Contents

[**Introduction** 2](#_Toc39933226)

[**OAuh 2.0 Framework** 3](#_Toc39933227)

[**OAuth 2.0 Framework Advantage and Disadvantage** 4](#_Toc39933228)

[Advantages of OAuth 2.0 4](#_Toc39933229)

[Disadvantages of OAuth 2.0 4](#_Toc39933230)

[**Developing Web Application Using Google OAuth 2.0** 5](#_Toc39933231)

[Step 1: Create project 7](#_Toc39933232)

[Step 2: Give project name 7](#_Toc39933233)

[Step 3: Select user type as External 8](#_Toc39933234)

[Step 4: Give application name 8](#_Toc39933235)

[Step 5: Create credentials 9](#_Toc39933236)

[Step 6: Select application type and give redirect URL 9](#_Toc39933237)

[Step 7: You can see your client ID & your secret ID. 9](#_Toc39933238)

[**Final PHP Application** 10](#_Toc39933239)

[OAuth Login Page 10](#_Toc39933240)

[Login Gmail Account 10](#_Toc39933241)

[Provide Password For Gmail 11](#_Toc39933242)

[Token Retrieve from Google 11](#_Toc39933243)

[Logged User Information 12](#_Toc39933244)

[Database Details 12](#_Toc39933245)

[**Appendixes** 13](#_Toc39933246)

[PHP Code 13](#_Toc39933247)

**Introduction**

OAuth is an open standard access delegation method which mainly used for given permission for users to access application or website without giving them passwords. Recently IOT devices are very popular and share their services and resources between IoT domain and it becomes a huge challenge. Though OAuth 2.0 framework has some limitations in provide the capacity of computer systems or software to exchange and make use of the information. The standard leaved several mechanisms somewhat or totally undefined. Therefore large organization like Google, Facebook, and Microsoft implements OAuth 2.0 with different authorization capabilities and used token access mechanisms to fulfill interoperability.

OAuth implementations began in 2006 by Blaine Cook. The OAuth 1.0 protocol was published under RFC5849 in April 2010. OAuth 1.0 has recorded session fixation security drawback issue and OAuth 2.0 introduced in October 2012. OAuth 2.0 in not backward compatible and it has specific authorization flows for web application, Desktop application, android and smart devices.

In this report will demonstrate OAuth 2.0 protocol structure, Different roles in OAuth and advantage and disadvantage of OAuth 2.0 protocol as well as developing web application using Google OAuth 2.0 framework.

**OAuh 2.0 Framework**

OAuth is the essential technology that used for website authentication by sites. It allow users to access defined resources of the application using their associated account such as Facebook, Microsoft, LinkedIn, Google, GitHub. For an example, a user can clicks on the Google login option when logging into another website, Google authenticates them, and the original website logs them in using permission obtained from Facebook. Four roles has defined to OAuh 2.0

* Owner of the resource

Resource owner is the one who authorize to enter their resources that allocated to that particular account. Application access rights will define as a scope.

* Client

The client application is the one which wants to access the users account. User authorization will require by user before validate by the API.

* Resources Server

The resource server is the host which is hold secured user profiles

* Authorization Server

The authorization server will validates the identity of the user and it will issues the access tokens to the application.

**OAuth 2.0 Framework Advantage and Disadvantage**

Current context is world famous social media providers like Facebook, Google, LinkedIn, and Twitter they are providing OAuth 2.0 facility for 3rd party applications. OAuth 2.0 framework has following features and corns and pros.

**OAuth 2.0 Features**

* It is a very simple protocol which is provides the facility to access resources without haring passwords.
* It delivers the user agent flow for run the client’s application using a scripting language, such as JavaScript. Typically, a browser can be the user agent.
* It access the data using tokens

## Advantages of OAuth 2.0

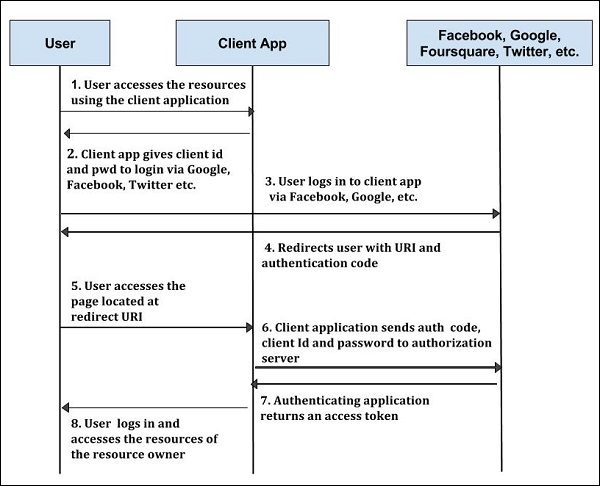
* OAuth 2.0 is flexible protocol and all data transmission happen through SSL to provide secure tokens.
* Ensure cryptography industry protocols
* It gives user to best experience while saving time. They no need to register with several website to perform the task. One single sign on can facilitate to the shared data of the users without having the personal information.

## Disadvantages of OAuth 2.0

* When we use several redirects of applications such as Facebook, Google, then we have to write lot of codes for specific logins.
* Security issue like phishing - asking users to login to a different site really similar Google or Facebook but they are taking user credentials providing fake url.
* If one account get hacked every site will infected and user sensitive information on risk.

**Developing Web Application Using Google OAuth 2.0**

Now you should have an idea about OAuth 2.0 protocal. Now we try to figure out how exactly authentication happen between client and server and how server issues token for client. Following figure 1 demonstrate how client getting google token to access application services behalf of google server authentication.



