



88823759 Software Development Training Camp

Bachelor of Science Program in Software Engineering Faculty of Informatics, Burapha University Thailand

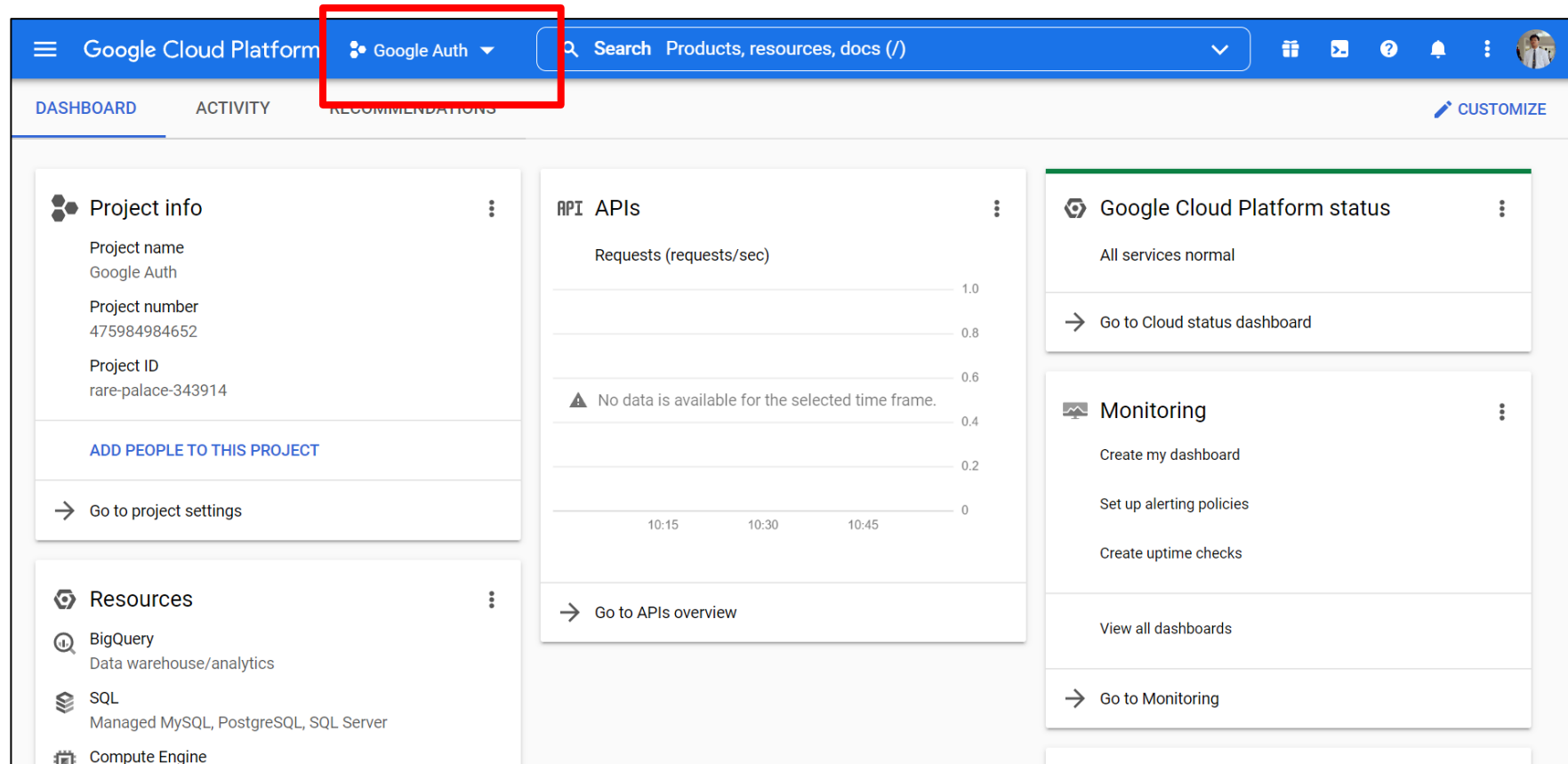
Google Auth

Login ด้วยบัญชี Gmail

Google Console Developer

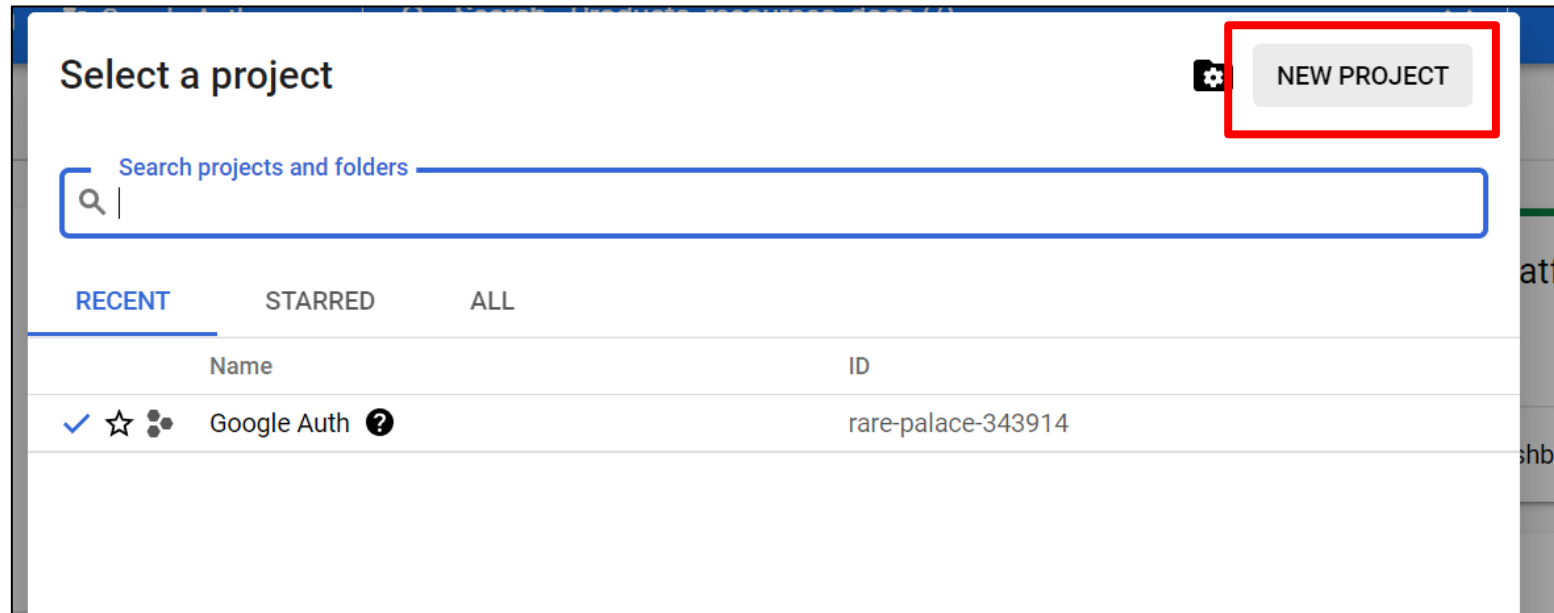
Google Console Developer

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



