The API

Quick start for PHP developers

1. Read the Authorization section bellow to understand the requirements.
2. Follow “a new client application”
3. Download and use the Web File Share PHP API Client library:https://github.com/WebFileShare/api-client

Authorization

The WebFileShare API uses the OAuth 2.0 protocol for authentication and authorization.

If you are new to OAuth2, here you can find a good article about it
here: https://aaronparecki.com/articles/2012/07/29/1/oauth2-simplified

Important note: To use the WebFileShare API, your webserver needs to be configured with a SSL certificate. The URL of the Web File Share installation needs to start with HTTPS. Unsecured HTTP connections will be refused, as it represents a serious security vulnerability. Get a free SSL certificate here: https://letsencrypt.org

Testing without SSL

Adding the following line inside /customizables/config.php would allow OAuth2 to be enabled even though you do not access the Web File Share installation via HTTPS:

$config['app']['api']['oauth2']['allow_over_http'] = true;

Warning: This disables the entire security of the API. Your Web File Share users private information will be at risk. Do not use it for production!

Adding a new client application

Before you can start using OAuth2 with your application, you’ll need to tell Web File Share a bit of information about the application. Follow these steps:

1. Login to Web File Share as superuser
2. Open the control panel and navigate to “System configuration” > “Oauth2” > “Clients”
3. Click “Add” and fill in the form
4. Web File Share will generate a “client id” and a “client secret”. Make a note of these two, as you will need to set them in your application.

Obtain an access token

Before your application can access private data using a Web File Share API, it must obtain an access token that grants access to that API. A single access token can grant varying degrees of access to multiple APIs. A variable parameter called “scope” controls the set of resources and operations that an access token permits. During the access-token request, your application sends one or more values in the “scope” parameter.

There are several ways to make this request, and they vary based on the type of application you are building. For example, a web-based application might request an access token using a browser redirect to Web File Share, while an application installed on a device that has no browser uses web service requests.

Some requests require an authentication step where the user logs in with their Web File Share account. After logging in, the user is asked whether they are willing to grant the permissions that your application is requesting. This process is called *user consent*.

If the user grants the permission, the Web File Share Authorization Server sends your application an access token (or an authorization code that your application can use to obtain an access token). If the user does not grant the permission, the server returns an error.

The authorization sequence begins when your application redirects a browser to a specific Web File Share URL; the URL includes query parameters that indicate the type of access being requested.

For web applications

This method is called in OAuth 2.0 terms “the authorization code flow”.

Authentication Endpoint URL: /oauth2/authorize/

The set of query string parameters supported by the Web File Share Authorization Server for web server applications are:

Parameter	Value	Description
response_type	code	Determines whether the Web File Share OAuth 2.0 endpoint returns an authorization code. Web server applications should use code.
client_id	The “client id” you obtain from the Web File Share control panel	Identifies the client that is making the request. The value passed in this parameter must exactly match the value shown in the Web File Share Control Panel
redirect_uri	One of the “redirect uri” values listed for this application	Determines where the response is sent. The value of this parameter must exactly match one of the values listed for your application in the Web File Share control panel, including the http or https scheme, case, and trailing '/').
scope	Space-delimited set of permissions that the application requests.	Identifies the Web File Share API access type that your application is requesting.
state	Any string	Provides any state that might be useful to your application upon receipt of the response. The Web File Share Authorization Server roundtrips this parameter, so your application receives the same value it sent. To mitigate against cross-site request forgery (CSRF), it is strongly recommended to include an anti-forgery token in the state, and confirm it in the response.

An example request URL is shown below, with line breaks for readability.

```
https://www.your-site.com/WebFileShare/oauth2/authorize/?
  scope=email%20profile&
  state=SOME-RANDOM-DATA&
  redirect_uri=https%3A%2F%2Fwww.your-app.com%2Fdo-something-with-the-code&
  response_type=code&
  client_id=f9c6f82cb3e872a20e6a310f33a9c450
```

You web application will be redirecting the users to a similar URL. Web File Share then handles the user authentication and consent. The result is an authorization code, which your application can exchange for an “access token” and a “refresh token”.

Handling the response

The response will be sent to the “redirect_uri” as specified in the request URL. If the user approves the access request, then the response contains an authorization code and the state parameter (if included in the request). If the user does not approve the request, the response contains an error message.

Important: if your response endpoint renders an HTML page, any resources on that page will be able to see the authorization code in the URL. Scripts can read the URL directly, and all resources may be sent the URL in the Referer HTTP header. Carefully consider if you want to send authorization credentials to all resources on that page (especially third-party scripts such as social plugins and analytics). To avoid this issue, we recommend that the server first handle the request, then redirect to another URL that doesn't include the response parameters.

