

PowerShell APIs Samples V1.0

Table of Content

1.	Getting Started	3
2.	Configuration Settings	4
3.	Samples	7

1. Getting Started

The API gives you programmatic access to the Cloudiway service. Our API is based on REST principles making it easy for you to develop and test applications.

We provide a full PowerShell library with all standard operations. You'll have access to samples for each product. The goal of this guide is to make it easier for you to start using our APIs and have a better overview of the process. Please note that the complete APIs document can be found on our site.

The provided zip file contains the followings:

Classes and Functions Folder: this folder contains all the classes and function definition

MainScript.ps1: this is the main script that you'll be running and updating based on what you're trying to achieve.

You need to have a valid account on <https://portal.cloudiway.com> and create a project.

Please note that this guide intends to just provide you with samples that you can use and update depending on your objectives. You can find the Swagger definition of our public APIs here: <https://api-production.cloudiway.com/index.html>

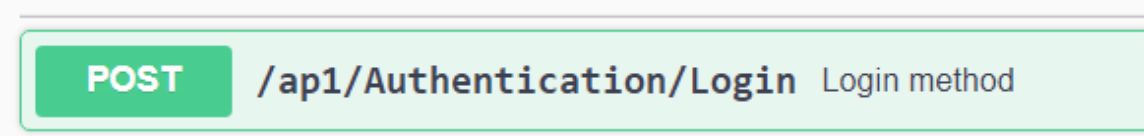
2. Configuration Settings

The first step to get you started is to authenticate using your credentials and retrieve the Bearer Token. You'll need to enter your Cloudiway credentials

```
##### GET CONFIG SETTINGS #####
$global:HOSTURL = 'https://api-production.cloudiway.com'
$userId = 'Name@cloudiway.com'           #Your Login
$userPass = 'YourPassword'              #Your Password

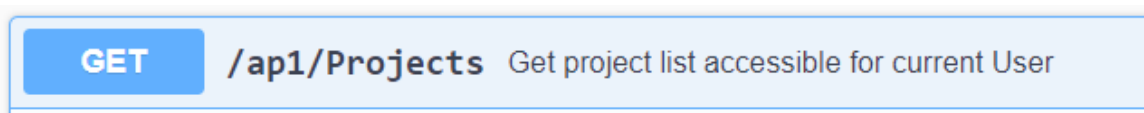
# Login
$global:BEARERTOKEN = Get-BearerToken $userId $userPass
```

The **Get-BearerToken** function will retrieve and return the token that will be used later. This function invokes the following API:



POST /ap1/Authentication/Login Login method

Once you have a valid account on the platform, you'll be able to create a project. Each project has its unique ID which we'll be using to invoke any of the sample APIs. In the main script we provide a function called **GetProjects**. It will invoke this API using the bearer token:



GET /ap1/Projects Get project list accessible for current User

This will return the list of projects that you have access to with their corresponding IDs which will be used later. Example of running **"GetProjects"**:

```

27
28     #Run this to get the list of projects and their corresponding IDs
29     GetProjects
30
31
32
33
34

```

PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
		<pre> id : 260 collectionId : 183 name : teeeest creationDate : 2020-09-29T14:16:23.463Z lastAccessDate : 2021-01-20T10:35:45.06Z datacenterLocation : technicalLocation : id : 354 collectionId : 183 name : Google Groups creationDate : 2021-02-26T09:53:03.883Z lastAccessDate : 2021-02-26T09:53:35.81Z </pre>	

Once you identify the ID of the project you want to work on, you'll need to store it in the global variable PROJID:

```
#Identify the project you want to work on and insert the projectId here
$global:PROJID = 713
```

The project ID is required to invoke any APIs within the project. Depending on your level of permission the API will return an error if you don't have enough privileges. Example: You're a viewer on the project Google Groups (ID 354) and you're trying to delete a connector or a user. The API will return a 403 Forbidden. It also applies when you're trying to invoke an API within a project you don't have permissions on.

Once your project is created, you'll need to set-up your connectors. We do not recommend creating the connectors with the APIs. One thing about creating connectors with the API is that there are many validity checks that occur on the front-end only – not in the API, so it's possible that you could create some invalid connectors.

You'll need to identify your source and target pool IDs to perform any future migration tasks. This API returns the list of available connectors within your specific project:

GET	/ap1/Connectors Get all connectors
------------	-------------------------------------------

It can be invoked using the function “**GetConnectors**” as below:

```
38 #Run this to get the list of connectors with their corresponding Pool IDs
39 GetConnectors
40
41
42 |
43
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Get Connectors
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Connectors

id          : 2
poolId      : 4
connectorTypeId : 2
name        : Target Lab
isTarget    : True
rawSettings : {"TenantName":"Cloudiwaylab2","RestClientId":"dce47e37-f5ea-47c9-b362-9b55c071e690","RestClientSecret"
```

It returns the list of connectors with their respective IDs and Pool IDs.