* The client requests an entree token only with the help of client authorizations.
* The client credentials authorization flow is used to acquire the access token to authorize API requests.
* Using client credentials authorization, access token which is acquired, only grants permission for your client application to search and get catalog documents.

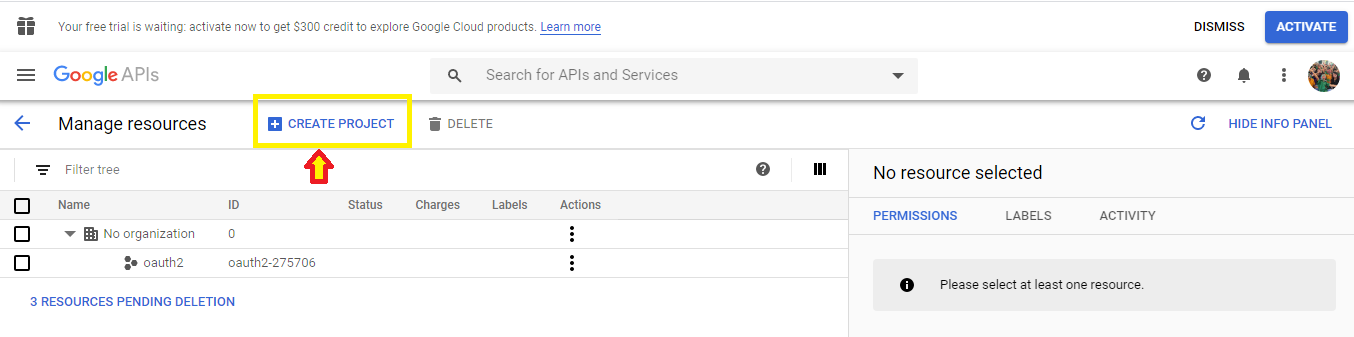


* **client\_id** − It should be set to the client id of your application.
* **redirect\_uri** − It should be set to the URL. After the request is authorized, the user will be redirected back.
* **response\_type** − It can either be a code or a token. The code must be used for server side applications, whereas the token must be used for client side applications. In server side applications, you can make sure that the secrets are saved safely

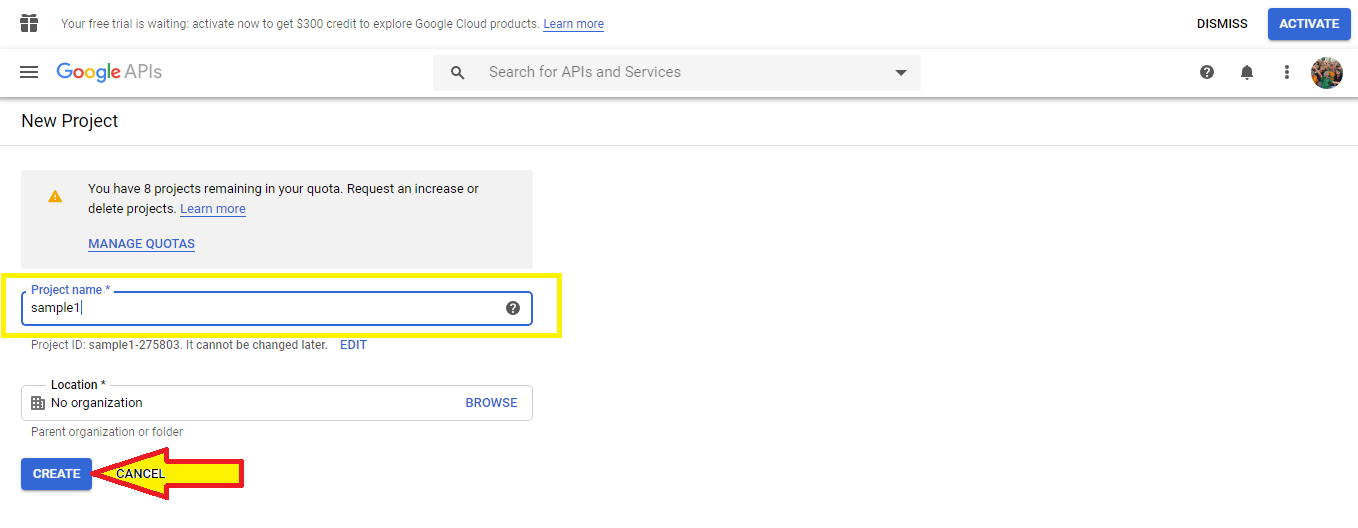
First need to get client ID and client secret from Google. Following are steps to get it.

