from a URL

POST: Used to send data to a server (such as the content of a form) based on a URL

HEADERS

Finally, in addition to the main content of the request and response, HTTP allows additional data to be sent in the form of headers. They can be sent with the request to the server and the response from the server, and they can contain arbitrary text data. There are many standard headers, and the connection API contains methods for easily accessing some of the most common ones.

What do you want to do?

 <p>Prepare to work with the API</p>	<ul style="list-style-type: none">➔ Meet the Requirements➔ FileCloud's XML Responses➔ See a List of API Change History
 <p>Practice Working with the API</p>	<ul style="list-style-type: none">➔ Authentication Exercise➔ Logging Exercise➔ "Hello World" Exercise➔ Sample PHP File



FULL API DOCUMENTATION

FileCloud Admin API:

- <http://www.getfilecloud.com/apidocs/fcadmin/adminapi/doc/index.html>
- Swagger: <https://fcapi-admin.getfilecloud.com/>

FileCloud User API:

- <https://www.getfilecloud.com/apidocs/fcuser/userapi/doc/index.html>
- Swagger: <https://fcapi.getfilecloud.com/>

FileCloud User API (deprecated): <https://www.getfilecloud.com/apidocs/fcuser/myapp/doc/index.html>

PHP Sample for FileCloud API



Before you try this code, make sure you have installed PHP cURL,

The sample for FileCloud API "loginquest" , "getfilelist" and "upload" is done using PHP cURL

```
<?php

function getCurlValue($file, $mimetype = '') {
    // PHP 5.5 introduced a CurlFile object that deprecates the old @filename syntax
    // See: https://wiki.php.net/rfc/curl-file-upload
    if (function_exists('curl_file_create')) {
        return curl_file_create(realpath($file), $mimetype, $file);
    }

    // Use the old style if using an older version of PHP
    $value = '@' . realpath($file);

    return $value;
}

$cookie_jar = tempnam('/tmp','cookie');

// OPTIONS
$username = "USER";
$password = "PASSWORD";
$serverurl = "http://URL";
$pathval = "/SHARED/fcteamfolder/HR/Policies";
$filename = 'helloworld.txt'; // name of the file to upload
$filefullpath = 'helloworld.txt'; // Full path to file to upload

//assign post data
$params = array('userid' => $username, 'password' => $password);

$api = "loginquest";
$url= $serverurl."/core/".$api;

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$params);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_jar);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_jar);
$data = curl_exec($ch);
curl_close($ch);

$xmlstr = new \SimpleXMLElement($data);
$result = $xmlstr->command->result;
if($result == 1)
{
    echo "Profile Logged in Successfully";

    // getfilelist api call
    $api = "getfilelist";
    $path = $pathval;

    //assign post data
    $params = array('path' =>$path);

    $url= $serverurl."/core/".$api;
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_jar);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_jar);
```

```

$data = curl_exec($ch);
curl_close($ch);

$xmlstr = new \SimpleXMLElement($data);
$total = $xmlstr->meta->total;
if($total == 0)
{
    echo "</br>". "No of file's in the system";
}
else
{
    echo "</br>". "Total no. of files:". $total;
    $count = 0;
    foreach ($xmlstr->entry as $file)
    {
        echo "</br>". ++$count. ". ". $file->name;
    }
}

//upload api call
$api = "upload";
$appnamevalue = "explorer";
$pathvalue = $pathval;
$offset = '0';
$complete = '1';

        $cfile = getCurlValue($filefullpath);
        $post = array('file_contents' => $cfile);

    $url= $serverurl."/core/upload?appname=explorer" . $appnamevalue . '&path=' . $pathvalue .
'&offset=0&complete='.$complete.'&filename=' . $filename;
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_jar);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_jar);
    $data = curl_exec($ch);
    curl_close($ch);

    if($data = "OK")
    {
        echo "</br>". "File uploaded Successfully";
    }
    else
    {
        echo $data;
    }
}
else
{
    echo "Error:". $xmlstr->command->message;
}
?>

```

PHP Sample To Upload File With Custom Form



- Before you try this code, make sure you have installed PHP cURL,
- This sample uses FileCloud APIs "loginquest" and "upload"
- This code creates custom upload form doing a single shot upload
- This can be modified for multiple uploads as well
- The code can upload files with same name as input or a custom name.
- The following sample reads a file named helloworld.txt and uploads into FileCloud as hellothere.txt

```
<?php

function create_upload_form_for_datachunk($data) {
    $headers = array();
    $form[] = implode("\r\n", array(
        "Content-Disposition: form-data; name=\"filedata\"; filename=\"blob\"",
        "Content-Type: application/octet-stream",
        "",
        $data,
        "blob"
    ));

    // generate safe boundary
    do {
        $boundary = "-----" . md5(mt_rand() . microtime());
    } while (preg_grep("/{ $boundary }/", $form));

    // add boundary for each parameters
    array_walk($form, function (&$part) use ($boundary) {
        $part = "--{ $boundary }\r\n{ $part}";
    });

    // add final boundary
    $form[] = "--{ $boundary }--";
    $form[] = "";

    // set options
    $headers[] = "Content-Type: multipart/form-data; boundary={ $boundary}";
    return array($headers, $form);
}

$cookie_jar = tempnam('C:\\testfolder', 'cookie');

// Input Options
$username = "testuser";
$password = "password123";
$serverurl = "http://127.0.0.1";
$pathval = "/testuser/folder";
$filename = 'hellothere.txt'; // name of the file to upload
$filefullpath = 'C:\\testfolder\\helloworld.txt'; // Full path to file to upload
//assign post data
$params = array('userid' => $username, 'password' => $password);

//login to the site
$sapi = "loginquest";
$url = $serverurl . "/core/" . $sapi;
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_jar);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_jar);
$data = curl_exec($ch);
curl_close($ch);
//end of login
```



```

$xmlstr = new \SimpleXMLElement($data);
$result = $xmlstr->command->result;
if ($result == 1) {
    echo "Profile Logged in Successfully";

    //upload api call
    $api = "upload";
    $appnamevalue = "explorer";
    $pathvalue = $pathval;

    //creating a single shot upload call
    $offset = '0';
    $complete = '1';
    $filedata = file_get_contents($filefullpath);
    list($headers, $form) = create_upload_form_for_datachunk($filedata);

    //make the upload call
    $url = $serverurl . "/core/upload?appname=explorer" . $appnamevalue . '&path=' . $pathvalue .
    '&offset=0&complete=' . $complete . '&filename=' . $filename;
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, implode("\r\n", $form));
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_jar);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_jar);
    $result = curl_exec($ch);
    curl_close($ch);
    //end of upload call

    if ($result == "OK") {
        echo "<br>" . "File uploaded Successfully";
    } else {
        echo $result;
    }
} else {
    echo "Error:" . $xmlstr->command->message;
}