สร้าง Project

ทำการสร้าง Project โดยกดปุ่ม “NEW PROJECT”



สร้าง Project (2)

กรอกข้อมูล Project จากนั้นกดปุ่ม “CREATE”

Project name *

Google Login

?

Project ID: serene-obelisk-344516. It cannot be changed later. [EDIT](#)

Location *

No organization

BROWSE

Parent organization or folder

CREATE

CANCEL

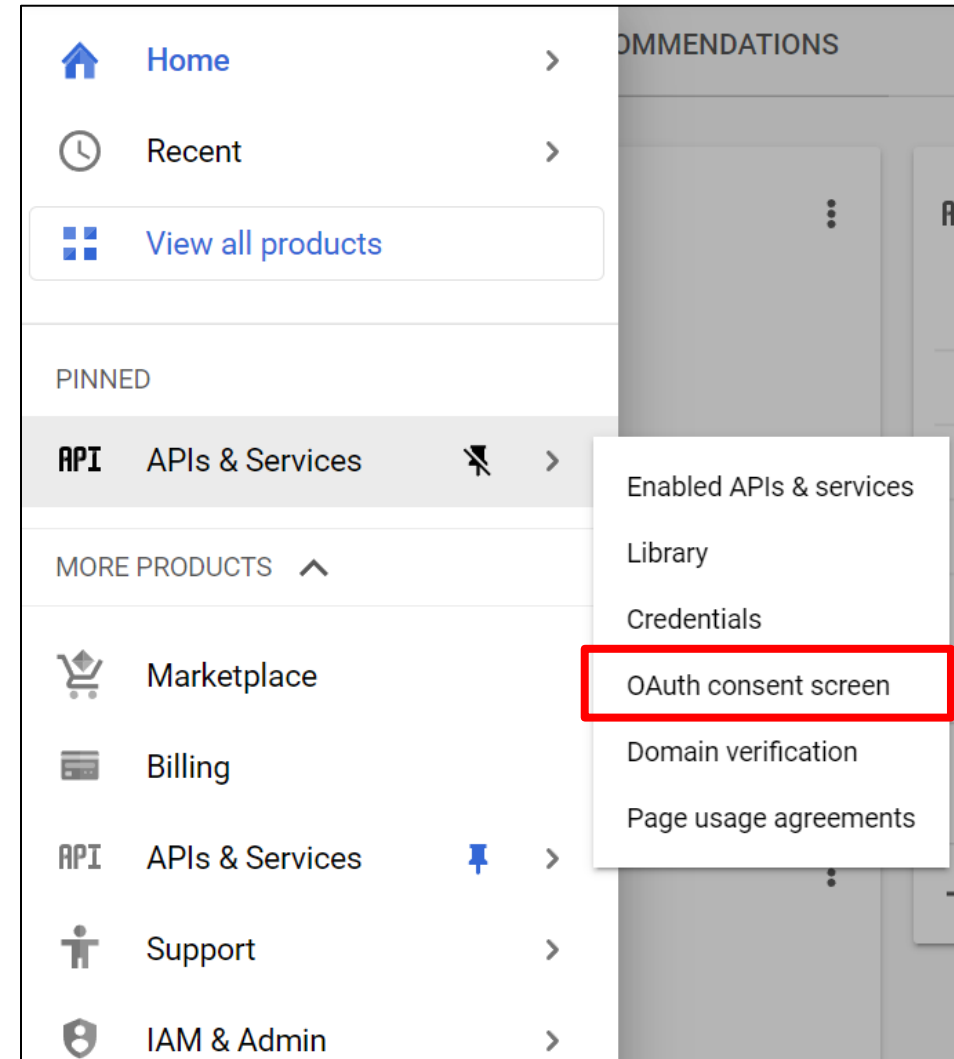
สร้าง Project (3)