Goto <https://console.developers.google.com/>

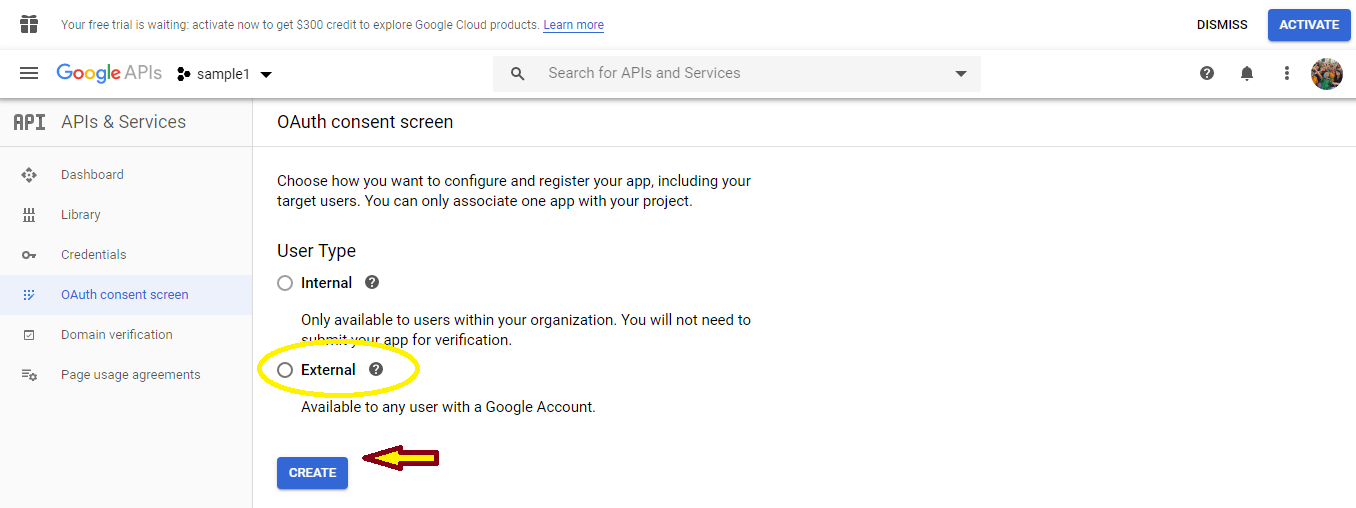
## Step 1: Create project



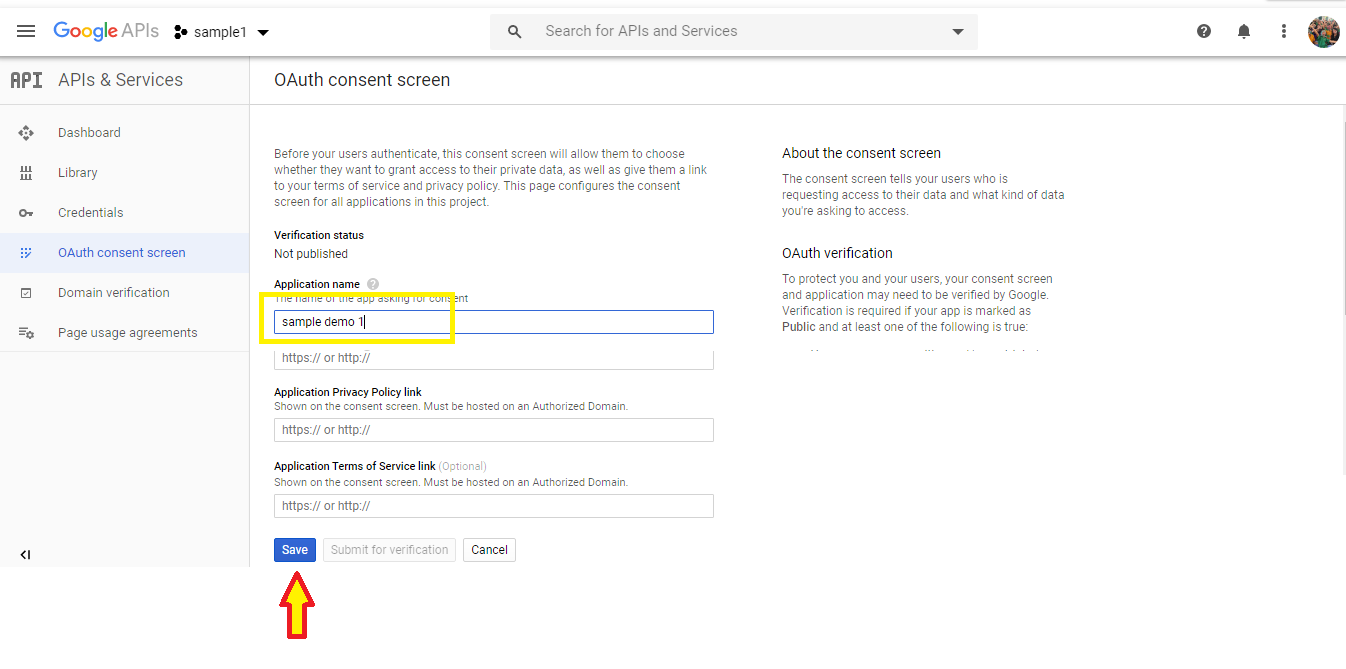
## Step 2: Give project name