```

API Changes

- [API Change History](#)
- [CSRF Token API changes](#)
- [Policy Related API Changes](#)

API Change History

Admin API

Date	Affected API	Description	Notes
May 1, 2015	addgroup	groupemail parameter is now optional	Add's a new group to the system
May 1, 2015	unlock	Newly added API	Unlocks a folder , that was previously locked by a user
May 1, 2015	getalllocks	Newly added API	Gets all the locked path's of the system
May 1, 2015	checkadlogin	Newly added API	Check's the entered AD setup , connects properly
May 1, 2015	checkclamav	Newly added API	Check's the entered Clam AV setup, connects properly
May 1, 2015	checkldaplogin	Newly added API	Check's the entered LDAP setup, connects properly
May 1, 2015	checksendemail	Newly added API	Check's the entered email Id is correct and a test email is sent out
May 1, 2015	checksettingpath	Newly added API	Check's the entered path , exist and is writable
May 1, 2015	checkstoragesetting	Newly added API	Check's the entered path , exist and is writable
May 1, 2015	getadgroupmembers	Newly added API	Gets all the members of an AD group
May 1, 2015	getadgroups	Newly added API	Gets all the AD groups
May 1, 2015	getconfigsetting	Newly added API	Gets all the setting value of the system
May 1, 2015	getemailid	Newly added API	Gets email id configured with the AD account
May 1, 2015	getemailidforldap	Newly added API	Gets email id configured with the LDAP account
May 1, 2015	setconfigsetting	Newly added API	Set the required setting for system
May 1, 2015	cryptfsactivate	Newly added API	Activate encryption by entering the password set
May 1, 2015	cryptfsdecryptall	Newly added API	Decrypt all the files
May 1, 2015	cryptfsencryptall	Newly added API	Encrypt all the files
May 1, 2015	cryptfsinit	Newly added API	Initialize encryption in the system
May 1, 2015	cryptfsreset	Newly added API	Reset encryption setup
May 1, 2015	cryptfsstatus	Newly added API	Display's encryption status
May 1, 2015	search	Newly added API	Retrieves all the user's in the system
May 1, 2015	setadminstatus	Newly added API	Add's a user as an admin user or to remove from admin user
May 1, 2015	deleteadminuser	Removed	
May 1, 2015	superadminaddsite	Added a new (Optional) parameter "duplicatesitename"	Optional parameter to duplicate site is added , while adding a new site
May 28, 2015	exportaudit	Added parameter "startdate" and "enddate"	Now, export of audit log works only when you specify start date
Sept 19,2016	getpolicyforuser	Newly added API	Gets details list of Policies for a user
Sept 19,2016	reset2fagaforuser	Newly added API	Reset 2FA Google Authenticator secret code for a user
Sept 19,2016	updatepolicyforuser	Newly added API	Update User Policy for a user
Sept 19,2016	getstats	Newly added API	Get Statistics of the server
Sept 19,2016	import	Newly added API	Import users from CSV file
Sept 19,2016	downloadimportresult	Newly added API	Download import result