Getting the access token

After your web application receives the authorization code, it should exchange it for an access token and a refresh token, by making an HTTP POST request to the following URL:

Token Endpoint URL: /oauth2/token/

Parameters:

Parameter	**Description**
code	The authorization code returned from the initial request.
client_id	The “client id” obtained from the Web File Share control panel
client_secret	The client secret obtained from the Web File Share control panel.
redirect_uri	One of the redirect URIs listed for this project in the
grant_type	As defined in the OAuth 2.0 specification, this field must contain a value of “authorization_code”.

A successful response to a request contains the following fields:

Parameter	Description
access_token	The token that needs to be sent to the Web File Share API for a regular request.
refresh_token	A token that may be used to obtain a new access token. Refresh tokens expire in 30 days.
expires_in	The remaining lifetime of the access token. Access tokens expire in 60 minutes.
token_type	Identifies the type of token returned. At this time, this field will always have the value Bearer.

Here's how an example response looks like:

```
{
"access_token":"PJIeg5uIs31JBmTGmcUFap6Gv2xhJQs84IqetJeL",
"token_type":"Bearer",
"expires_in":3600,
"refresh_token":"Sj5267kclpjhrvT0pdcE8mVbYxoZTu3u8flqg5cY"
}
```

The application should store the refresh token for future use and use the access token to access the Web File Share API. Once the access token expires, the application uses the refresh token to obtain a new one.

For installed applications

This method is called in OAuth 2.0 terms the “resource owner credentials flow”. It is also known as the “password” flow.

Desktop and mobile application, if they cannot redirect the user to the Web File Share URL for authentication and providing consent, they usually just prompt the users for their Web File Share username and password.

The process requires just a direct HTTP POST call to the token endpoint (/oauth2/token/), with the following parameters:

Parameter	Description
username	The Web File Share user account username.
password	The Web File Share user account password.
scope	Space-delimited set of permissions that the application requests. Identifies the Web File Share API access type that your application is requesting. Each API method that your application will be using requires a certain scope. See that further down in the documentation.
client_id	The “client id” obtained from the Web File Share control panel
client_secret	The client secret obtained from the Web File Share control panel.
redirect_uri	One of the redirect URIs listed for this application inside the Web File Share control panel.
grant_type	As defined in the OAuth 2.0 specification, this field must contain a value of “password”.

Please see the above section Getting the access token for handling the response.

Note: This type of authorization is protected against brute force attacks, just as the regular Web File Share login. If yo File Share user account will get deactivated.

Example

```
curl -X POST -d "username=john&password=love123&scope=upload&client_id=WebFileShare00000000000000000000Mobile&client_se
uri=http://localhost&grant_type=password" https://demo.WebFileShare.co/oauth2/token/
```

- `john` and `love123` - are the Web File Share account's username and password
- `WebFileShare00000000000000000000Mobile` - is the the default API client id used by the mobile apps. It is recommend application.
- `00000000000000000NoSecret0000000000000000` - the API client secret
- `http://localhost` - one of the API configured redirect URLs for the particular API client
- `https://demo.WebFileShare.co` - the URL of your WebFileShare installation

Refreshing the access token

As access tokens expire, you will need to get fresh one once in a while. You do that by making a HTTP call to the follo

Refresh Token Endpoint URL: **`/oauth2/token/`**

Parameters:

Parameter	Description
client_id	The “client id” obtained from the Web File Share control panel
client_secret	The client secret obtained from the Web File Share control panel.
grant_type	As defined in the OAuth 2.0 specification, this field must contain a value of “refresh_token”.

Parameter	Description
refresh_token	The refresh token you have received along with the access token.

A successful response to a request will be identical to the response you receive when you are requesting an initial acc

Note: Save refresh tokens in secure long-term storage and continue to use them as long as they remain valid.

Calling the Web File Share API with the access token

After your application obtains an access token, you can use the token to make calls to the Web File Share API on beha
token in a request to the API by including the “Authorization: Bearer” HTTP header.

Example:

```
GET /WebFileShare/api.php/account/info HTTP/1.1
Authorization: Bearer 8vDeNtzJ8Nf1P0fH1YsvIubOMGttXpqOmupl3oD1
Host: www.your-site.com
```

Where “`8vDeNtzJ8Nf1P0fH1YsvIubOMGttXpqOmupl3oD1`” is the access token received on the previous step.

