

Uploading a file on "Google Drive"

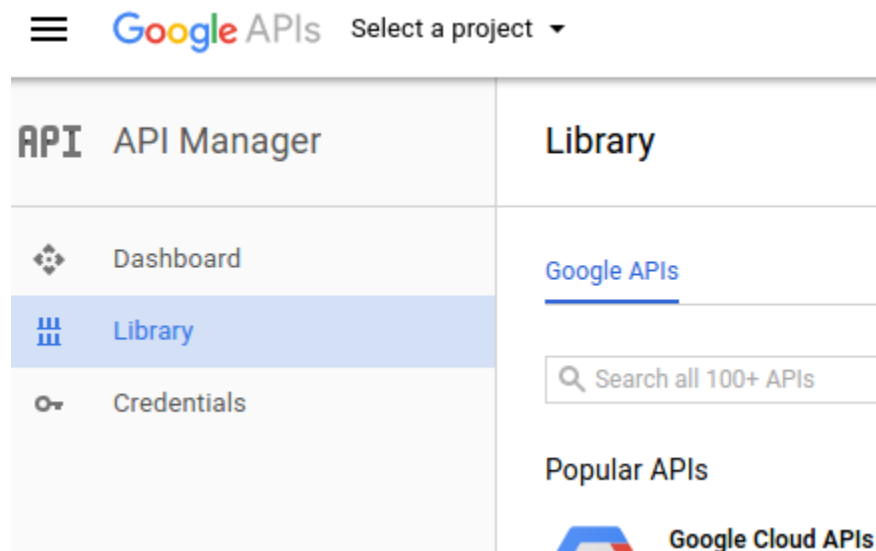
using "Google OAuth Token"

Using PHP

- We are implementing file upload on Google drive by using Google's Client Library.
- Google also provides a PHP client library — it takes care of all the APIs that Google provides.
- We use Google+ API and Google Drive API Library to get Auth and Access Token and then using Access token we can upload file on drive from directly our web application.

1. Creating a Google App

1. Go to [Google API Console](#)
2. If you have not created a project, create a project by clicking "**Select a project**" (at the top), and then clicking on the "+" button in the dialog box. In the next screen enter your project name, and agree with T&C.

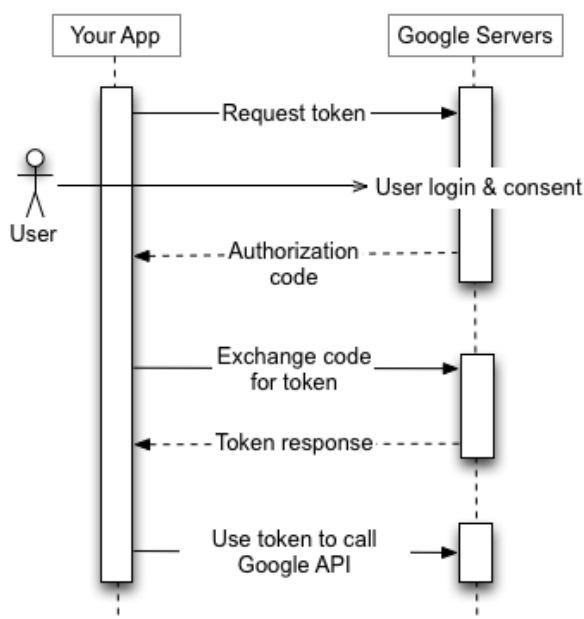


3. After the project is created, select the created project from the top dropdown.
4. Click the **Library** tab on the left. Search for "**Google+ API**" and enable it. By enabling "**Google+ API**", your Google application can get user information such as id, name, picture, email etc. And also Search for **Google Drive API** and Enable both for our use.

5. Now click on **Credentials** tab on the left. In the next screen click on "**OAuth consent screen**". Fill out the mandatory fields. Save it.
6. Now click on the "**Credentials**" tab (just beside "**OAuth consent screen**"). In the screen, click on "**Create credentials**". Choose "**OAuth Client ID**" as the type.
7. In the next screen fill out the name. The Application type should be "**Web application**"
8. Add a redirect url in the section **Authorised redirect URIs**. This url should point to your redirect url script.
9. Click on the **Create** button.
10. On success you will get the App Client ID and App Secret. Download this credential json file to your computer they will be required later.

2. Basic Understanding of the Process

- First, you need to register your application as an API Project in Google's API Console, and enable the "Google+ API" and "Drive API" services
- You place a link in your HTML code that will redirect to Google's servers for login (called the OAuth login url).
- You also provide a redirect url, Google will redirect the user to this url after he does the login. Google will also pass an authorization code to this redirect url.
- It is in this redirect url script that you must use the passed authorization code (which is available as a GET parameter). This authorization code can be exchanged for an access token from Google (you have to implement an API call to get the access token from the authorization code).
- You can then use the access token to upload file on google drive and also get user information such as id, name, picture, email etc. For upload file you need to use Google Client Library.