Sept 19,2016	getauthstatus	Newly added API	Get Authentication status
Sept 19,2016	getinfo	Newly added API	Get information about FileCloud Server
Sept 19,2016	getexternalindexstats	Newly added API	Get external index status
Sept 19,2016	clearrealtimeindex	Newly added API	Cleat real time index
Sept 19,2016	checkifntfshelperinstalled	Newly added API	Check if NTFS helper is installed
Sept 19,2016	checkhelperstatus	Newly added API	Check NTFS helper status
Sept 19,2016	ispathaccessible	Newly added API	Check if the given location is accessible
Sept 19,2016	isvalidsmartmountpath	Newly added API	Check if the given location is a valid smart mount path
Sept 19,2016	gettos	Newly added API	Get terms of service details
Sept 19,2016	getgroupsforuser	Newly added API	Get groups for a user
Sept 19,2016	checkstoragedb	Newly added API	Check Storage database details
Sept 19,2016	getadminlanguagelist	Newly added API	Get the list of Languages for Admin Portal
Sept 19,2016	setchecklist	Newly added API	Setup Check List
Sept 19,2016	getchecklist	Newly added API	Get Check List
Sept 19,2016	2fasuperadminlogin	Newly added API	Admin has to enter a code that has been sent to the email id configured with the account , to successfully login.
Sept 19,2016	hidegswizard	Newly added API	Hide getting started wizard
Sept 19,2016	getbackupclientsforuser	Newly added API	Gets all the backup clients for a user
Sept 19,2016	getbackupsetsforclient	Newly added API	Gets all the backup sets for a client
Sept 19,2016	getbackuprecordsforbackupset	Newly added API	Gets all the backup records for a backupset
Sept 19,2016	getserverlinkstatus	Newly added API	Gets serverlink status
Sept 19,2016	getsactivities	Newly added API	Gets all serverlink activity informations
Sept 19,2016	clearallsactivities	Newly added API	Clear all serverlink activities
Sept 19,2016	getsjournal	Newly added API	Gets all serverlink journal informations
Sept 19,2016	clearallsjournal	Newly added API	Clear all serverlink journal
Sept 19,2016	getworkflows	Newly added API	Gets all the workflow items
Sept 19,2016	getconditions	Newly added API	Gets all the Condition for workflow
Sept 19,2016	getactions	Newly added API	Gets all the actions available for a condition in workflow
Sept 19,2016	addworkflow	Newly added API	Add workflow
Sept 19,2016	updateworkflow	Newly added API	Update workflow
Sept 19,2016	removeworkflow	Newly added API	Remove workflow
Sept 19,2016	getversion	Newly added API	Get version detail of FileCloud server
Sept 19,2016	getauthenticationinfo	Newly added API	Get Authentication detail of FileCloud server
Sept 19,2016	resetpassword	Newly added API	Reset password of a user in FileCloud system
Sept 19,2016	setuserpassword	Newly added API	Set new password for a user
Sept 19,2016	getexportauditbatchlist	Newly added API	Get audit export batch list
Sept 19,2016	isdbupdateneeded	Newly added API	Check if the database is up to date
Sept 19,2016	updatedb	Newly added API	Update database
Sept 19,2016	importadgroup	Newly added API	Import AD Group and support autosync
Sept 19,2016	getsystemstatus	Newly added API	Get system status
Sept 19,2016	getrmccommands	Newly added API	Get RMC Commands
Sept 19,2016	addrmccommand	Newly added API	Add commands to the client
Sept 19,2016	removermccommand	Newly added API	Remove the commands for the client

Sept 19,2016	getrmcclients	Newly added API	Get RMC Client Details
Sept 19,2016	removermcclient	Newly added API	Remove the RMC client
Sept 19,2016	installcheck	Newly added API	Get all Install Checks in FileCloud System
Sept 19,2016	clearconfigsetting	Newly added API	Clear selected settings and set it to default
Sept 19,2016	clearallconfigsetting	Newly added API	Clear all the settings and set it to default (Reset All)
Sept 19,2016	getsharesbyowner	Newly added API	User can get share details by using share owners name
Sept 19,2016	getdoelist	Newly added API	Get the do not email id list
Sept 19,2016	removefromdoelist	Newly added API	Remove the selected email id from the list
Sept 19,2016	clearalldoelist	Newly added API	Clear the do not email id list
Sept 19,2016	getadminoperations	Newly added API	Get all the available admin operations for a admin user
Sept 19,2016	getadminuseroperationpermission	Newly added API	Get the permissions for a admin user for specific operation
Sept 19,2016	superadminlogout	Newly added API	Super Admin Logout
Sept 19,2016	superadminauthstatus	Newly added API	Super admin authentication status
Sept 19,2016	cryptfsdownloadrecoverykey	Newly added API	Download recovery key
Sept 19,2016	superadmingetconfigsetting	Newly added API	Get the Filecloud Super Admin Settings for a particular site.
Sept 19,2016	superadminsetconfigsetting	Newly added API	Set the Filecloud Super Admin Settings for a particular site.
Sept 19,2016	emptynetworkfolderrecyclebin	Newly added API	Empty all the deleted files/folders for a network share location.
Sept 19,2016	encrypts3bucket	Newly added API	Encrypt S3 Bucket
Sept 19,2016	decrypts3bucket	Newly added API	Decrypt S3 bucket
Sept 19,2016	gets3encryptionstatus	Newly added API	Get S3 encryption status details
Sept 19,2016	aborts3enctask	Newly added API	Abort S3 Encryption task
Nov 12,2016	adminlogin	Added a new header 'X-XSRF-TOKEN'	Additional anti CSRF measures for all Web browser client API actions
Jan 24,2018	addpolicy	Newly added API	Add a new policy
Jan 24,2018	copypolicy	Newly added API	Copy Policy
Jan 24,2018	getallpolicies	Newly added API	Get all the policy
Jan 24,2018	getdefaultpolicy	Newly added API	Get the Default Policy
Jan 24,2018	geteffectivepolicyforuser	Newly added API	Get effective policy for a user
Jan 24,2018	getgroupsforpolicy	Newly added API	Get the groups for policy
Jan 24,2018	getpolicyforgroup	Newly added API	Get policy for a group
Jan 24,2018	getpolicyforuser	Newly added API	Get policy for a user
Jan 24,2018	getusersforpolicy	Newly added API	Get the users for policy
Jan 24,2018	removepolicy	Newly added API	Remove policy
Jan 24,2018	resetpolicy	Newly added API	Reset a policy
Jan 24,2018	resetpolicyforgroup	Newly added API	Reset a policy for a group
Jan 24,2018	resetpolicyforuser	Newly added API	Reset a policy for a user
Jan 24,2018	updatepolicy	Newly added API	Update a policy
Jan 24,2018	getavailablereportqueries	Newly added API	Get all the available report queries
Jan 24,2018	runreportquery	Newly added API	Run a report for query
Jan 24,2018	getreports	Newly added API	Get all the available reports
Jan 24,2018	addreport	Newly added API	Add a new report
Jan 24,2018	updatereport	Newly added API	Update an existing report

