# OAuth 2 services on Moodle

To allow Google account users to login directly to Moodle platform

Documented by: Dr. Jingjing Lin, Center for IT-based Education, Toyohashi University of Technology (lin.jingjing.qc@tut.jp)

Note: This tutorial uses the Moodle instance of Toyohashi University of Technology (TUT) as the demonstration case. This site is not registered publicly as the Moodle site.

**TUTORIAL BEGINS BELOW:**

On Moodle, it allows third-party accounts to directly login to Moodle. They include: Google, Microsoft, Facebook, Nextcloud, LinkedIn, and other custom services.



Because TUT is using G suits as an organization. Therefore, every TUT user basically has one Google account. That is why we can login directly to external platforms such as https://medium.com/ using our TUT Gmail account without any effort of signing up.

The same rationale works for Moodle.

This documentation guides you to finish such a bridging, and what needs to be done. Two sides of efforts are required:
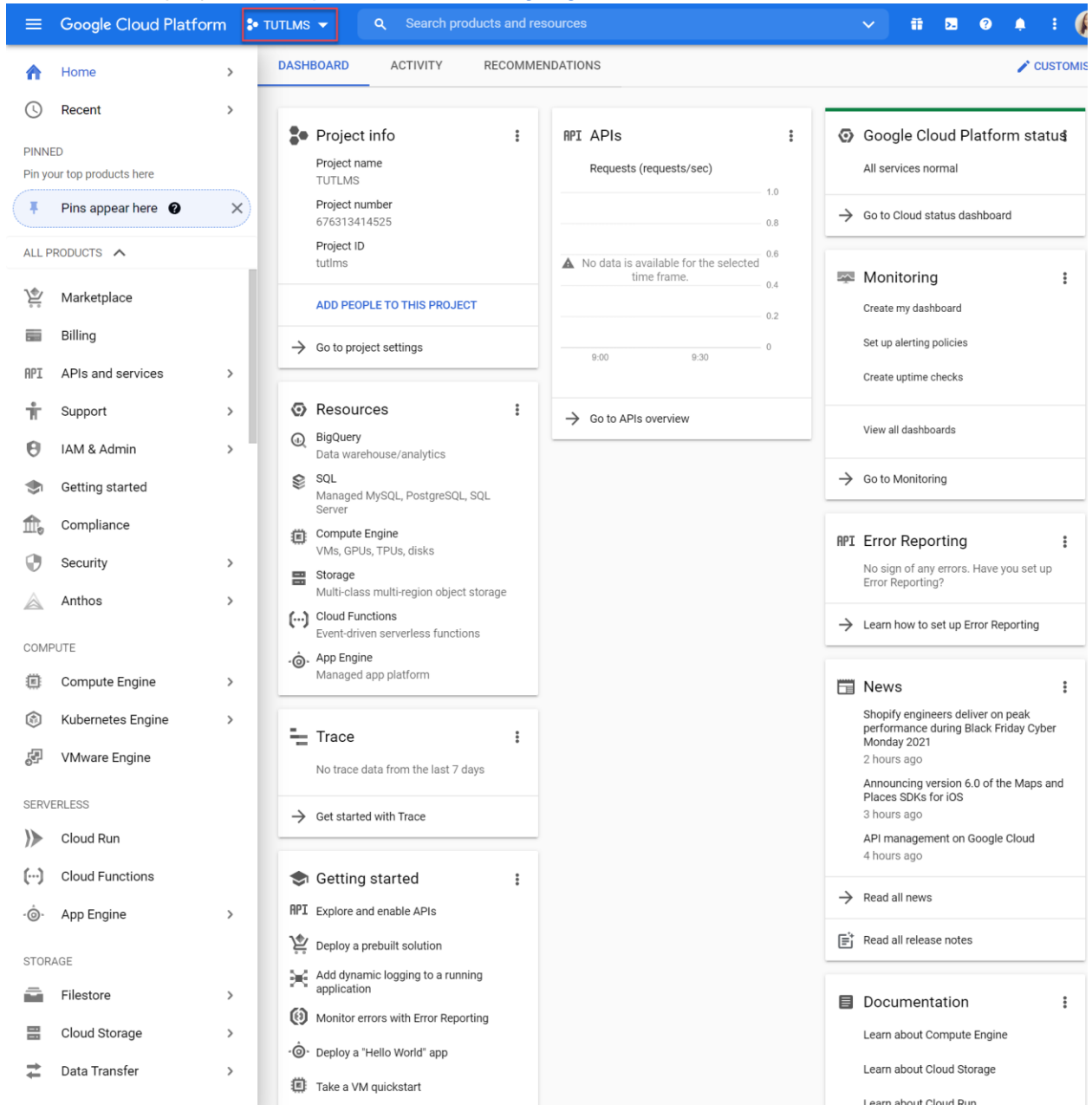- Google side
- Moodle side