1. Send client authentication request via Google OAuth API.
2. Google OAuth server validates against its backend and returns authentication code.
3. The client executes callback PHP script.
4. Application callback script sends the authentication code for getting the access token.
5. Google API responds to the client with success or failure token.
6. On success use this token to upload a file on Google Drive.

3. Implement to login and upload

1. Create the project directory
2. execute the following command in your project root to install this library

composer require google/apiclient:"^2.2.2"

3. If failed in step 2 check this link to work [composer installation instructions](#).
4. If failed in step 1 and 2 then follow this direct download link [Download Here](#)
 - i. Open the link
 - ii. Go to the title **V2.2.2**
 - iii. Download this file **google-api-php-client-2.2.2.zip**
 - iv. Unzip the file and copy **src** and **vendor** directory and past in root directory of your application.
5. Done with Google Client Library place the credential json file download from google console api in 1. Creating google app section and rename this file to **"oauth-credentials-file.json"**
6. Create file **validation.php** that contain the some validation of this application
 - i. This file contain we write 3 simple function
 1. **setHeader** : using this function we add basic html header with basic initial html codes
 2. **getOAuthCredentialsFile** : In this function we read the credential json file from our root directory that you download from google api console.
 3. **missingOAuth2CredentialsWarning** : Using this method we give error message in case of if credential json file not found or invalid formate.
 4. Check the code for 3 in below image to better understand with validation.php file

```

1 <?php // Set page headers function
2 function setHeader($title)
3 {
4     // Start the session for storing access tokens
5     if (!headers_sent()) {
6         session_start();
7     }
8     $header_data = "<!doctype html>
9     <html>
10    <head>
11        <title>" . $title . "</title>
12        <link href='styles/style.css' rel='stylesheet' type='text/css' />
13    </head>
14    <body>\n";
15    $header_data .= "<a href='file-upload-demo.php'>Back to Home</a>&nbsp;&nbsp;&nbsp;";
16    if(isset($_SESSION['upload_token'])){
17        $header_data .= "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href='file-upload-demo.php?logout=1'>Logout</a>";
18    }
19    $header_data .= "<header><h1>" . $title . "</h1></header>";
20    return $header_data;
21 }
22
23 function missingOAuth2CredentialsWarning()
24 { // Function to print error on missing credentials
25     $response_data = "
26     <h3 class='warn'>
27         Warning: You need to set the location of your OAuth2 Client Credentials from the
28         <a href='http://developers.google.com/console'>Google API console</a>.
29     </h3>
30     <p>
31         Once downloaded, move them into the root directory of this repository and
32         rename them 'oauth-credentials-file.json'.
33     </p>";
34
35     return $response_data;
36 }
37 function getOAuthCredentialsFile()
38 { // get the credentials from credentials file
39     $oauth_creds = 'oauth-credentials-file.json';
40     if (file_exists($oauth_creds)) {
41         return $oauth_creds;
42     }
43     return false;
44 }

```

7. Create file **file-upload-demo.php** that contain the logic to Login with Google, get access token and upload file on Google Drive.

- i. Top of the file include the **autoload.php** from **vendor** directory and **validation.php** that we create in previews step.

```

include_once 'vendor/autoload.php';
include_once "validation.php";

```

- ii. Next call that function **setHeader** to setup basic html and header for css with 1 parameter of page title.

```

echo setHeader("Uploading a file");

```

- iii. Check for credential json file using validation.php file's function

```
// check for oauth credentials
if (!$oauth_credentials_file = getOAuthCredentialsFile()) {
    echo missingOAuth2CredentialsWarning();
    return;
}
```

- iv. Now create Google client class object and set login config and then create Google service class object with Google Client class object.

```
// set redirect URI is to the current page
$redirect_uri = 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'];

// set config for google OAuth with google client library
$client = new Google_Client(); // create instant of google client
$client->setAuthConfig($oauth_credentials_file); // set credentials
$client->setRedirectUri($redirect_uri); // set redirect url
$client->addScope("https://www.googleapis.com/auth/drive"); // set scope of access
$service = new Google_Service_Drive($client); // set google client object with google service
```

- v. Get the authorization code and then use it to send a POST request in order to obtain an access token from the Google Authorization Server
 1. Use this code to get access token save it to session and redirect to self URL or any other from where you want to upload file (we here redirect to self URL because we upload file from same file).

```
if (isset($_GET['code'])) {
    // get access token
    $token = $client->fetchAccessTokenWithAuthCode($_GET['code']);
    $client->setAccessToken($token);

    // store in the session also
    $_SESSION['upload_token'] = $token;

    // redirect back to the example
    header('Location: ' . filter_var($redirect_uri, FILTER_SANITIZE_URL));
}
```

- vi. Set the access token as part of the client to upload file on Google Drive.
 1. Check session and set auth token as part of the client. If session is empty then create auth URL to login and get auth code.

```
if (!empty($_SESSION['upload_token'])) {
    $client->setAccessToken($_SESSION['upload_token']);
    if ($client->isAccessTokenExpired()) {
        unset($_SESSION['upload_token']);
    }
} else {
    $authUrl = $client->createAuthUrl();
}
```