Jan 24,2018	removereport	Newly added API	Remove a report
Jan 24,2018	seed	Newly added API	Import large data
Jan 24,2018	seedstatus	Newly added API	Seeding status details
Jan 24,2018	abortseed	Newly added API	Abort seeding process
Jan 24,2018	enableooediting	Newly added API	Enable office online editing
Jan 24,2018	disableooediting	Newly added API	Disable office online editing
Jan 24,2018	getoostatus	Newly added API	Get office online status
Jan 24,2018	getusersessions	Newly added API	Get the user sessions details
Jan 24,2018	removeusersession	Newly added API	Remove user session
Jun 03,2018	resetadminpassword	Newly added API	Reset admin password
Jun 03,2018	verifyresetadminpassword	Newly added API	Verify the resetted admin password
Jun 03,2018	getmetadatasetdefinitions	Newly added API	Get all the metadata set
Jun 03,2018	addmetadataset	Newly added API	Add metadata
Jun 03,2018	updatemetadataset	Newly added API	Update metadata
Jun 03,2018	deletemetadataset	Newly added API	Delete metadata
Jun 03,2018	searchuserprofiles	Newly added API	Search users

User API

Date	Affected API	Description	Notes
May 1, 2015	lock	Newly added API	Locks a file/folder
May 1, 2015	unlock	Newly added API	Unlocks a file/folder
May 1, 2015	getalllocks	Newly added API	Gets all the lock of that user
May 1, 2015	2falogin	Newly added API	2 Factor Authentication login with code and token
May 1, 2015	resend2facode	Newly added API	Resend 2 Factor Authentication code to the email
May 1, 2015	getactivitystream	Newly added API	Get all the activity of a file/folder
May 1, 2015	showhideactivity	Newly added API	Hide activity panel
May 1, 2015	addcommentforitem	Newly added API	Add comment for a file/folder
May 1, 2015	removecommentforitem	Newly added API	Remove comment from a file/folder
May 1, 2015	getcommentsforitem	Newly added API	Get all the comment for a file/folder
May 1, 2015	deleteshare	Newly added API	Deletes share
May 1, 2015	geturlforemail	Newly added API	Gets the shared URL
May 1, 2015	sendsharetoemail	Newly added API	Send the shared file/folder via email
May 1, 2015	addnotificationfilter	Newly added API	Adds Notification filter to a path
May 1, 2015	removenotificationfilter	Newly added API	Removes Notification filter to a path
May 1, 2015	isnotificationfilterset	Newly added API	Check's if Notification filter is set for a path
May 28, 2015	removefavoritelist	Newly added API	Removes Favorite list from the system
Aug 6, 2015	searchprofiles	Newly added API	Search for user to share file/folder
Aug 6, 2015	getprofilesettings	Newly added API	Get's Profile Settings details
Aug 6, 2015	createprofile	Newly added API	Create a new user
Aug 6, 2015	verifyemail	Newly added API	Verify the new user's email id
Aug 6, 2015	checkid	Newly added API	Checks whether user name is available
Aug 6, 2015	getdiskusagedetails	Newly added API	Get's storage details

Aug 6, 2015	getfolderproperties	Newly added API	Get's information of files and folders in a folder
Aug 6, 2015	getguestuserlist	Newly added API	Get's guest user list
Aug 6, 2015	updatefavoritelist	Newly added API	Update Favorite list
Aug 6, 2015	replacefavoritelist	Newly added API	Replace Favorite list
Aug 6, 2015	unsetfavorite	Newly added API	Remove an item from Favorite list
Aug 6, 2015	getfavoritesinlist	Newly added API	Get's all the item's in the favorite list
Aug 6, 2015	getfavoritesinnamedlist	Newly added API	Get's all the item's from the name of favorite list specified
Aug 6, 2015	clearfavoritesinnamedlist	Newly added API	Clear all item's from favorite specified
Aug 6, 2015	getfavoritelistforitem	Newly added API	Get's all the available favorite list for an item
Aug 6, 2015	fileexists	Newly added API	Check if the file exists in the system
Aug 6, 2015	fileinfo	Newly added API	Provides all information about a file
Aug 6, 2015	search	Newly added API	Search for file/folder in system
Aug 6, 2015	deletepartialuploads	Newly added API	Delete all partially uploaded files
Aug 6, 2015	emptyrecyclebin	Newly added API	Empty Recycle bin
Aug 6, 2015	updatesharelink	Newly added API	Update share link
Aug 6, 2015	getshareforid	Newly added API	Get all details of share by specifying the share id
Aug 6, 2015	getuseraccessforshare	Newly added API	Get permission details of a user for shared file/folder
Aug 6, 2015	getgroupaccessforshare	Newly added API	Get permission details of a group for shared file/folder
Aug 6, 2015	setgroupaccessforshare	Newly added API	Set permission details of a group for shared file/folder
Aug 6, 2015	getshareforpath	Newly added API	Get all details of share by specifying the shared file/folder path
Aug 6, 2015	searchgroup	Newly added API	Search for group to add to shared file /folder
Aug 6, 2015	getcustomheader	Newly added API	Get custom header details
Aug 6, 2015	shorten	Newly added API	Shorten the long url to share
Aug 6, 2015	getshares	Newly added API	Get all the shares in the system
Aug 6, 2015	getsharesfilter	Newly added API	Get a shares detail by specifying the path
Aug 6, 2015	getusersforshare	Newly added API	Get all the users that are added to shared file/folder
Aug 6, 2015	getprivateurlforemail	Newly added API	Get private URL for a shared file/folder
Aug 6, 2015	unsubscribe	Newly added API	Unsubscribe to not receive any email from system
Aug 6, 2015	subscribe	Newly added API	Subscribe to receive emails from system
Aug 6, 2015	share	Newly added API	Downloads publicly shared file
Aug 6, 2015	resetpassword	Newly added API	Reset password
Aug 6, 2015	deleteallversions	Newly added API	Delete all versioned files
Aug 6, 2015	getrmccommands	Newly added API	Get all Rmc command details
Aug 6, 2015	leaveShare	Newly added API	Leave a share
Aug 6, 2015	ackrmccommands	Newly added API	Acknowledge rmc command
Aug 6, 2015	docedit	Newly added API	Edit a document
Aug 6, 2015	getaccessdetailsforpath	Newly added API	Get permission a file/folder has in the system
Aug 6, 2015	getuploadform	Newly added API	Get a upload form for a publicly shared folder with upload permission
Sept 19,2016	get2fagasecret	Newly added API	Get 2FA secret code to login using Google Authenticator