- เมื่อสร้าง Project เสร็จสิ้น ให้เลือก Project ที่เพิ่งสร้าง
- โดยชื่อ Project จะต้องแสดงที่แถบเมนูด้านบน ดังภาพ



สร้าง Google API

- เลือกเมนู “APIs & Services”
- และเลือกเมนูย่อย “OAuth consent screen”



สร้าง Google API (2)

- เลือก User Type
 - Internal อนุญาตเฉพาะเมลล์ขององค์กร เช่น go.buu.ac.th
 - External อนุญาตทั้งหมด
- ในที่นี้เลือก External จากนั้นกดปุ่ม “CREATE”

User Type

☐ Internal ?

Only available to users within your organization. You will not need to submit your app for verification. [Learn more about user type](#)

☒ External ?

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. [Learn more about user type](#)

CREATE

สร้าง Google API (3)

- กรอกชื่อ App name
- และ User support email
- กรอก Application home page

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *

Camp System

The name of the app asking for consent

User support email *

sunneed.2555@gmail.com

For users to contact you with questions about their consent

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page

http://localhost/google_login/public/Login

Provide users a link to your home page

สร้าง Google API (4)

- Email address ของ Developer
- จากนั้นกดปุ่ม “SAVE AND CONTINUE”

Developer contact information

Email addresses *

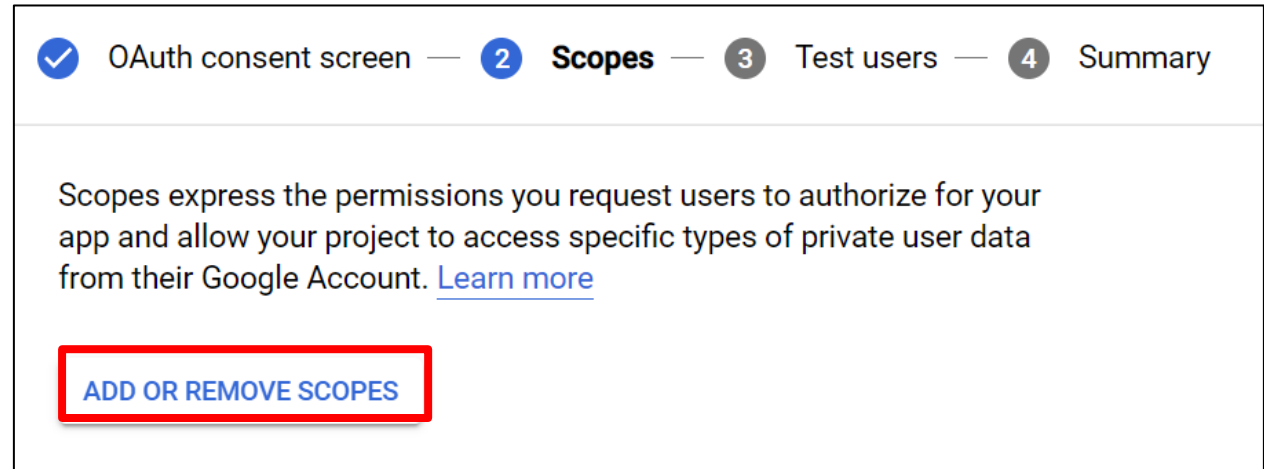
sunneed.2555@gmail.com ✕

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE CANCEL

สร้าง Google API (5)

- ขั้นตอนที่ 2 เป็นการกำหนดขอบเขตการเข้าถึงข้อมูล (Scope)
- ให้กดปุ่ม “ADD OR REMOVE SCOPES”
- เลือก email และ profile
- จากนั้นกดปุ่ม “UPDATE” ด้านล่าง





The screenshot shows a table of selected scopes. At the top, there is a 'Filter' section with the text 'Enter property name or value' and a help icon. The table has three columns: a checkbox column, an 'API' column with an upward arrow, a 'Scope' column, and a 'User-facing description' column. Two rows are visible, both with checked checkboxes. A red rectangular box highlights the 'Scope' column for both rows.

	API ↑	Scope	User-facing description
<input checked="" type="checkbox"/>		.../auth/userinfo.email	See your primary Google Account email address
<input checked="" type="checkbox"/>		.../auth/userinfo.profile	See your personal info, including any personal info you've made publicly available

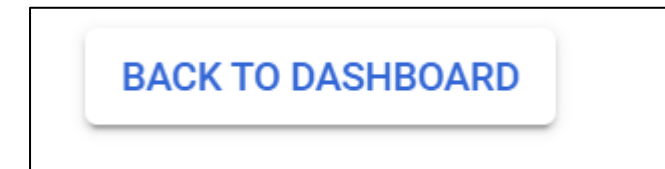
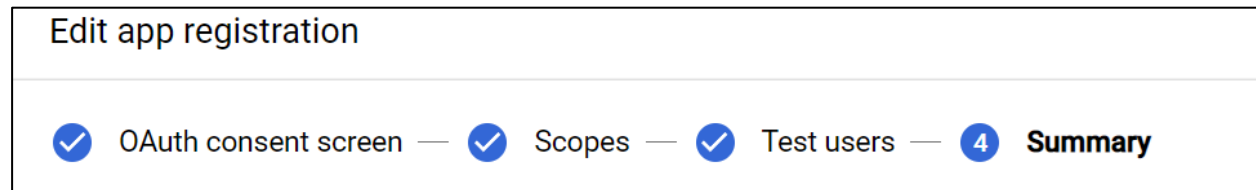
สร้าง Google API (6)