For most API calls, the server reply will contain a JSON object in the response body. Successful requests will have a pr
For failed requests, the “success” value will be set to “false” and the “error” property will be populated with an textua
supposed to provide information, such as attaching a web link to a file, the property “data” will be populated if the op

Access tokens are valid only for the set of operations and resources described in the scope of the token request. For e
listing directory contents (scope=list), it cannot be used for accessing the user's profile information (scope=profile). Y
WebFileShare API multiple times for similar operations.

Access tokens have limited lifetimes (around 1 hour). If your application needs access to the Web File Share API beyo
obtained refresh token to get a new access token.

API methods

Getting user account information

Target URL	**`/api.php/account/info`**
Required scope	profile
HTTP Method	GET/POST
Output format	JSON

Retrieving lists of files and folders

Target URL	**`/api.php/files/browse/`**
Required scope	list
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Default value	Required	Description

Parameter	Type	Default value	Required	Description
path	string		Yes	Examples: /ROOT - shows a list with items like “My Files”, “Shared with me”, “Starred” (the lis future) / - same as above /ROOT/HOME - items located inside the users home folder (My Files) /STARRED - starred items /PHOTOS - latest photos /MUSIC - latest audio files /SHARES - items shared by the user /LINKS - items shared through web links /ROOT/SHARED - users with shares or folders shared anonymously by other users /ROOT/123 - lists folders shared by user with ID 123. /ROOT/123/456 - list items inside the share with ID 456 owned by user with ID 123.
itemType	string		Yes	Choose type of items to list. Possible values: any - lists both files and folders files - lists only files folders - lists only folders
recursive	boolean	false	No	List items from all the subfolders.
details	array		No	Allows you to choose what information should be retrieved for each file.
details[uuid]	array key		No	unique id which can be used for referencing the file or folder
details[mdate]	array key		No	modified date
details[mdateHuman]	array key		No	modified date in a friendly format
details[cdate]	array key		No	creation date
details[hasWebLink]	array key		No	if file has weblink attached to it or not
details[weblink]	array key		No	retrieve weblink URL
details[weblink-full]	array key		No	retrieve full weblink details
details[description]	array key		No	file type description
details[ext]	array key		No	file extension
details[type]	array key		No	type of file (defined inside system/data/filetypes.php)
details[icon]	array key		No	filename of the Web File Share icon associated with this type of files
details[hasThumb]	array key		No	shows if Web File Share can generate a thumbnail for the file
details[fileSize]	array key		No	includes the file size in bytes
details[nicerFileSize]	array key		No	includes formatted file size
details[commentsCount]	array key		No	includes number of attached user comments
details[label]	array key		No	includes files labels
details[isLocked]	array key		No	shows if file is locked
details[version]	array key		No	includes current file version
details[isShared]	array key		No	shows if folder is currently shared

Example

Listing only files from the users home folder, retrieving information about their attached weblinks and also including a

```
path=/ROOT/HOME
itemType=files
details[[]]=nicerFileSize
details[[]]=weblink
```

path=/ROOT/HOME - the users home folder

itemType=files - listing only files

details[]=nicerFileSize - including a formatted filesize

details[]=weblink - including the URL, if a weblink is attached

Expected output:


```
{
  "success":true,
  "error":false,
  "data":{
    "meta":{
      "path":"\/ROOT\/HOME",
      "parentPath":"\/ROOT",
      "folderName":"Home Folder",
      "perms":{
        "upload":true,
        "download":"1",
        "alter":true
      }
    },
    "files":[
      {
        "filename":"WebFileShare_Admin_Guide.pdf",
        "weblink":"http:\/\/demo.WebFileShare.com\/wl\/?id=89M",
        "is_dir":false,
        "nicerFileSize":"123 KB"
      },
      {
        "filename":"WebFileShare_License_Agreement.pdf",
        "is_dir":false,
        "nicerFileSize":"116 KB"
      },
      {
        "filename":"WebFileShare_User_Guide.pdf",
        "is_dir":false,
        "nicerFileSize":"195 KB"
      },
      {
        "filename":"Welcome.jpg",
        "is_dir":false,
        "nicerFileSize":"17 KB"
      }
    ]
  }
}
```


Retrieving metadata

Target URL	**/api.php/files/metadata/**
Required scope	metadata
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Default value	Required	Description
path	string		Yes	Examples: /ROOT/HOME/file.ext - retrieves metadata for a file named `file.ext` available in the Wel folder

Searching files and folders by name

Target URL	**/api.php/files/search/**
Required scope	list
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Default value	Required	Description
path	string		Yes	Path relative to the user's home folder.
keyword	string		Yes	The keyword to search the file names for.
details	array		No	The same as as for the task above.