Sept 19,2016	get2fagasecretqrcode	Newly added API	Get 2FA secret QR code to login using Google Authenticator
Sept 19,2016	beginprofilecreation	Newly added API	Profile creation
Sept 19,2016	getprofilesettings	Newly added API	Profile settings details
Sept 19,2016	getlanguagelist	Newly added API	Get the list of Languages
Sept 19,2016	searchcontent	Newly added API	User can perform context search for a file/folder.
Sept 19,2016	getfilelockinfo	Newly added API	Get lock details
Sept 19,2016	getemailtemplate	Newly added API	Get email template to send it to the user
Sept 19,2016	sendemail	Newly added API	Send email to the user
Sept 19,2016	getemailssubject	Newly added API	Get email subject
Sept 19,2016	requestdeviceaccess	Newly added API	Request RMC device access permission
Sept 19,2016	getrmclients	Newly added API	Get details of all the RMC clients logged in
Sept 19,2016	approvedeviceaccess	Newly added API	Approve RMC Client
Sept 19,2016	deletedevice	Newly added API	Delete RMC Device
Sept 19,2016	addaclentry	Newly added API	Add security permissions
Sept 19,2016	deleteaclentry	Newly added API	Delete security permissions on the path for a specific user or group
Sept 19,2016	deleteacl	Newly added API	Delete security permissions on the specified path
Sept 19,2016	getacl	Newly added API	Get security permission details for specified path
Sept 19,2016	setaclinheritance	Newly added API	Set security permissions on the specified path to inherit
Sept 19,2016	geteffectiveacl	Newly added API	Set security permissions on the specified path to inherit
Sept 19,2016	loginprotectedshare	Newly added API	Login to protected share
Sept 19,2016	getsharepassword	Newly added API	Get protected share password
Sept 19,2016	getrandompassword	Newly added API	Get random password for protected share
Sept 19,2016	getpublicshareauthinfo	Newly added API	Get public share authentication info
Sept 19,2016	updateshare	Added a new (Optional) parameter "sharepassword"	To update a share to a password protected share, you have to set this parameter
Nov 12,2016	logginguest	Added a new header 'X-XSRF-TOKEN'	Additional anti CSRF measures for all Web browser client API actions
Jan 24, 2018	getwopiediturl	Newly added API	Office online web editing document files
Jan 24, 2018	gettoasts	Newly added API	Get the list of Toast Notifications
Jan 24, 2018	acktoasts	Newly added API	Acknowledge the toast notifications
Jan 24, 2018	dosearch	Newly added API	User can search for a file/folder
Jan 24, 2018	cancelsearch	Newly added API	Cancel search
Jan 24, 2018	searchtermsearch	Newly added API	Specify the search term
Jan 24, 2018	clearsavedsearchterms	Newly added API	Clear the saved search terms
Jan 24, 2018	getnotificationstream	Newly added API	Get the list of notifications
Jan 24, 2018	addshortcut	Newly added API	Add Shortcut
Jan 24, 2018	deleteshortcut	Newly added API	Delete Shortcut
Jan 24, 2018	getshortcuts	Newly added API	Get list of Shortcuts
Jan 24, 2018	getshareactivityforshare	Newly added API	Get the list of share activity
Jan 24, 2018	getauditgeoipdata	Newly added API	Get the geoip data
Jan 24, 2018	search	Deprecated	
Jan 24, 2018	searchext	Deprecated	

Jun 03, 2018	getrecentshareactivityforuser	Newly added API	Get the recent share activity for user
Jun 03, 2018	getmostactiveshares	Newly added API	Get the most active shares
Jun 03, 2018	verifyoourl	Newly added API	Verify office online URL
Jun 03, 2018	getoourl	Newly added API	Get office online URL
Jun 03, 2018	getuserconsent	Newly added API	GDPR related API
Jun 03, 2018	getanonymoususerconsent	Newly added API	GDPR related API
Jun 03, 2018	privacyconsentaccept	Newly added API	GDPR related API
Jun 03, 2018	getavailablemetadatasets	Newly added API	Get all the available metadata set
Jun 03, 2018	getmetadatasetsforsearch	Newly added API	Search using metadata
Jun 03, 2018	getmetadatavalues	Newly added API	Get metadata values
Jun 03, 2018	addsettofileobject	Newly added API	Add attribute value for a metadata set
Jun 03, 2018	removesetfromfileobject	Newly added API	Remove attribute value from a metadata set
Jun 03, 2018	saveattributevalues	Newly added API	Save the attribute value

CSRF Token API changes

- [INTRODUCTION](#)
- [What has Changed?](#)
- [How to modify your existing API Setup calls:](#)
- [Disabling these checks for Backwards Compatibility:](#)
- [Conclusion:](#)

INTRODUCTION

This note on API changes affect all applications that use FileCloud API using a Web Browser user agent.