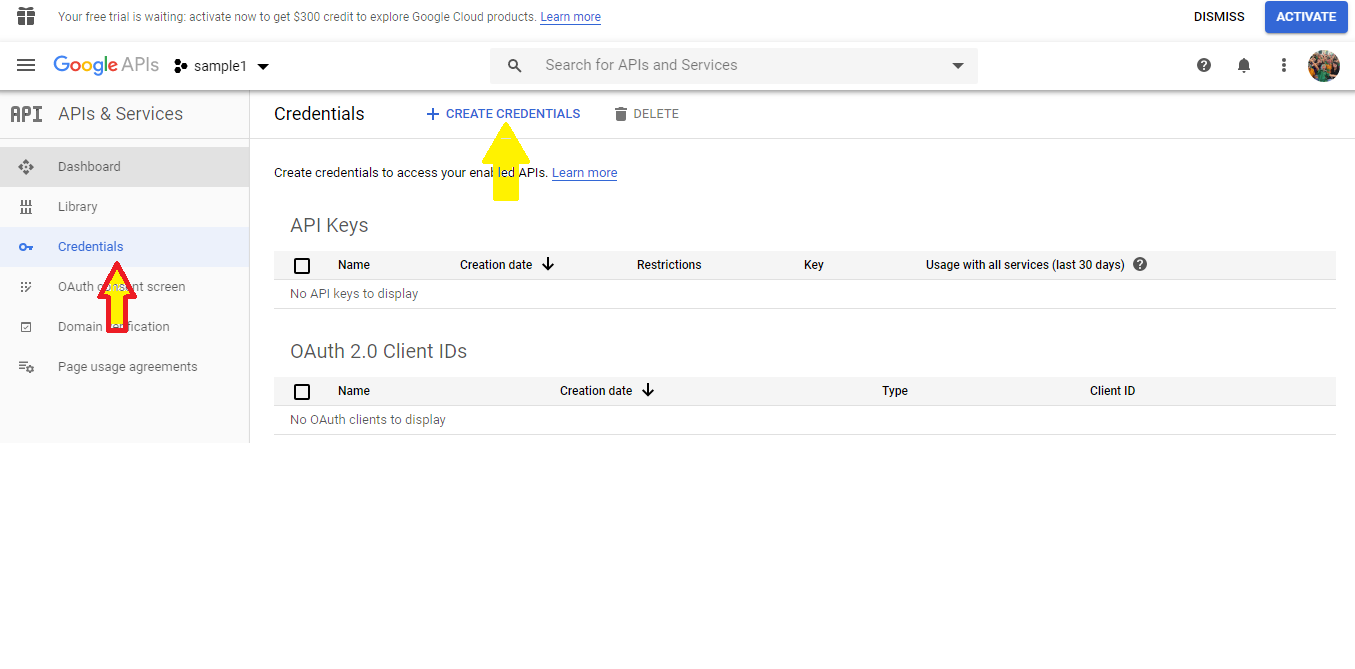
## Step 3: Select user type as External

****

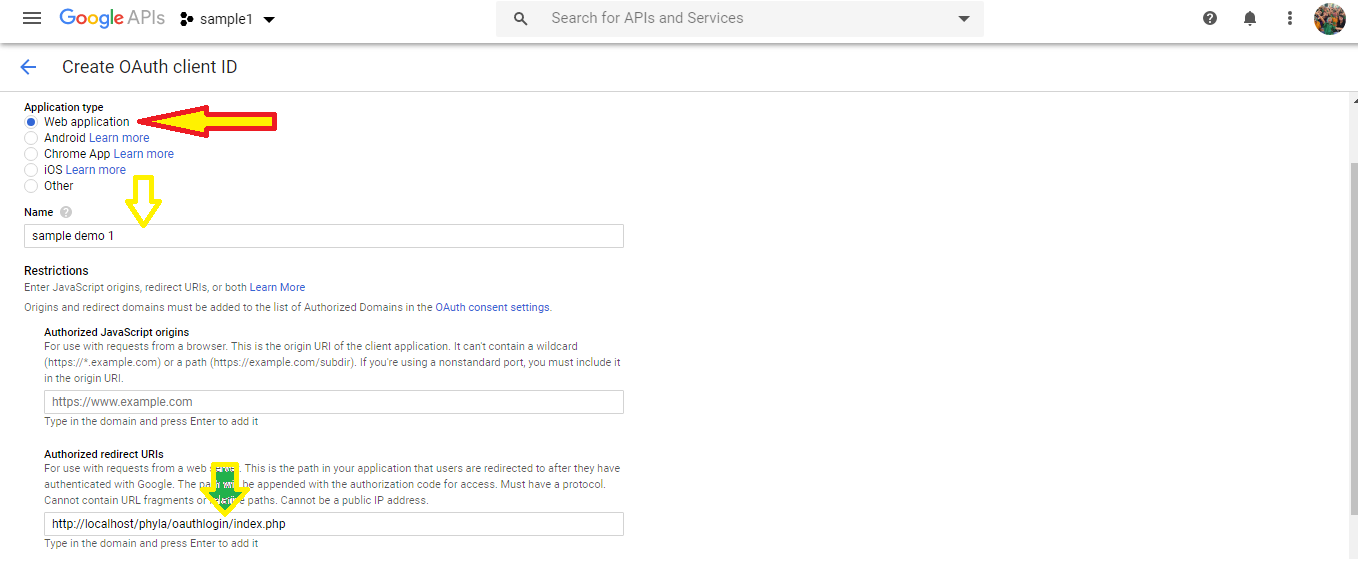
## Step 4: Give application name



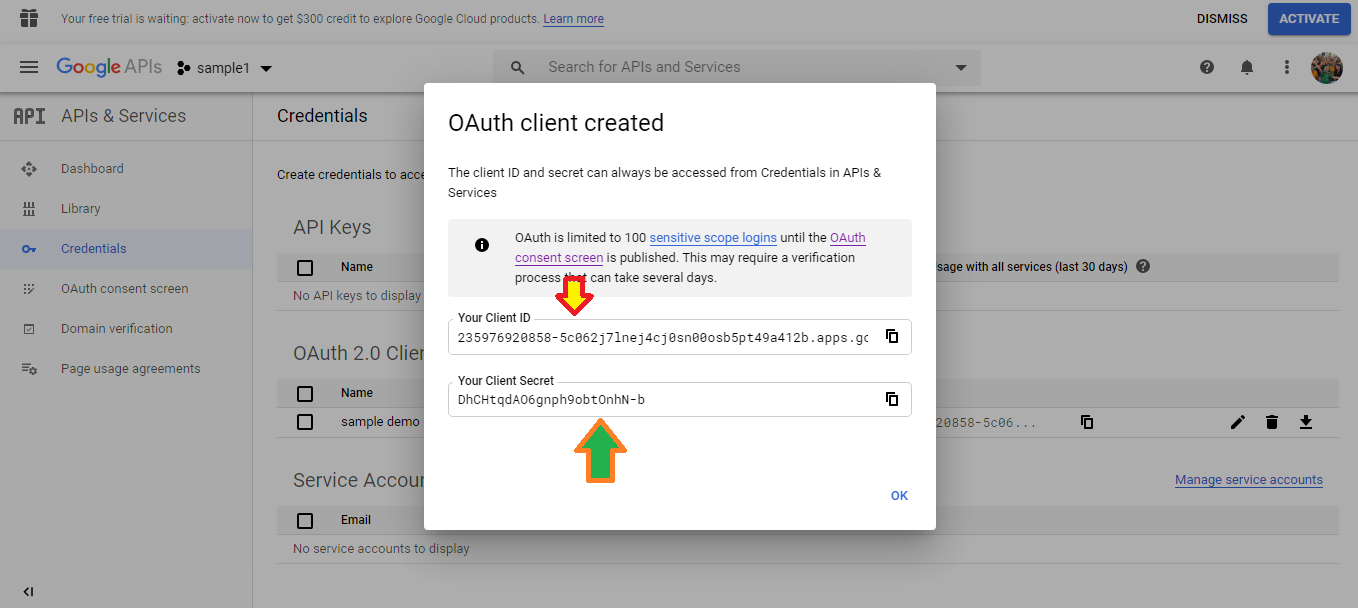