- ระบบจะแสดง Scope ที่เลือก
- ในที่นี้ email และ profile เป็นข้อมูลที่อยู่ในกลุ่ม “Non-sensitive”
- หาก Publish App ไม่จำเป็นต้องรับการตรวจสอบจาก Google

Your non-sensitive scopes			
API ↑	Scope	User-facing description	
	.. ./auth/userinfo .email	See your primary Google Account email address	
	.. ./auth/userinfo .profile	See your personal info, including any personal info you've made publicly available	

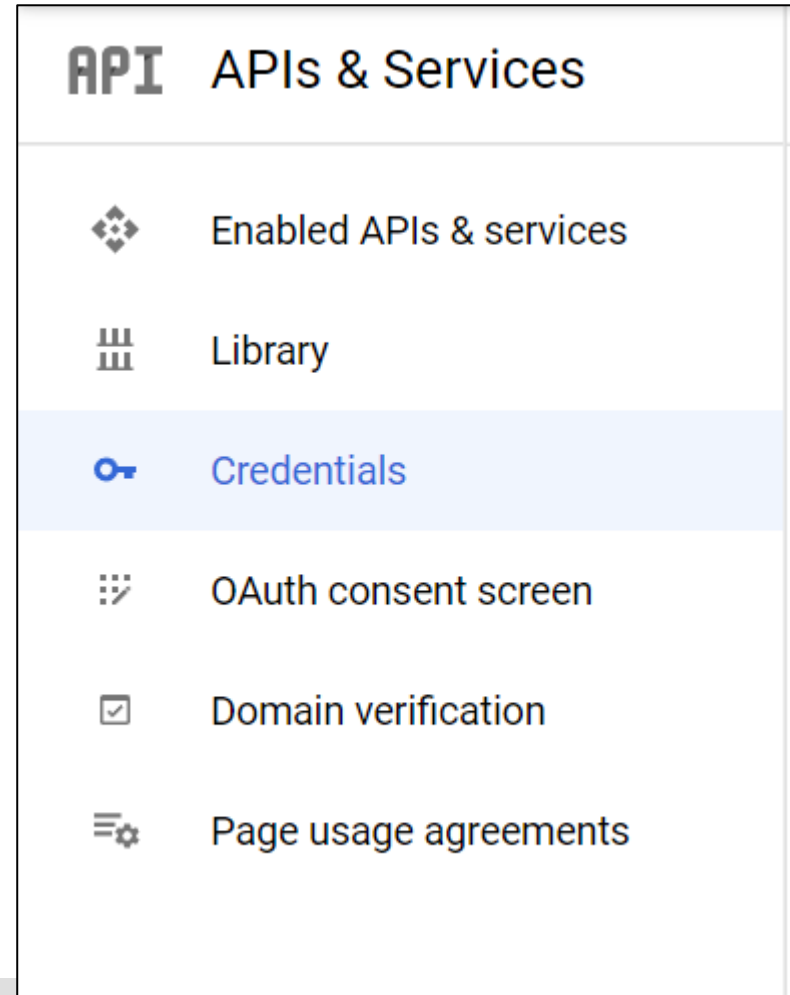
สร้าง Google API (7)

- ทำการกดปุ่ม “SAVE AND CONTINUE” จนถึงขั้นตอนที่ 4 “SUMMARY”
- กดปุ่ม “BACK TO DASHBOARD” ถือเป็นอันเสร็จสิ้น