- vii. Upload a file using any php file upload function. Here we use **move_uploaded_file** function to upload file with validation of file extension, file size etc.

```
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
    && $imageFileType != "gif" && $imageFileType != "JPG" && $imageFileType != "pdf" && $
    imageFileType != "doc" && $imageFileType != "docx") {
    $errorMsg = "Sorry, only JPG, JPEG, PNG, GIF, PDF & DOC files are allowed.";
    $uploadOk = 0;
}
// check for file size limit
else if ($_FILES["fileToUpload"]["size"] > 500000) {
    $errorMsg = "Sorry, your file is too large.";
    $uploadOk = 0;
}
// move file to our directory
else if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
    DEFINE("DEMOFILE", $target_file);
    DEFINE("DEMOFILENAME", $_FILES["fileToUpload"]["name"]);
} else {
    $uploadOk = 0;
    $errorMsg = "Sorry, there was an error uploading your file.";
}
```

- viii. IF file upload successfully to your local memory then time to upload this file on Google Drive using **Google_Service_Drive_DriveFile**
1. Create **Google_Service_Drive_DriveFile** object.
 2. Use **setName("filename");** to give name of the file you can use any name if you want to rename file before upload.
 3. Call the **create()** function to upload file.
 4. Use results to access uploaded file by id and name.

```
// Now lets try and send the metadata as well using multipart!
$file = new Google_Service_Drive_DriveFile();
$file->setName(DEMOFILENAME); // DEMOFILENAME is Constant define file name
$result2 = $service->files->create(
    $file,
    array(
        'data' => file_get_contents(DEMOFILE), // DEMOFILENAME is Constant define file path
        'mimeType' => mime_content_type(DEMOFILE), // 'image/jpeg',
        'uploadType' => 'multipart'
    )
);

// Output after upload
echo "<a href='https://drive.google.com/open?id='. $result2->id <?>= $result2->name ?></a>";
```

4. HTML File and UI of this Process

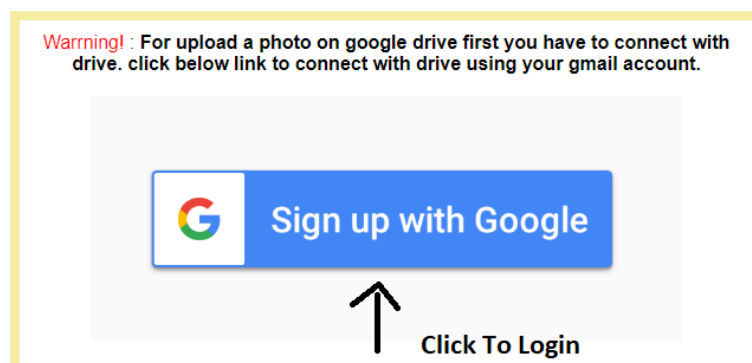
1.Main File HTML Content

```
<div class="box">
<?php if (isset($authUrl)): ?>
  <div class="">
    <center><span class="warn">Warning!</span> : <strong>For upload a photo on google drive first you have to connect
      with drive. click below link to connect with drive using your gmail account.</strong></center><br>
    <a class='login' href='<?=$authUrl ?>'></a>
    </div>
  </div>
<?php elseif($_SERVER['REQUEST_METHOD'] == 'POST'): ?>

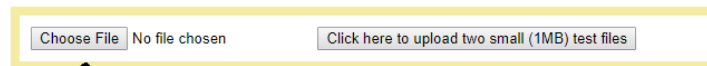
  <?php if ($uploadOk == 0){ ?>
    <div>
      <p class="warn"><?=$errorMsg; ?></p>
      <a href='file-upload-demo.php'>Try with diffrent file</a>
    </div>
  <?php }else{ ?>
    <div>
      <p>Your call was successful! Check your drive for the following files:</p>
      <ul>
        <li><a href="https://drive.google.com/open?id=<?=$result2->id ?>" target="_blank"><?=$result2->name ?></a></li>
      </ul>
      <a href='file-upload-demo.php'>Upload more files</a>
    </div>
  <?php } ?>
<?php else: ?>
  <form method="POST" enctype="multipart/form-data">
    <input type="file" name="fileToUpload" required="required">
    <input type="submit" name="submit" value="Click here to upload two small (1MB) test files" />
  </form>
<?php endif ?>
</div>
```

2. UI

Uploading a file on "Google Drive" using "Google OAuth Token"

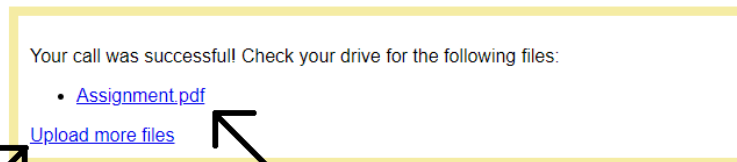


Uploading a file on "Google Drive" using "Google OAuth Token"



Choose and Upload file to drive

Uploading a file on "Google Drive" using "Google OAuth Token"



Upload more files

Click to View Uploaded File

More Reference

<https://developers.google.com/drive/api/v3/quickstart/php>

<https://github.com/google/google-api-php-client/releases>