## Step 5: Create credentials



## Step 6: Select application type and give redirect URL



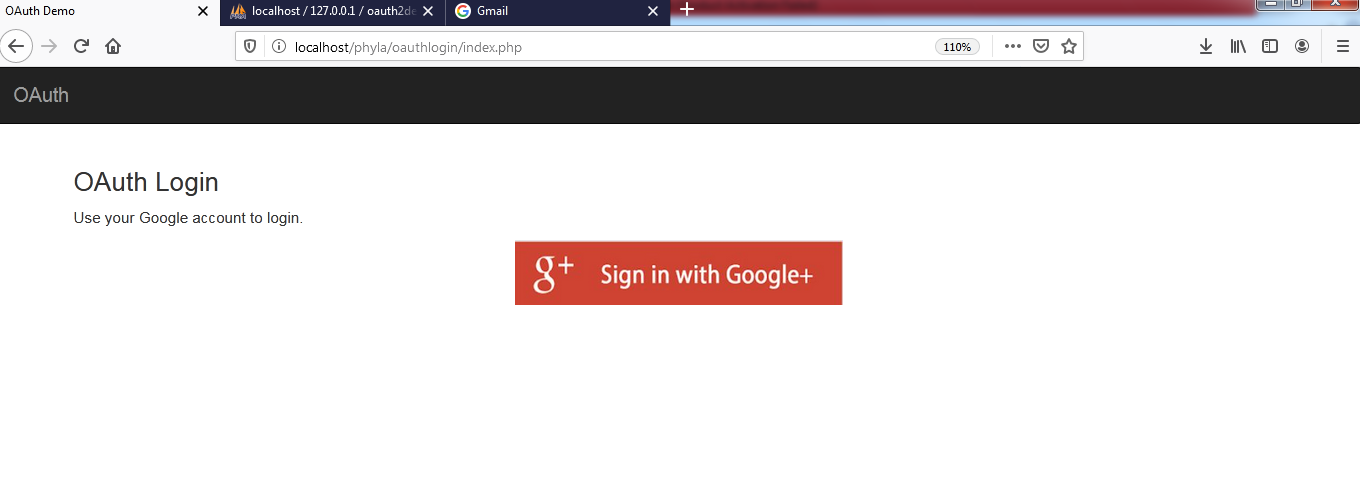
## Step 7: You can see your client ID & your secret ID.