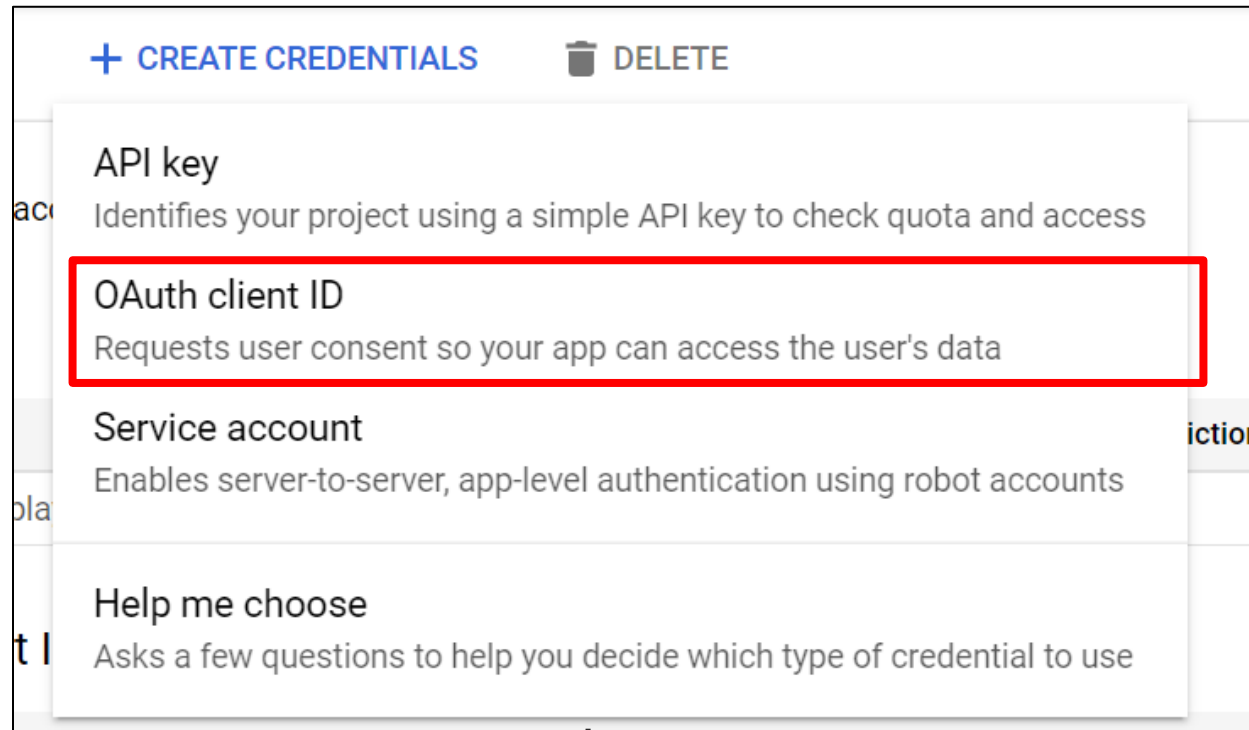
สร้าง API Key

- เลือกเมนู “Credentials”




สร้าง API Key (2)

กดปุ่ม “+ CREATE CREDENTIALS” จากนั้นเลือก OAuth client ID



สร้าง API Key (3)

- เลือก Application type เป็น “Web application”
- จากนั้นกรอก Name ของ OAuth client ID

 Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *

Web application ▼

Name *

Camp System

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

สร้าง API Key (4)

- กรอก Redirect URL (เป็น URL ที่ระบบจะไปหลังจากเลือก Email แล้ว)
- จากนั้นกดปุ่ม “CREATE”

Authorized redirect URIs

For use with requests from a web server

URIs 1 *

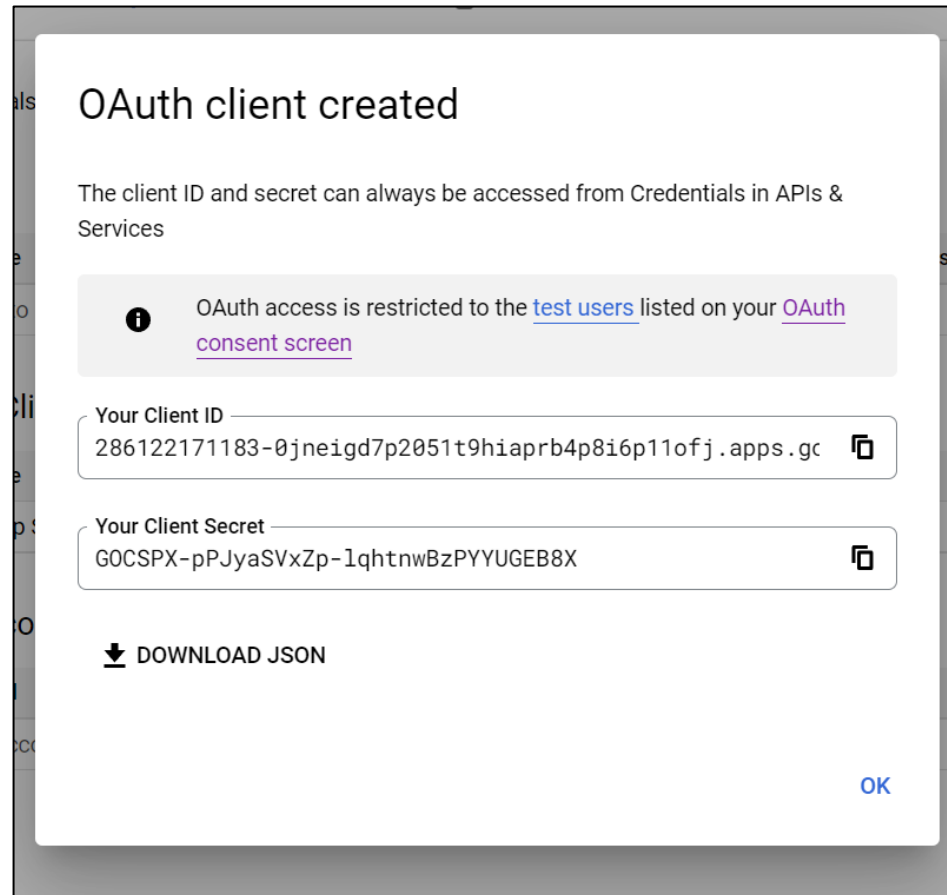
[+ ADD URI](#)

Note: It may take 5 minutes to a few hours for settings to take effect

[CREATE](#) [CANCEL](#)

สร้าง API Key (5)