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

FileCloud 14.0 (planned for release in early January 2017) introduces API changes to add additional anti CSRF measures for all Web browser client API actions.

What has Changed?

1) When a user authenticates using a web browser, an additional cookie is now set, which is called X-XSRF-TOKEN. The value of this cookie needs to be passed in as a HTTP Header 'X-XSRF-TOKEN' for all subsequent API calls.

2) All API calls using a web browser need to also set the X-Requested-With: XMLHttpRequest HTTP header.

How to modify your existing API Setup calls:

If you are using FileCloud API using a Web Browser HTTP agent here is what you have to do,

- Add the 'X-Requested-With: XMLHttpRequest' HTTP header to all calls made via a web browser
- Retrieve all the cookies when doing a login call in admin or user portal of FileCloud
- Get the Value of 'X-XSRF-TOKEN' cookie.
- On successfully getting the value of this token 'X-XSRF-TOKEN', pass this token in the headers section , while making any POST or GET calls in admin or user portal of FileCloud.

```
Eg:
$headers = array(
    'X-Requested-With: XMLHttpRequest',
    'X-XSRF-TOKEN: ' . $this->xsrftoken,);
curl_setopt($this->curl_handle, CURLOPT_HTTPHEADER, $headers);
```

Disabling these checks for Backwards Compatibility:

To turn off CSRF security feature, open cloudconfig.php at

Windows Location: XAMPP DIRECTORY/htdocs/config/cloudconfig.php

Linux Location: /var/www/config/cloudconfig.php

and add the following to disable the checks temporarily.

```
define("TONIDOCLOUD_CSRF_CHECK", 0);
```

Conclusion:

Adding these checks makes it harder for any CSRF attempts to succeed against FileCloud installations.

Policy Related API Changes

 This change affects FileCloud versions 17.3 and greater.

Background

Prior to FileCloud 17.3, many settings and options were either set globally or overridden on a per-user basis. This is hard to manage for hundreds of users in the system and can be error-prone. To help alleviate this issue, 17.3 introduces the concept of [policies](#) that allow administrators a great deal of flexibility in managing a large FileCloud installation. Policies are a collection of rules and settings that apply to a specific user or sets of users. Any number of policies can be created by an administrator and then can be associated to a set of users individually or to set of FileCloud Groups.

What API is affected?

1. **Quota Settings for User Accounts:** If you are using "size" parameter in [updateuser](#) API to set the allowed user quota for managed storage, it is no longer supported and the API will ignore the setting and log an error message.
2. **System Config Settings:** If you are using the [setconfigsetting](#) API or [updatepolicyforuser](#) API, then following keys can no longer be set using this API and should be managed by the new Policy infrastructure

```
TONIDOCLOUD_DEFAULT_SIZE_PERUSER
TONIDOCLOUD_DISABLE_NOTIFICATION
TONIDOCLOUD_GLOBAL_SHARE_MODE
TONIDOCLOUD_ENABLE_PRIVACY_SETTINGS
TONIDOCLOUD_ENABLE_2FA
TONIDOCLOUD_2FA_DELIVERY_MODE
TONIDOCLOUD_DISABLE_INVITES
TONIDOCLOUD_DEVICE_LOGIN_CODE_MODE
TONIDOCLOUD_ADMIN_DEVICE_APPROVAL_REQUIRED
TONIDOCLOUD_ENFORCE_SESSIONTIMEOUT_DEVICES
TONIDOCLOUD_CREATE_ACCOUNT_ON_INVITE
TONIDOCLOUD_SET_ACL

TONIDOCLOUD_REQUIRE_PASSCODE_MOBILECLIENTS
TONIDOCLOUD_DISABLE_CONNECT_MOBILECLIENTS
TONIDOCLOUD_DISABLE_PRINT_MOBILECLIENTS
TONIDOCLOUD_DISABLE_DOWNLOAD_MOBILECLIENTS
TONIDOCLOUD_DISABLE_OPENWITH_MOBILECLIENTS
TONIDOCLOUD_DISABLE_SHARE_MOBILECLIENTS
TONIDOCLOUD_DISABLE_FAV_MOBILECLIENTS
TONIDOCLOUD_DISABLE_EDITS_MOBILECLIENTS
TONIDOCLOUD_DISABLE_CONFIG_CHANGES_MOBILECLIENTS
TONIDOCLOUD_APPLY_CONFIG_MOBILECLIENTS

TONIDOCLOUD_NOTIFICATION_DISABLE_ADD
TONIDOCLOUD_NOTIFICATION_DISABLE_UPDATE
TONIDOCLOUD_NOTIFICATION_DISABLE_DELETE
TONIDOCLOUD_NOTIFICATION_DISABLE_DOWNLOAD
TONIDOCLOUD_NOTIFICATION_DISABLE_PREVIEW
TONIDOCLOUD_NOTIFICATION_DISABLE_LOCK
TONIDOCLOUD_NOTIFICATION_DISABLE_SELF_NOTIFICATION
```

Handling Changes

If you are upgrading to 17.3 and if you are using the above APIs. Please note that you will need to handle it differently.