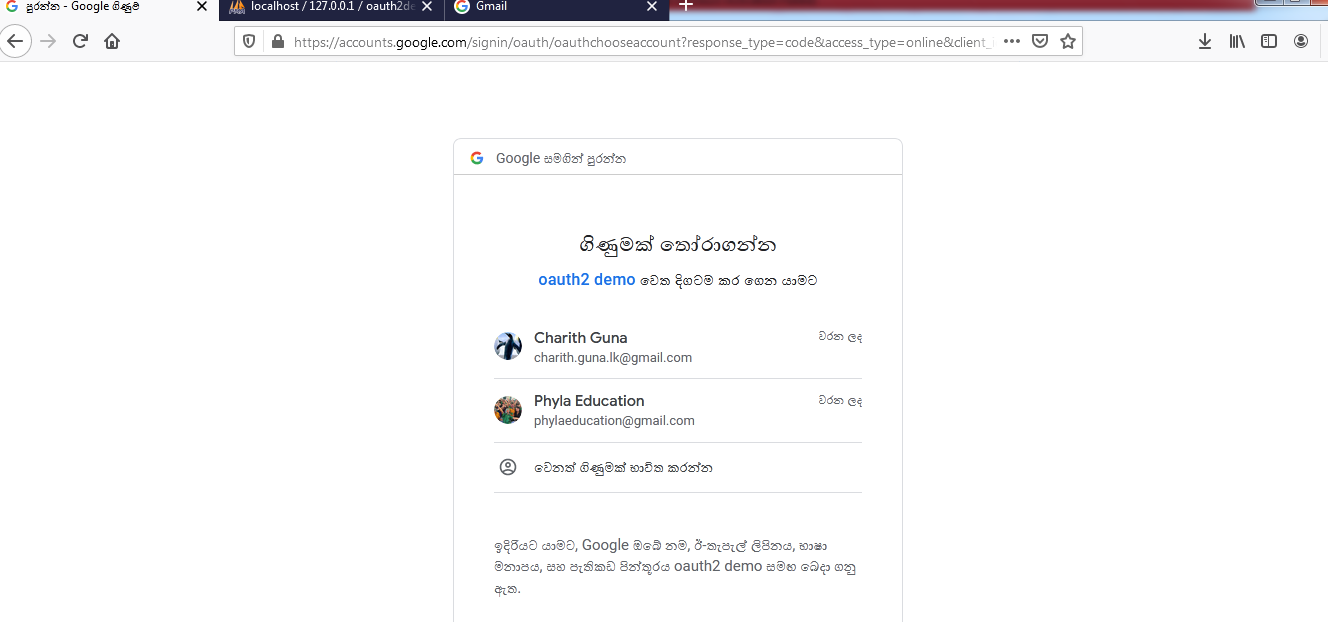
Following are php codes to connect with google api.