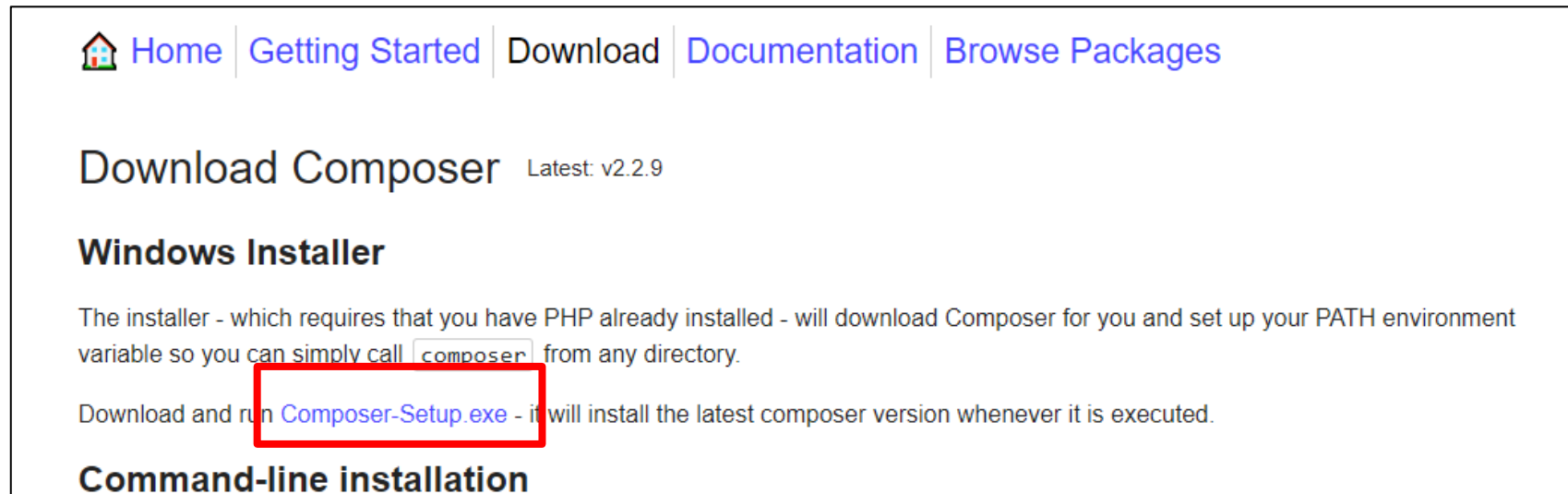
หลังจากกดปุ่ม “CREATE” จะสร้าง Client ID และ Client Secret ให้



ติดตั้ง Composer

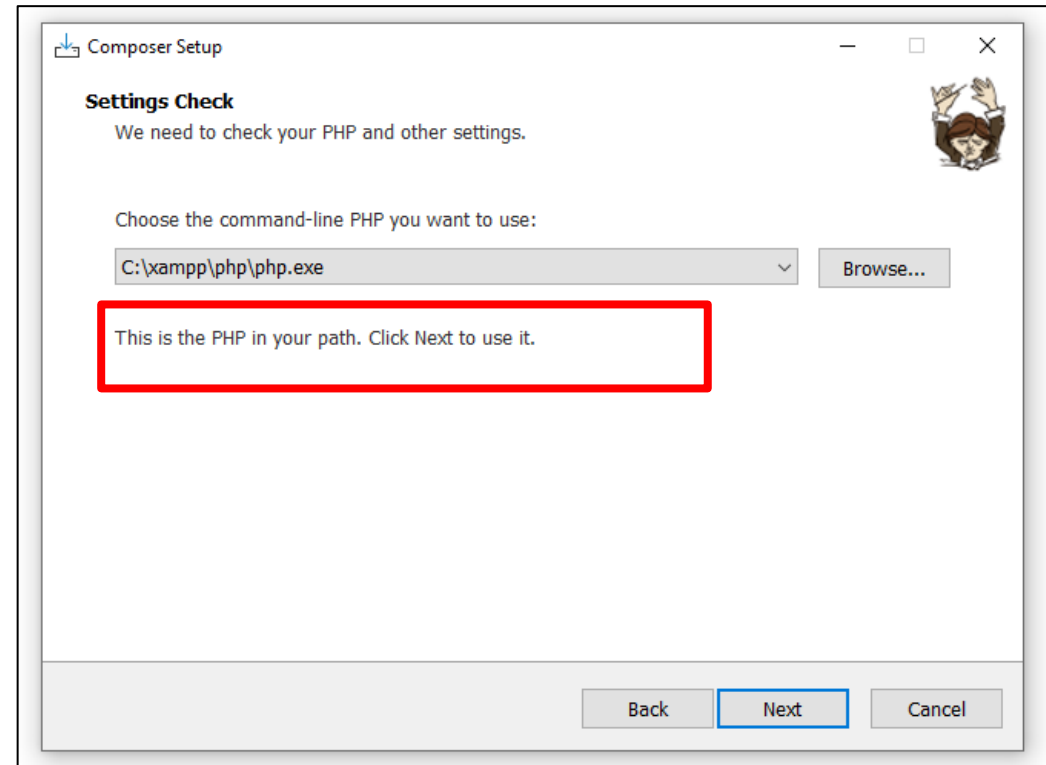
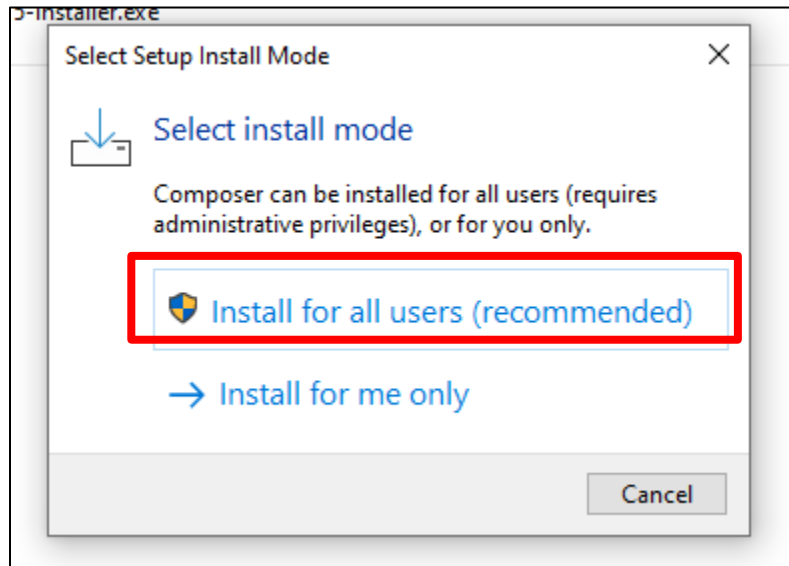
ดาวน์โหลด Composer

