

API documentation

Paris Mou Data Exchange service API
This documentation describes version 1.0 of the DES API.
Status: concept
Date : february 11,2021
Author: Marcel Doorn

Introduction

The Paris Mou Data Exchange Service (DES) API is a REST API using GET requests to obtain files or information from the DES server. This document describes which methods are at your disposal and how to use them.

Please be informed that a PHP library is available that wraps most of the REST functionality into convenient functions.

In general a request to the DES server is made by using the following URL format:

https://fileservier.parismou.org/api/<AUTHORIZATION_TOKEN>/<METHOD_NAME>/ANY/PARAMETERS

With the exception of file requests, all requests will return a JSON object in response.

Response JSON object

The response JSON object always has the following keys:

- status
- meta

Depending on the request method, any number of other keys may be used.

status

code : either "error" or "success"

message : "Message explaining the error (if any)"

meta

request_path : The request path

request_time : Request time as an integer

request_time_hr: Request time (human readable)

sourceip : The IP making the request

api

version : DES API version number

Obtaining an authorization token

Since every file server transaction requires authentication through an authorization token, you'll most always start with an authentication token request.

The URL for this request is the same as the one above but instead of an authorization token you have to include your API key.

https://fileservier.parismou.org/api/<API_KEY>/getauthorizationtoken

In return you'll receive a JSON object like above supplemented with a `access_token` key, for example:

status

code : either "error" or "success"

message : "Message explaining the error (if any)"

meta

request_path : The request path

request_time : Request time as an integer

request_time_hr: Request time (human readable)

sourceip : The IP making the request

api

version : DES API version number

access_token : vWTNEQm1EUjN6MHU5MFpramVEN0FOU1hkSmRod210WU9OV1o0OH

An authorization token (access token) is valid till 10 minutes after retrieval.

API key

You can find your Paris Mou API key either with your account data on the website or by contacting Paris Mou.

Public methods

- **getauthorizationtoken**
 - **getfilelist**
 - **getfile**
-

getauthorizationtoken

Retrieves an authorization token. This method requires an API key.

Arguments

No arguments required.

Returns

The JSON object has a `access_token` key holding the authorization token.

Example

https://fileserver.parismou.org/api/<API_KEY>/getauthorizationtoken

getfilelist

Retrieves an array of all files available on the file server.

Arguments

No arguments required.

Returns

The JSON object has a `files` key holding the array of file names.

Example

https://fileserver.parismou.org/api/AUTHORIZATION_TOKEN>/getfilelist

getfile

Downloads a single file.

Arguments

File name.

Returns

A file.

Example

<https://fileserver.parismou.org/api/AUTHORIZATION> TOKEN>/getfile/SomeFile_20201102_0682.xml.zip

Examples

Example 1. Downloading a single file

```
<?php

// First get an authorization key
$apiKey = 'test12345';
$apiUrl = 'https://fileserver.parismou.org/api';
$url = $apiUrl . '/' . $apiKey . '/getauthorizationtoken';

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);

$responseDecoded = json_decode($response);

if ($responseDecoded->status->code == 'success') {

    $authorizationToken = $responseDecoded->access_token;
}

// Secondly get the file
$fileName = 'test.zip';
$localFilePath = '.';
$url = $apiUrl . '/' . $authorizationToken . '/getfile/' . $fileName;

file_put_contents($localFilePath . '/' . $fileName, file_get_contents
($url));
```

Example 2. Updating local folder with files from the DES server

```
<?php

$apiKey = 'test12345';
$apiUrl = 'https://fileserver.parismou.org/api';
$url = $apiUrl . '/' . $apiKey . '/getauthorizationtoken';
```

```

$localFilePath = './localFiles';

// Make an array of locally present files
$localFilesArray = [];

foreach (scandir($localFilePath) as $file) {
    if ($file != '.' && $file != '..') {
        $localFilesArray[] = $file;
    }
}

// Make an array of files present on the DES server
// First get an authorization key
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch) ;
curl_close($ch);

$responseDecoded = json_decode($response);

if($responseDecoded->status->code == 'success') {

    $authorizationToken = $responseDecoded->access_token;
}

// Secondly get a list of all files on the DES server
$url = $apiUrl . '/' . $authorizationToken . '/getfilelist' ;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);

$responseDecoded = json_decode($response);
$remoteFilesArray = [];

if ($responseDecoded->status->code == 'success') {

    $remoteFilesArray = $responseDecoded->files;
}

// Get files that are not locally present
$diffArray = [];

foreach ($remoteFilesArray as $file) {
    if (!in_array($file, $localFilesArray)) {
        $url = $apiUrl . '/' . $authorizationToken . '/getfile/' .
$file;
        file_put_contents($localFilePath . '/' . $file,

```

```
file_get_contents($url) ;  
    }  
}
```