Creating folders

Target URL	**/api.php/files/createfolder/**
Required scope	upload
HTTP Method	POST/GET

Request Parameters Reference

Parameter	Type	Description
path	string	Web File Share path of the new folder's parent.
name	string	Name of the new folder.

Uploading files

Target URL	**/api.php/files/upload/**
Required scope	upload
HTTP Method	PUT
Output format	JSON

Request Parameters Reference

Parameter	Type	Description

Parameter	Type	Description
path	string	The Web File Share path of the target file.

Example

```
curl -X PUT --header "Authorization: Bearer neY6uAjKO1KqQh98RZZ5DOgYjIPMuu9duvvHGUiN" -T your-file.ext https://demo.WebFileSha
my-file.ext
```

- `neY6uAjKO1KqQh98RZZ5DOgYjIPMuu9duvvHGUiN` - is the previously received “access_token”
- `your-file.ext` - is the path of the file you want to upload from the local computer
- `https://demo.WebFileShare.co` - is the URL of your Web File Share installation
- `/ROOT/HOME/make-new-folder/my-file.ext` - is the remote path where you wish the file to be uploaded. Web File Share already exist.

Downloading files

Target URL	**/api.php/files/download/**
Required scope	download
HTTP Method	GET/POST
Output format	HTTP DOWNLOAD

Request Parameters Reference

Parameter	Type	Description
path	string	The Web File Share path of the file.

Downloading thumbnails

Target URL	**/api.php/files/thumbnail/**
Required scope	download
HTTP Method	GET/POST
Output format	HTTP DOWNLOAD

Request Parameters Reference

Parameter	Type	Description
path	string	The Web File Share path of the file.

Renaming files or folders

Target URL	**/api.php/files/rename/**
Required scope	modify

HTTP Method	GET/POST
Output format	HTTP DOWNLOAD

Request Parameters Reference

Parameter	Type	Description
path	string	
newName	The new file/folder name.	

Deleting files or folders

Target URL	**/api.php/files/delete/**
Required scope	delete
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Description
path	string	The Web File Share path of the target file.
permanent	boolean (1/0)	Either the file should be permanently removed, instead of just moved to the trash folder.

Starring files or folders

Target URL (add)	**/api.php/files/star/**
Target URL (remove)	**/api.php/files/unstar/**
Required scope	modify
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Description
path	string	The Web File Share path of the target file/folder.

Create web links on files and folders

Target URL	**/api.php/files/weblink/**
Required scope	weblink

HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Description
path	string	The Web File Share path of the target file/folder.
singleDownload	boolean	Returns a link which is valid for a single download. This does not affect web links the user might have previously created file/folder.
temporary	boolean	Returns a link which is valid for 15 minutes. This does not affect web links the user might have previously created on the

Example reply:

```
{
  "success": true,
  "error": false,
  "data": {
    "status": "created", //can also return "existing"
    "url": "http:\/\/www.yoursite.com\/WebFileShare\/wl\/?id=CtmsT8lWoen3JDZlVbxvR3SH45gvvvxs",
    "isdir": false //or true if you are linking a folder
  }
}
```

Sharing folders

Target URL	**/api.php/files/share/**
Required scope	share
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Required	Description
path	string	Yes	The Web File Share path of the folder.
uid	integer	Yes if no “gid”	ID of Web File Share user to share folder with.
gid	integer	Yes if no “uid”	ID of Web File Share group to share folder with.
anonymous	boolean	No	Specify if folder is to be shared anonymously.
upload	boolean	No	Specify if upload permission is granted.
download	boolean	No	Specify if download permission is granted.
comment	boolean	No	Specify if the permission to post comments is granted.
read_comments	boolean	No	Specify if the permission to read comments is granted.
alter	boolean	No	Specify if the permission to make file changes is granted.
share	boolean	No	Specify if the permission to share files is granted.
alias	string	No	Specify an alias for the shared folder name.

Note: If the folder was already shared, the share settings will be updated. No errors will be returned in that case.

Unsharing folders

Target URL	**/api.php/files/unshare/**
Required scope	share
HTTP Method	GET/POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Required	Description
path	string	Yes	The Web File Share path of the folder.
uid	integer	Yes if no “gid”	ID of Web File Share user to be removed from the share.
gid	integer	Yes if no “uid”	ID of Web File Share group to be removed from the share.

Note that the call will return an error if the folder is not shared with the specified user or group.