- Composer เป็นเครื่องมือช่วยดาวน์โหลด Package ของ PHP
- ในที่นี้ จะใช้ติดตั้ง Package “Google API Client”
- ดาวน์โหลด Composer ได้ที่ [Composer \(getcomposer.org\)](https://getcomposer.org)



ติดตั้ง Composer

- ติดตั้ง Composer สำหรับทุก Users
- ในขั้นตอนตรวจสอบ PHP ในเครื่อง หากมี Checkbox เพื่อให้เพิ่ม PHP เข้าสู่ Environment Variable ให้เลือก Checkbox



ติดตั้ง Google API Client

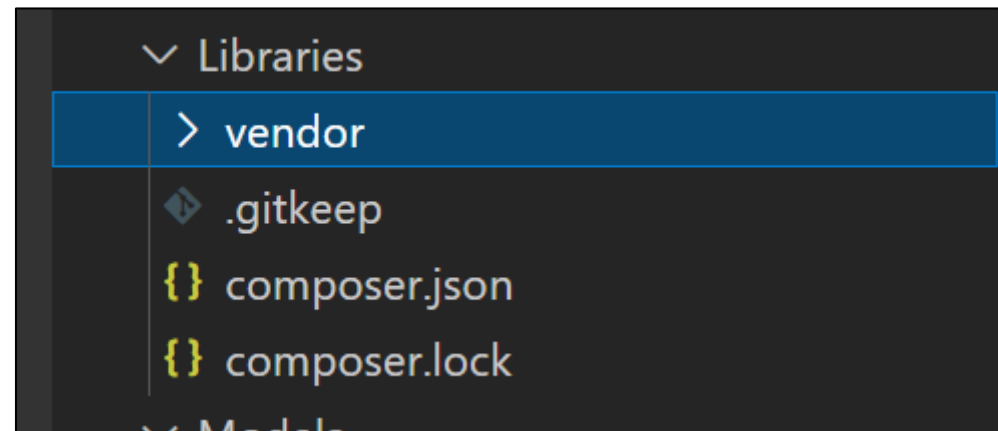
ติดตั้ง Google API Client

- เปิด Project ใน Visual Studio Code
- จากนั้นเปิด Terminal ด้วย คำสั่ง CTRL + J
- เปลี่ยนตำแหน่งไปยัง app\Libraries
- ใช้คำสั่ง composer require google/apiclient:^2.7
- Composer จะถามว่าต้องการใช้ไฟล์ composer.json ใน Folder Root หรือไม่
- ตอบ n (No)

```
PS C:\xampp\htdocs\google_login> cd .\app\Libraries\  
PS C:\xampp\htdocs\google_login\app\Libraries> composer require google/apiclient:^2.7  
No composer.json in current directory, do you want to use the one at C:\xampp\htdocs\google_login? [Y,n]? n  
./composer.json has been created  
Running composer update google/apiclient  
Loading composer repositories with package information
```

ติดตั้ง Google API Client (ต่อ)

- รอสักครู่ จนจบการติดตั้ง
- จะพบ Folder vendor ไฟล์ composer.json และ composer.lock
- ถือเป็นการเสร็จสิ้นการติดตั้ง Google API Client



Code Implementation

Controller Login.php

Method construct ของ Class Login ทำการกำหนดค่า Google Client

```
class Login extends BaseController
{
    private $google_client = NULL;
    function __construct(){
        require_once APPPATH. "libraries/vendor/autoload.php";
        session_start();

        // Create Google Client object
        $this->google_client = new \Google_Client();

        // Set client id & secret
        $this->google_client->setClientId("286122171183-0jneigd7p2051t9hiaprb4p8i6p11ofj.apps.googleusercontent.com");
        $this->google_client->setClientSecret("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");

        // Set redirect page
        $this->google_client->setRedirectUri("http://localhost/google_login/public/Login/google_auth");

        // Set scopes
        // To access email
        // To access profile
        $this->google_client->addScope("email");
        $this->google_client->addScope("profile");
    }
}
```