From there you can identify your source and target pool IDs and get started with your migration. You’ll need to store the pool IDs in the global variables “**SPool**” and “**TPool**”:

```
#Identify your Pools IDs
$global:SPool = 3
$global:TPool = 4
```

3. Samples

The following section covers the provided API samples for the product File. The same methodology can be followed for the other products.

3.1. Get User List

To get the list of users within your project you can invoke the following APIs:

GET /ap1/File Gets the list of users from file_list table

Or

POST /ap1/File/Displayable Get files with status and batch

We chose to provide a sample for displaying the File_List using the Post/ap1/File/Displayable since it is more generic and provide more information in the response(batch, status...). You can invoke this API when running the function “GetFileUserList”:

```
Get File Users List
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/Displayable

id      : 1
firstName : tt
lastName : tt
sourceEmail : test@source.com
targetEmail : test@target.com

.....ENDING MAIN SCRIPT .....
```

The ID of the user is a unique identifier and is used to invoke many APIs. It is also referred to as the “ObjectID”.

Example of an API where the ObjectID is required and passed in the payload:

GET /ap1/File/FileStatistics/{objectId} GetFileStatistics

3.II. Create a File User

The “**CreateFileUser**” function is a sample for creating a File user where the source is a Google Drive and the target is a OneDrive. It invokes the following API:



The parameters are the followings:

Parameters:

\$FName	String	First name of the user to be created
\$LName	String	Last name of the user to be created
\$SEmail	String	Source email of the Google Drive/ OneDrive
\$TEmail	String	Source email of the Google Drive/ OneDrive
\$SPool	Integer	Id of the source Pool that you want to assign to your user
\$TPool	Integer	Id of the target Pool that you want to assign to your user
\$TRecipient	Integer	1 (OneDrive), 2(SharePoint)

Example of running “**CreateFileUser**”:

```

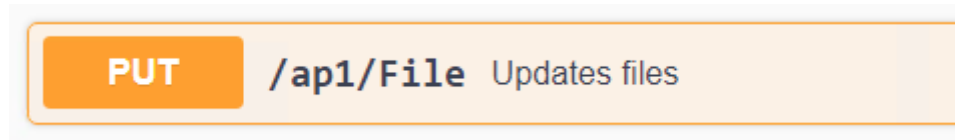
52 # ##### CROD #####
53 # param ([string] $FName, [string] $LName,[string] $SEmail,[string] $TEmail, [int] $SPool, [int] $TPool
54 | CreateFileUser 'FName' 'LName' 'test@domain.com' 'TEmail@domain.com' $global:SPool $global:TPool 1
55 # GetFileUserList
56 # DeleteFileUser 9
57 # UpdateFileUser 5 'FName' 'LName' 'SEmail@domain.com' 'TEmail@domain.com' 2 3 1 # TargetRecipient

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
. 'c:\Users\aidir\source\repos\TestFrontAPI\ps1\SampleAPIsScript\MainScript.ps1' -configFile MyConfigFile.xml
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Create File User
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File

id                : 10
firstName         : FName
lastName          : LName
sourcePoolId      : 3
targetPoolId      : 4
sourceEmail       : test@domain.com
targetEmail       : TEmail@domain.com
sourceFolder      :
  
```


3.III. Update File User

The “**UpdateFileUser**” function is a sample for updating an existing user within your project. It invokes the following API:



The parameters are the followings:

Parameters:

Parameter	Type	Description
\$ID	Integer	Id of the user to update
\$FName	String	First name of the user to be created
\$LName	String	Last name of the user to be created
\$SEmail	String	Source email of the Google Drive/ OneDrive
\$TEmail	String	Source email of the Google Drive/ OneDrive
\$SPool	Integer	Id of the source Pool that you want to assign to your user
\$TPool	Integer	Id of the target Pool that you want to assign to your user
\$TRecipient	Integer	1 (OneDrive), 2(SharePoint)

Example of running “UpdateFileUser”:

```

55 # GetFileUserList
56 # DeleteFileUser 9
57 UpdateFileUser 10 'FName' 'LName' 'SEmail@domain.com' 'TEmail@domain.com' $global:SPool $global:TPool 1
58
59 # ##### File Logs and Stats #####
60 # GetFileUserBySourceEmail 'test@domain.com'