**Final PHP Application**

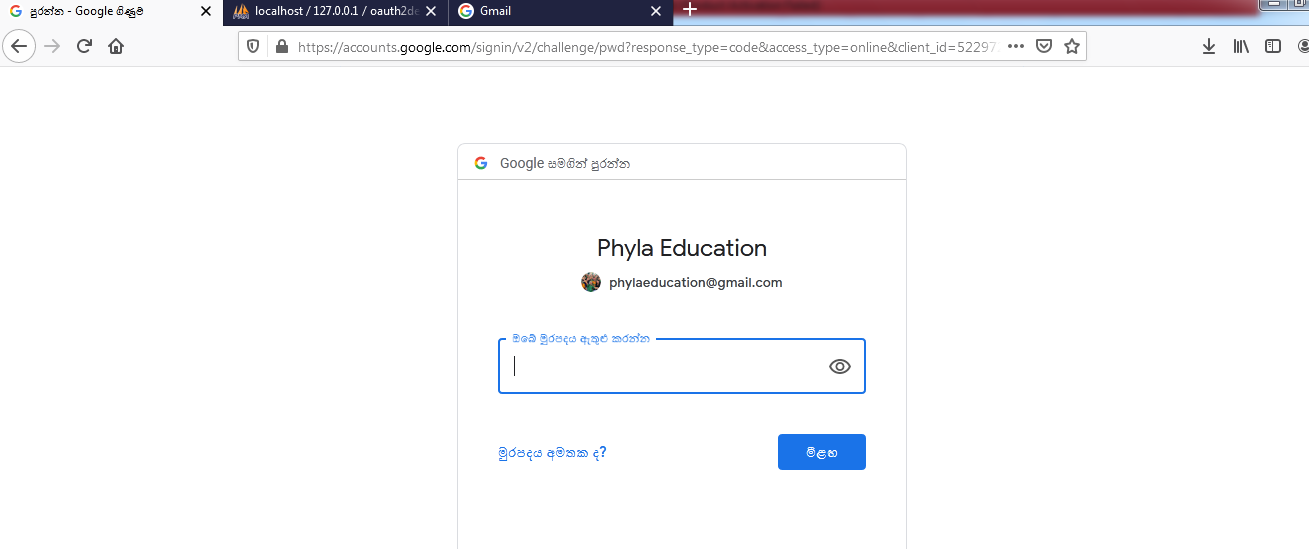
## OAuth Login Page



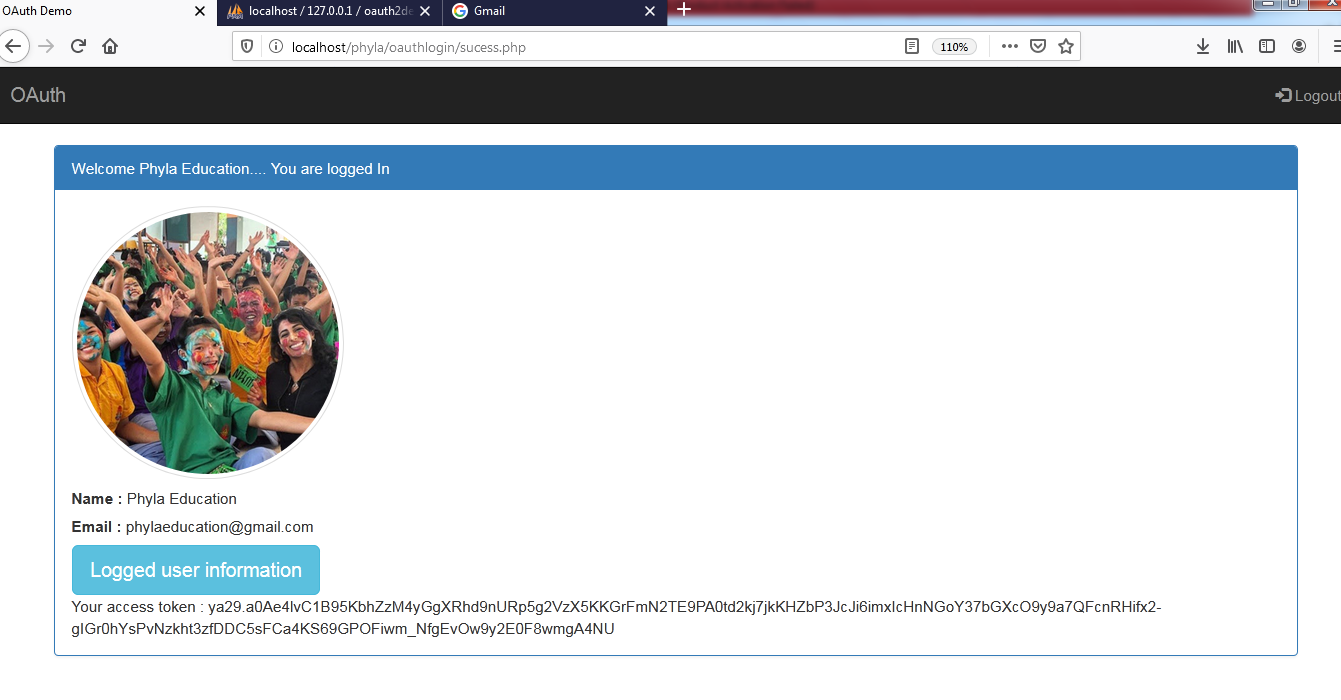
## Login Gmail Account



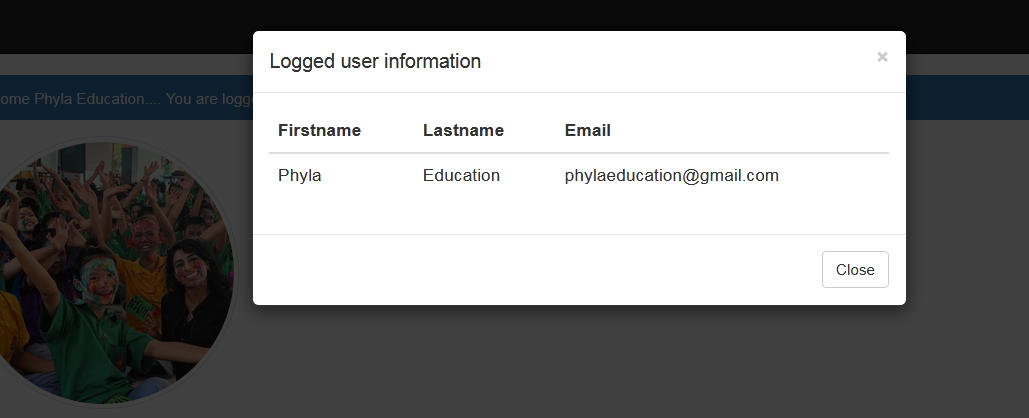
## Provide Password For Gmail



## Token Retrieve from Google



## Logged User Information

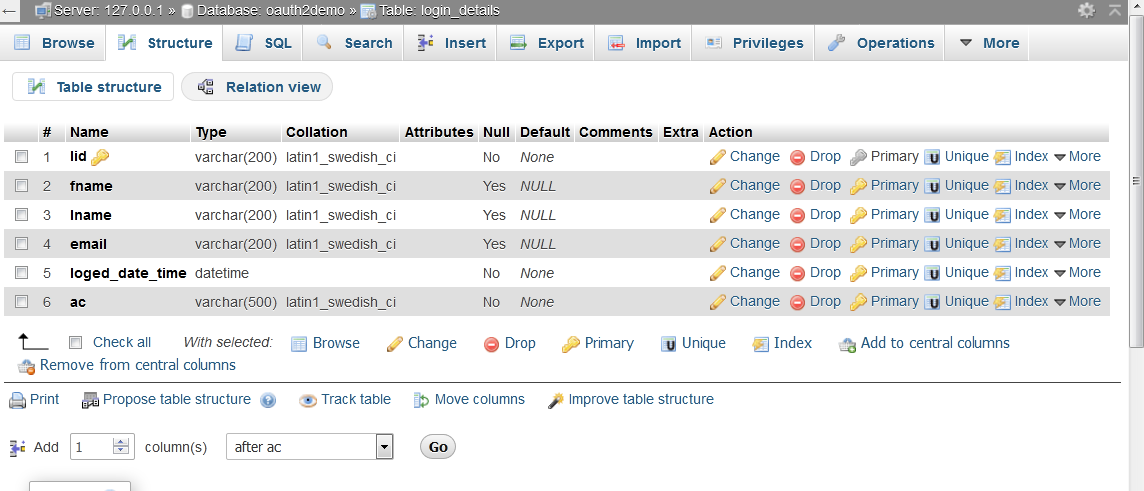


## 

## Database Details

Mysql database: [oauth2demo](http://localhost/phpmyadmin/db_structure.php?db=oauth2demo)

Table: login\_details



**Appendixes**

## PHP Code

<?php

//Include Google Client Library for PHP autoload file

require\_once 'vendor/autoload.php';

//Make object of Google API Client for call Google API

$google\_client = new Google\_Client();

//Set the OAuth 2.0 Client ID

$google\_client->setClientId('522972087523-ltgr274j8l3m8aqt05j1bjpbu7dapfi8.apps.googleusercontent.com');

//Set the OAuth 2.0 Client Secret key

$google\_client->setClientSecret('BkvH1-McbAF2On511ZgvY0C7');

//Set the OAuth 2.0 Redirect URI

$google\_client->setRedirectUri('http://localhost/phyla/oauthlogin/index.php');

//

$google\_client->addScope('email');

$google\_client->addScope('profile');

//start session on web page

session\_start();

?>

Index.php

<?php

//Include Configuration File

include('config.php');

$login\_button = '';

//This $\_GET["code"] variable value received after user has login into their Google Account redirct to PHP script then this variable value has been received

if(isset($\_GET["code"]))

{

//It will Attempt to exchange a code for an valid authentication token.