Controller Login.php

- Method index() ใช้สำหรับแสดงหน้าจอ Login
- โดยส่งปุ่ม Login ไปพร้อมกับการแสดงหน้า View

```
public function index()
{
    // Create google login button
    $data['google_button'] = '<button><a href="'. $this->google_client->createAuthUrl().' ">Login with Google</a></button>';

    // Show login page with google login button
    return view('v_login', $data);
}
```

View v_login.php

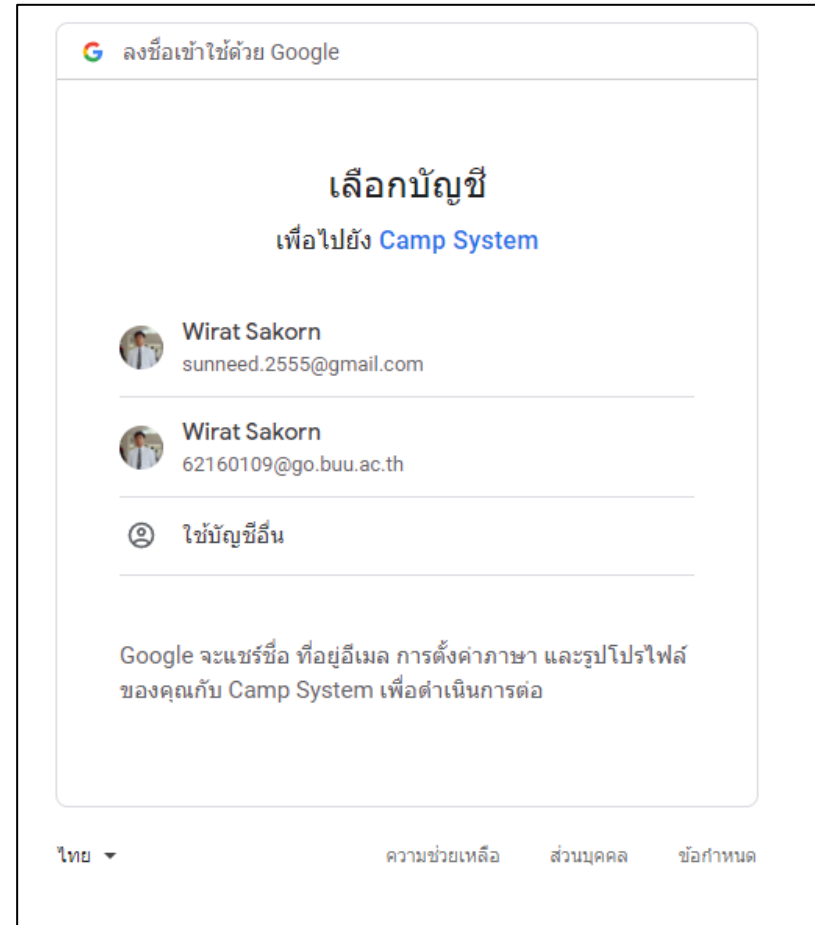
- แสดงปุ่ม Login ในหน้า View
- ไปที่ http://localhost/google_login/public/Login เพื่อดูผลลัพธ์

```
<body>  
|   <?php echo $google_button ?>  
</body>
```



View v_login.php

- เมื่อกดปุ่ม “Login with Google”
- ระบบจะให้เลือก Email



Controller Login.php

- เมื่อเลือก Email แล้ว ระบบจะทำงาน Method นี้
- ทำการรับค่า Account จาก Google
- จากนั้นกำหนดค่าที่ได้ ลงในตัวแปร SESSION
- สุดท้ายคือการนำไปแสดงหน้าที่จอ profile

```
public function google_auth()
{
    // Get token form Google by method GET
    $token = $this->google_client->fetchAccessTokenWithAuthCode($this->request->getGet('code'));

    // Access token without ERROR
    if(!isset($token['error'])){
        $this->google_client->setAccessToken($token['access_token']);

        // Create Google Service object
        $google_service = new \Google_Service_Oauth2($this->google_client);

        // Get Google Account data
        $data = $google_service->userinfo->get();

        // Set Google Account data to your application
        $user = [
            'oauth_id'    => $data['id'],
            'name'        => $data['givenName']. " ".$data['familyName'],
            'email'        => $data['email'] ,
            'profile_img' => $data['picture']
        ];

        // Set user data as SESSION
        $_SESSION["user"] = $user;
    }

    // Show profile page
    return redirect()->to(base_url()."/Login/profile");
}
```

```
public function profile() {
    return view('v_profile');
}
```

View v_profile.php

- แสดงข้อมูล Account จากตัวแปร SESSION

```
<body>
  <h1>Profile</h1>
  " alt="Profile image">
  <p>oauth id : <?php echo $_SESSION["user"]["oauth_id"] ?></p>
  <p><?php echo $_SESSION["user"]["name"] ?></p>
  <p><?php echo $_SESSION["user"]["email"] ?></p>

  <button>
    <a href="<?php echo base_url() . "/Login/logout" ?>">Logout</a>
  </button>
</body>
```

View v_profile.php

- แสดงข้อมูล Account จากตัวแปร SESSION

Profile



oauth id : 102949583202159114611

Wirat Sakorn

62160109@go.buu.ac.th

[Logout](#)