```

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
. 'c:\Users\aidir\source\repos\TestFrontAPI\ps1\SampleAPIScript\MainScript.ps1' -configFile MyConfigFile.xml
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Update File User
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/10

id                : 10
firstName         : FName
lastName         : LName
sourcePoolId     : 3
targetPoolId     : 4
sourceEmail      : SEmail@domain.com
targetEmail      : TEmail@domain.com
sourceFolder     :

```

3.IV. Delete File User

The “**DeleteFileUser**” function is a sample for deleting an existing user within your project. It invokes the following API:

DELETE /ap1/File/{objectId} Deletes a File

The parameters are the followings:

Parameters:

\$ID	Integer	Id of the user to update
------	---------	--------------------------

Example of running “**DeleteFileUser**”:

```

56 DeleteFileUser 10
57 # UpdateFileUser 10 'FName' 'LName' 'SEmail@domain.com' 'TEmail@domain.com' $global:SPool $global:
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
. 'c:\Users\aidir\source\repos\TestFrontAPI\ps1\SampleAPIsScript\MainScript.ps1' -configFile MyConfigFile.xml
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Delete File User: 10
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/10
True
.....ENDING MAIN SCRIPT .....
  
```

3.V. Get File User

The “**GetFileUserBySourceEmail**” function is a sample That can be user to get the ID of a user within your project. It invokes the following API:

POST /ap1/File/FileUserSearch Search files with file Search object

The parameters are the followings:

Parameters:

\$SEmail	String	Source email of the Google Drive/ OneDrive
----------	--------	--------------------------------------------

Example of running “**GetFileUserBySourceEmail**”:

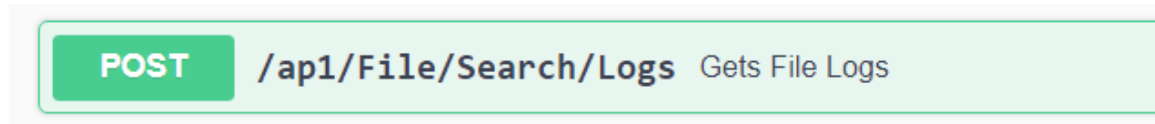
```

58
59 # ##### File Logs and Stats #####
60 GetFileUserBySourceEmail 'test@domain.com'
61 # GetFileUserLogs 5
62 # GetFileUserStats 5]
63
64
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
. 'c:\Users\aidir\source\repos\TestFrontAPI\ps1\SampleAPIsScript\MainScript.ps1' -configFile MyConfigFile.xml
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Get File User: test@domain.com
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/FileUserSearch

fileUser      : @{id=11; firstName=Test; lastName=l; sourcePoolId=3; targetPoolId=4; sourceEmail=test@domain.com;
                targetEmail=Target@domain.com; sourceFolder=; sourceRecipientType=0; targetRecipientType=1;
                targetChannelName=; targetlibrary=; targetfolder=; sku=0; isLocked=False; timeStamp=2021-12-31T09:16:23.32Z;
                preProcessed=False; targetSiteName=; size=0}
jobStatus     : 0
jobType       : 0
hasSpecificSettings : False
batchName     :
  
```

3.VI. Get File User Logs

The “**GetFileUserLogs**” function is a sample to display the logs for a specific user. It invokes the following API:



The parameters are the followings:

Parameters:

\$ID	Integer	Id of the user
------	---------	----------------

Example of running “**GetFileUserLogs**”

```

61 | GetFileUserLogs 11
62 # GetFileUserStats 5
63
-----
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
-----
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Get FileUser Logs
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/logs/11
.....ENDING MAIN SCRIPT .....
timeStamp      message
-----
2021-12-31T09:17:08.9403908Z ** Starting job **
2021-12-31T09:17:11.6376974Z Audit of test@domain.com for connector type : GSuite
2021-12-31T09:17:11.6880934Z Starting to list all files and folders from the source at 12/31/2021 09:17:11
2021-12-31T09:17:58.0631524Z Audit Error for test@domain.com : Google Api Error One or more errors occurred. (Error:"unauthorized_...
  
```

3.VII. Get File User Stats

The “**GetFileUserStats**” function is a sample to display the statistics for a specific user. It invokes the following API:



The parameters are the followings:

Parameters:

\$ID	Integer	Id of the user
------	---------	----------------

Example of running “GetFileUserStats”