Setting Quota for Specific Users

If you want to just a specific quota for a user, you should instead

- [create a new policy](#) for that user if policy doesn't exist and attach it to the user

- (or) if policy is already attached to that user, [retrieve that policy](#)
- [set the required quota](#) in that user specific policy

Setting Default Quota Globally

If you want to set the default quota and make that change apply to all users, you should instead

- [Retrieve the current global default policy](#)
- [change the global policy's quota setting](#)

Setting specific user level settings

If you want to set specific user level settings(via `updatepolicyforuser` API) that were managed by the above configuration flags (listed in section above), you should instead

- [create a new policy](#) for that user if policy doesn't exist and attach it to the user
- (or) if policy is already attached to that user, [retrieve that policy](#)
- [set the required setting](#) in that policy

Setting Global Settings

If you want to set the global settings and make that change apply to all users, you should instead

- [Retrieve the current global default policy](#)
- [change the global policy's settings as needed](#)

FileCloud API - "Hello World" Exercise



"Hello World" for FileCloud is not exactly the program that prints Hello World.

- We will see how to login and authenticate into FileCloud in a default authentication setting.
- These calls are fundamental to understand the XML messages that are used to communicate between the FileCloud server and client.

Authentication is a one step process.

- The *loginquest* command is issued with account name and password.
- When authentication is complete and successful, the FileCloud HTTP server creates a cookie and returns the cookie value to the client through the HTTP header.

7.1 COMMAND: LOGINGUEST

loginquest command accepts the account name and password and attempts to login into the profile. The URL follows the standard format as we discussed above.

```
HTTP://HOST:PORT/CORE/OPERATION {? OPTIONAL PARAMS }
```

When applied for loginquest call, the above URL takes the form.

<http://host:port/core/loginquest>

The input parameters (see table below) userid and password must be passed as name/value pairs from client to FileCloud HTTP server as a part of HTML body through the HTTP POST method.

From the FileCloud Client, issue an HTTP POST command with the above URL when your FileCloud Development Instance is running.

When doing HTTP POST please ensure that the content-type is set correctly to "application/x-www-form-urlencoded"

Also, if you are connecting as an agent, additional parameters such as RMC details of the agent might be needed. To avoid that, we recommend spoofing the User-Agent and passing in any web browser User Agent String.

Header Parameter	Value
content-type	application/x-www-form-urlencoded
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36

FileCloud HTTP server will return an XML result in the following format.

```
<commands>
  <command>
    <type>loginquest</type>
    <result>1</result>
    <message> </message>
  </command>
</commands>
```

In the above XML response, RESULT 1 indicates that the profile was logged in successfully. As an additional help, use the FileCloud Logs to follow the communication between your client and FileCloud HTTP server.

Input HTTP, Method: POST

PARAMETER	Required	DATA TYPE	DESCRIPTION
-----------	----------	-----------	-------------

userid	Yes	String	account name
password	Yes	String	Clear Text password

Output

PARAMETER	Description
type	Name of the call. Possible value is "logginguest"
result	Returns 1 if success, 0 is failed. However, a RESULT 0 indicates the logginguest failed. Most likely cause will be password is not correct. password
message	Returns any error message generated in the call. 8 FILECLOUD

7.2 COMMAND: ADMINLOGIN

adminlogin command accepts the admin name and admin password and attempts to login into the admin site. The URL follows the standard format as we discussed above.

```
HTTP://HOST:PORT/ADMIN/?OP=OPERATION { &OPTIONAL PARAMS }
```

When applied for adminlogin call, the above URL takes the form.

```
http://host:port/admin/?op=adminlogin
```

The input parameters (see table below) op, adminuser and adminpassword must be passed as name/value pairs from client to FileCloud HTTP server as a part of HTML body through the HTTP POST method.

From the FileCloud Client, issue an HTTP POST command with the above URL when your FileCloud Development Instance is running.

When doing HTTP POST please ensure that the content-type is set correctly to "application/x-www-form-urlencoded"

We recommend spoofing the User-Agent and passing in any web browser User Agent String.

Header Parameter	Value
content-type	application/x-www-form-urlencoded
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36

FileCloud HTTP server will return an XML result in the following format.

```
<commands>
  <command>
    <type>adminlogin</type>
    <result>1</result>
    <message> </message>
  </command>
</commands>
```

In the above XML response, RESULT 1 indicates that the profile was logged in successfully. As an additional help, use the FileCloud Logs to follow the communication between your client and FileCloud HTTP server.

Input HTTP, Method: POST

PARAMETER	Required	DATA TYPE	DESCRIPTION
op	Yes	String	Operation
adminuser	Yes	String	Admin user name
adminpassword	Yes	String	Clear Text password

Output

PARAMETER	Description
type	Name of the call. Possible value is "adminlogin"
result	Returns 1 if success, 0 is failed. However, a RESULT 0 indicates the adminlogin failed.
message	Returns any error message generated in the call.

FileCloud API - Authentication Exercise



After you have successfully created a new user and logged in to FileCloud User Site, use this exercise to call the `getauthenticationinfo` operation.

- You must use the following URL: `http://www.yourdomain.com/core/getauthenticationinfo` (where `www.yourdomain.com` is your URL)
- For the purpose of this exercise, Authentication Type must be set to DEFAULT → [Enabling Default Authentication](#)



In FileCloud, the first fundamental requirement is the ability to identify the FileCloud user uniquely.

1. The first time you install FileCloud Server and log in to the Admin Portal, you will have to → [Create a new FileCloud user account](#).
2. After account creation, you login into the User Portal with that account using the password you created. → [Log in to the User Portal](#)
3. When you login correctly, an authentication cookie is sent back to the browser which is used for all subsequent HTTP calls.
4. The client has to pass this cookie back to the server with all the calls going forward to successfully communicate with the FileCloud HTTP server.
5. Usually, passing of cookies between client and server is handled automatically in the browser.
6. However, if your client does not handle this communication automatically you have to capture and pass this cookie through the HTTP header from the client to the server.