$token = $google\_client->fetchAccessTokenWithAuthCode($\_GET["code"]);

//This condition will check there is any error occur during geting authentication token. If there is no any error occur then it will execute if block of code/

if(!isset($token['error']))

{

//Set the access token used for requests

$google\_client->setAccessToken($token['access\_token']);

//Store "access\_token" value in $\_SESSION variable for future use.

$\_SESSION['access\_token'] = $token['access\_token'];

//Create Object of Google Service OAuth 2 class

$google\_service = new Google\_Service\_Oauth2($google\_client);

//Get user profile data from google

$data = $google\_service->userinfo->get();

//Below you can find Get profile data and store into $\_SESSION variable

if(!empty($data['given\_name']))

{

$\_SESSION['user\_first\_name'] = $data['given\_name'];

}

if(!empty($data['family\_name']))

{

$\_SESSION['user\_last\_name'] = $data['family\_name'];

}

if(!empty($data['email']))

{

$\_SESSION['user\_email\_address'] = $data['email'];

}

if(!empty($data['gender']))

{

$\_SESSION['user\_gender'] = $data['gender'];

}

if(!empty($data['picture']))

{

$\_SESSION['user\_image'] = $data['picture'];

}

}

}

//This is for check user has login into system by using Google account, if User not login into system then it will execute if block of code and make code for display Login link for Login using Google account.

if(!isset($\_SESSION['access\_token']))

{

//Create a URL to obtain user authorization

$login\_button = '<a href="'.$google\_client->createAuthUrl().'"><img src="google-login-button-png-3.jpg" width="300" height="60" /></a>';

}

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>OAuth Demo</title>

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<meta content='width=device-width, initial-scale=1, maximum-scale=1' name='viewport'/>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

<?php include 'menu\_version1.php'; ?>

<div class="container">

<div class="panel-group">

<?php

if($login\_button == '')

{

header("Location: sucess.php");

}

else

{

include 'menu\_version2.php';

echo '<div align="center">'.$login\_button . '</div>';

}

?>

</div>

</div>

</body>

</html>

Success.php – this page is redirect page after user login with google credentials

<?php

session\_start();

include('dbcon/dbconnect.php');

mysql\_select\_db("oauth2demo", $con);

$uimg=$\_SESSION["user\_image"];

$ufn=$\_SESSION['user\_first\_name'];

$uln=$\_SESSION['user\_last\_name'];

$uemail=$\_SESSION['user\_email\_address'];

$uac=$\_SESSION['access\_token'];

//$dt2=date("Y-m-d H:i:s");

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>OAuth Demo</title>

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<meta content='width=device-width, initial-scale=1, maximum-scale=1' name='viewport'/>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

<?php include 'menu\_version3.php'; ?>

<div class="container">

<div class="panel-group">

<div class="panel panel-primary">

<div class="panel-heading">Welcome <?php echo $ufn.' '.$uln; ?>.... You are logged In

</div>

<div class="panel-body">

<?php

echo '<img src="'.$uimg.'" class="img-responsive img-circle img-thumbnail" width="250" height="150" />';

echo '<h5><b>Name :</b> '.$ufn.' '.$uln.'</h5>';

echo '<h5><b>Email :</b> '.$uemail.'</h5>';

$check\_loger\_alredy\_registered=mysql\_query("select \* from login\_details where email='$uemail'")or die(mysql\_error);

$check\_loger\_alredy = mysql\_num\_rows($check\_loger\_alredy\_registered);

if($check\_loger\_alredy == 0){

$lid=md5(uniqid($uemail, true));

$sql="INSERT INTO login\_details (lid,fname,lname,email,ac)

VALUES

('$lid','$ufn','$uln','$uemail','$uac')";

if (!mysql\_query($sql,$con))

{

die('Error: ' . mysql\_error());

}

}

$loger\_details=mysql\_query("select \* from login\_details ORDER BY loged\_date\_time DESC")or die(mysql\_error);

?>

<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Logged user information</button>

<br>Your access token : <?php echo $uac; ?>

<!-- Modal -->

<div class="modal fade" id="myModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Logged user information</h4>

</div>

<div class="modal-body">

<table class="table">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<?php

while ($row\_loger\_details = mysql\_fetch\_array($loger\_details)) {

$firstname=$row\_loger\_details['fname'];

$lastname=$row\_loger\_details['lname'];

$loger\_email=$row\_loger\_details['email'];

?>

<tr>

<td><?php echo $firstname; ?></td>

<td><?php echo $lastname; ?></td>

<td><?php echo $loger\_email; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

Logout.php

<?php

//logout.php

include('config.php');

unset($\_SESSION['access\_token']);

//Reset OAuth access token

$google\_client->revokeToken();

//Destroy entire session data.

session\_destroy();

//redirect page to index.php

header('location:index.php');

?>

Logout.php

<?php

include('config.php');

unset($\_SESSION['access\_token']);

//Reset OAuth access token

$google\_client->revokeToken();

//Destroy entire session data.

session\_destroy();

//redirect page to index.php

header('location:index.php');

?>