```
62 GetFileUserStats 11
63
64
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Get FileUser Stats
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/FileStatistics/11

startDateTime : 2021-12-31T09:17:58.237Z
fileTotal     : 0
fileSuccess   : 0
fileFailure   : 0
fileIgnored   : 0
folderTotal   : 0

.....ENDING MAIN SCRIPT .....
```

3.VIII. Start File Jobs

The “**StartFileJobs**” function is a sample to start jobs for a list of users. It invokes the following API:

POST
/ap1/Jobs/StartJobs/{productType}/{jobTypeId} Start Jobs

The parameters are the followings:

Parameters:

\$JobType	Integer	Job to start: 24 (Audit), 26 (PreProcessing), 27 (Migration)
\$SubjectsId	Array of Integer	ObjectIDs to invoke the API for. To be passed in the Body

Example of running “StartFileJobs”

```

64
65 # ##### File Jobs #####
66 | StartFileJobs 24 "[10]" # JobType: 24 (Audit), 26 (PreProcessing), 27 (Migration)
67 # StopFileJobs "[5]"

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

..... STARTING MAIN SCRIPT

Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login

Start Jobs:

Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Jobs/StartJobs/6/24

```
{
  "isSuccess": true,
  "message": "StartJobs done",
  "count": 0,
  "responseData": null,
  "error": [
  ]
}
```

In this example we started an Audit for the user with the ID 10.

3.IX. Stop File Jobs

The “**StopFileJobs**” function is a sample to stop jobs for a list of users. It invokes the following API:

POST
/ap1/Jobs/StopJobs/{productType} Stop Jobs

The parameters are the followings:

Parameters:

\$SubjectsId	Array of Integer	ObjectIDs to invoke the API for. To be passed in the Body
--------------	-------------------------	--------------------------------------------------------------

Example of running “**StopFileJobs**”

```

67 StopFileJobs "[10]"
68
69 # ##### File Batches #####
70 # GetListFileBatches

```

PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE

```

..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Stop Jobs:
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Jobs/StopJobs/6
{
  "isSuccess": true,
  "message": "StopJobs done",
  "count": 1,
  "responseData": true,
  "error": [
    ]
  }

```

3.X. Get Batches

The “**GetListFileBatches**” function is a sample to get the list of batches within your project.. It invokes the following API:

GET /ap1/File/Batch Gets the list of Batches

```
68 |
69 # ##### File Batches #####
70 | GetListFileBatches
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
. 'c:\Users\aidir\source\repos\TestFrontAPI\ps1\SampleAPIsScript\MainScript.ps1' -cc
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
GetListBatches
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/File/Batch
Batch Name:
Batch1
Batch ID:
1
# Members in the Batch:
1

Batch Name:
Batch2
Batch ID:
```


3.XI. Start Batch Jobs

The “**StartFileBatchJobs**” function is a sample to start jobs for a specific batch. It invokes the following API:

POST /ap1/Jobs/batch/{batchId}/{jobTypeId}/{productType} Batch new jobs

The parameters are the followings:

Parameters:

\$JobType	Integer	Job to start: 24 (Audit), 26 (PreProcessing), 27 (Migration)
\$BatchId	Integer	Id of the batch to start the jobs for

Example of running “StartFileBatchJobs”

```

70 # GetListFileBatches
71 | StartFileBatchJobs 24 1 #JobType: # 24 (Audit), 26 (PreProcessing), 27 (Migration)
72 # StopFileBatchJobs 1
73
74 # End of file

```

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Start Batch Jobs:
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Jobs/batch/1/24/6
{
  "isSuccess": true,
  "message": "StartBatch done",
  "count": 0,
  "responseData": null,
  "error": [
  ]
}
.....ENDING MAIN SCRIPT .....

```

In this example we started an Audit for the user with the ID 1.

3.XII. Stop Batch Jobs

The “**StopFileBatchJobs**” function is a sample to stop jobs for a specific batch. It invokes the following API:

POST /ap1/Jobs/batch/{batchId}/{productType} Stop Batch

The parameters are the followings:

Parameters:

\$BatchId	Integer	Id of the batch to start the jobs for
-----------	---------	---------------------------------------

Example of running “**StopFileBatchJobs**”:

```

72 StopFileBatchJobs 1
73
74
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
..... STARTING MAIN SCRIPT .....
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Authentication/Login
Start Batch Jobs:
Invoke-FrontAPI https://api-production.cloudiway.com/ap1/Jobs/batch/1/6
{
  "isSuccess": true,
  "message": "StopBatch done",
  "count": 0,
  "responseData": true,
  "error": [
  ]
}
.....ENDING MAIN SCRIPT .....

```