Get Web File Share user account information

Target URL	**/api.php/admin-users/info**
Required scope	admin
HTTP Method	POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Required	Description
UID	integer	Yes, if `uname` not provided	User ID
uname	string	Yes, if `UID` not provided	Username

Add Web File Share user accounts

Target URL	**/api.php/admin-users/add**
Required scope	admin
HTTP Method	POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Default value	Required	Description

Parameter	Type	Default value	Required	Description
generate_password	boolean		No	Set to 1 to have Web File Share assign a randomly generated password
create_home_folder	boolean		No	Set to 1 to have Web File Share create the user's home folder if it doesn
data[username]	string		Yes	The username may not contain special characters, except for underscor
data[name]	string		Yes	
data[password]	string		No	
data[two_step_enabled]	boolean	0	No	
data[two_step_secret]	string		No	
data[last_pass_change]	date	NULL	No	
data[owner]	integer	NULL	No	This can be the ID of the parent independent admin user.
data[registration_date]	date	current date	No	
data[activated]	boolean	1	No	
data[expiration_date]	date	NULL	No	
data[require_password_change]	boolean	0	No	
data[email]	string		No	
data[receive_notifications]	boolean	0	No	
data[company]	string		No	
data[website]	string		No	
data[description]	string		No	
data[logo_url]	string		No	
perms[role]	integer	NULL	No	
perms[admin_type]	string	NULL	No	Possible values: `simple` , `indep`
perms[admin_users]	boolean	0	No	
perms[admin_roles]	boolean	0	No	
perms[admin_notifications]	boolean	0	No	
perms[admin_logs]	boolean	0	No	
perms[admin_metaperms]	boolean	0	No	
perms[admin_over]	mixed		No	Set to “-ALL-” if the user is an admin who can manage all other users
perms[admin_max_users]	boolean	0	No	
perms[admin_homefolder_template]	string		No	
perms[homefolder]	string		Yes	The is an absolute path to a folder existing in the server's file system. A servers.

Parameter	Type	Default value	Required	Description
perms[space_quota_max]	integer	0	No	
perms[space_quota_current]	integer	0	No	
perms[traffic_quota_max]	integer	0	No	
perms[traffic_quota_current]	integer	0	No	
perms[readonly]	boolean	0	No	
perms[upload]	boolean	1	No	
perms[download]	boolean	1	No	
perms[download_folders]	boolean	1	No	
perms[read_comments]	boolean	0	No	
perms[write_comments]	boolean	0	No	
perms[email]	boolean	0	No	
perms[weblink]	boolean	0	No	
perms[share]	boolean	0	No	
perms[btsync]	boolean	0	No	
perms[metaperms]	boolean	0	No	
perms[file_history]	boolean	0	No	
perms[users_may_see]	string	-ALL-	No	
perms[change_pass]	boolean	1	No	
groups	array		No	A list of group names. If groups with the specified names are not found,

Example response

Example response after successful request:

```
{
  "success": true,
  "error": false,
  "data":{
    "generated_password": "12345678",
    "uid": "44"
  }
}
```

Where “44” is the ID of the newly created user account and “12345678” is the password generated by Web File Shar

Example response after failed request:

```
{
  "success": false,
  "error": "The value of data[username] needs to be unique in the database",
  "code": "username_in_use"
}
```


Modify Web File Share user accounts

Target URL	**/api.php/admin-users/edit**
Required scope	admin
HTTP Method	POST
Output format	JSON

Request Parameters Reference

Besides the parameters described higher, for adding user accounts, this API method uses also the following:

Parameter	Type	Required	Description
UID	integer	Yes	The user ID

Delete Web File Share user accounts

Target URL	**/api.php/admin-users/delete**
Required scope	admin
HTTP Method	POST
Output format	JSON

Request Parameters Reference

Parameter	Type	Required	Description
UIDS	array	Yes	Array of user ID integers
deleteHomeFolder	boolean	No	If included, this will cause the user(s) home folders to also be deleted.

Revoking app authorization (for users)

Users can see the authorizations made for the various apps, inside the “Account Settings” and can revoke them from

Troubleshooting

"Check the "access_token" parameter"

If you cannot get past the error *“The request is missing a required parameter, includes an invalid parameter value, in malformed. Check the “access_token” parameter.”*, although you have checked and your HTTP request includes the “ perhaps PHP doesn't get the variable “ `$_SERVER['HTTP_AUTHORIZATION']` ” populated. In which case, if you are running A “ `.htaccess` ” file:

```
RewriteEngine On
RewriteCond %{HTTP:Authorization} .+
RewriteRule .* - [[E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]]
```

If you are using a virtual host, make sure the above is inside the Virtualhost tag, not in Directory tag.

Table of Contents

Revision #1
★Created 22 December 2021 14:22:24 by Mahesha Damayanthi
Updated 24 December 2021 17:45:57 by Mahesha Damayanthi