HTTP (GET/POST) response from FileCloud Server will always be in a XML format. FileCloud web server will return the following XML response.

```
<authenticationinfo>
<info>
<profile>sataad</profile>
<displayname>sataad</displayname>
<peerid>sataad</peerid>
<authenticated>1</authenticated>
<isguestauthenticated>0</isguestauthenticated>
<hash>sha1</hash>
<guesthash>sha1</guesthash>
<guesthashurl/>
<isremote>0</isremote>
<reasoncode>0</reasoncode>
<OS>TONIDO_CLOUD</OS>
<authtype>DEFAULT</authtype>
</info>
</authenticationinfo>
```

Most of the XML elements above are self explanatory, however do not worry if some properties are not clear. We will dive deep as we go through this document. Let us try to analyze the URL some more. The FileCloud HTTP server expects URL to be of a certain form.

`HTTP://HOST:PORT/CORE/OPERATION` {? OPTIONAL PARAMS VIA GET OR POST}

The "CORE" in the above URL refers to FileCloud User Site that provides programmable APIs, manages and provides HTTP connectivity. In our first example, we were using `getauthenticationinfo` as the operation that gets the basic authentication detail such as profile id, display name, authentication type etc of the profile logged into FileCloud server. The Optional Parameters can be used to pass parameters from client to server. They can either be passed via HTTP Get URL parameters or as POST parameters.

FileCloud API - Logging Exercise



FileCloud logging offers a simple but powerful means to follow and debug the communication between your client and FileCloud HTTP server.

- Logs files are stored under the SCRATCH folder under Apache web server root folder.
- To view the log file, you can use note pad or a Log file monitoring tool such as Bare Tail by Bare Metal Soft in Windows. <http://www.baremetalsoft.com/baretail/>.

As an additional activity, repeat the Authentication exercise and follow the FileCloud logs.



[FileCloud Logging](#)



[Authentication Exercise](#)

FileCloud API - Requirements









If you want to work with the FileCloud API, you will need to set up your test environment.

- You must have an instance of FileCloud Server running on a Windows or Linux server

Meet the Requirements

Before you can proceed with the practice exercises or develop a solution to work with the FileCloud API, you must complete the items in the Requirements checklist.

	Requirements	For more information
	FileCloud Server license	Request a free trial or buy FileCloud license from http://www.getfilecloud.com
	Read the release notes	 FileCloud Server Release Notes
	Read the upgrade notes if applicable	 FileCloud Server Upgrade Notes
	Install FileCloud Server	 Install FileCloud Server
	Verify you can log in to the FileCloud Admin Portal	 Log in to the Admin Portal
	Verify you can log in to the FileCloud User Portal	 Access the User Portal

FileCloud API - XML Responses



XML response received from FileCloud HTTP server follows a standard pattern with items, meta and item element.

The following is the standard XML pattern received from FileCloud server.

```
<items>
  <meta>
    <metaelement1>[Meta Element 1 value]
  </metaelement1> <metaelement2>[Meta Element 2 value]
  </metaelement2>
</meta>
<item>
  <itemdetail1>[Item detail 1 value]<itemdetail1>
  <itemdetail2>[Item detail 1 value]<itemdetail2>
  <itemdetail3>[Item detail 1 value]<itemdetail3>
</item>
</items>
```

The items and item element will be renamed appropriately for each XML response received as shown below in the samples. On the other hand, meta element is optional. Following are some of the sample XML response received.

In the loggingest response below, the items and item element are replaced with commands and command element respectively. The command element has type, result and message elements as itemdetail.

```
<commands>
  <command>
    <type>loggingest</type>
    <result>1</result>
    <message> </message>
  </command>
</commands>
```

In the getfilelist response below, the items and item element are replaced with entries and entry element respectively. The entry element has path, dirpath, name, ext, isroot, type, fullfilename, size, modified, favoritelistid, favoriteid, order, fullsize, modifiedepoch elements as itemdetail. Note that, the getfilelist also returns the optional meta element with parentpath and total entry elements.

```

<entries>
  <meta>
    <parentpath>/apptester</parentpath>
    <total>133</total>
    <realpath>/apptester/Docs</realpath>
    <canupload>1</canupload>
    <isshareable>1</isshareable>
    <candownload>1</candownload>
    <cansetacls>0</cansetacls>
    <showshareoption>0</showshareoption>
    <teamfolder>0</teamfolder>
    <result>1</result>
    <message/>
    <defaultfile/>
  </meta>
  <entry>
    <path>/apptester/Docs/CA</path>
    <dirpath>/apptester/Docs/</dirpath>
    <name>CA</name>
    <ext/>
    <fullsize>0</fullsize>
    <modified>Dec 10, 2017 8:26 AM</modified>
    <type>dir</type>
    <fullfilename>/apptester/Docs/CA</fullfilename>
    <size/>
    <modifiedepoch>1512894360</modifiedepoch>
    <modifiediso>2017-12-10T08:26:00+0000</modifiediso>
    <isroot>0</isroot>
    <isshareable>1</isshareable>
    <issyncable>0</issyncable>
    <isshared/>
    <canrename>1</canrename>
    <showprev>0</showprev>
    <canfavorite>1</canfavorite>
    <canupload>1</canupload>
    <candownload>1</candownload>
    <favoritelistid>0</favoritelistid>
    <favoriteid>0</favoriteid>
    <order>0</order>
    <showquickedit>1</showquickedit>
    <showlockunlock>1</showlockunlock>
    <showshareoption>0</showshareoption>
    <cansetacls>0</cansetacls>
    <locked>0</locked>
  </entry>
</entries>

```