At the end of the configuration, you will see the following screen on your homepage, which allows the users to login directly with their existing Google account by clicking the Google logo under the "Log in" button.
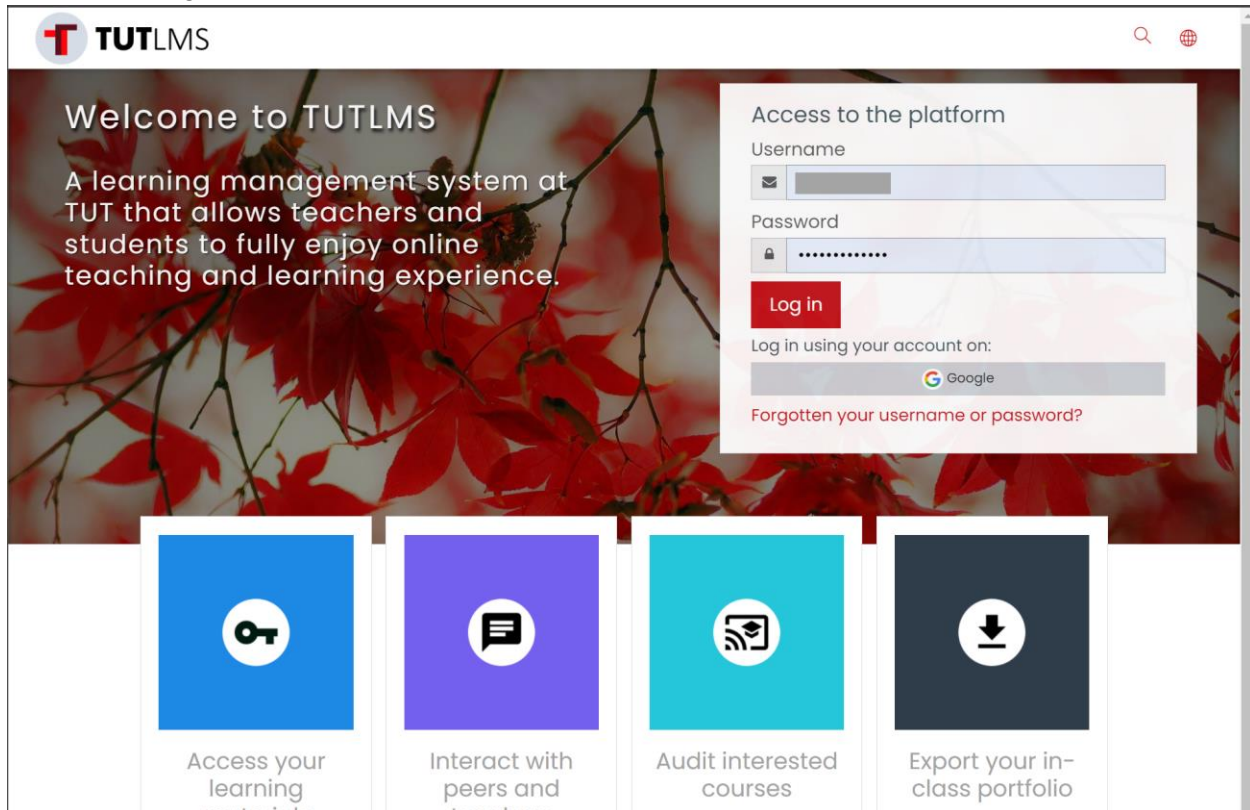
# Part I: Google side

## Step 1: Create a new project

Create a new project on https://console.cloud.google.com/



Click the red square will activate a pop-up window, as shown below.

Click "New project".

Give a name to your project, select the organization if you use a G suites admin account.
Because I use a personal Gmail account, therefore I skip the field of "Location" and directly click "Create" button to proceed.
You will be directed to the new project's admin home page as shown below.
Click "APIs and services" from the left menu in order to proceed to the next step.

# Step 2: Create a new OAuth consent screen

As for how to arrive at the APIs and services page, as you see below in the screenshot below, please refer to previous page.

Create a new OAuth consent screen (see the screenshot below). Choose Internal if you have a G suites admin account to limit the authentication to apply to only users within your organization. Select External, which is available to any test user with a Google account. For this documentation, I selected external because I do not have a G suites admin account in my organization.
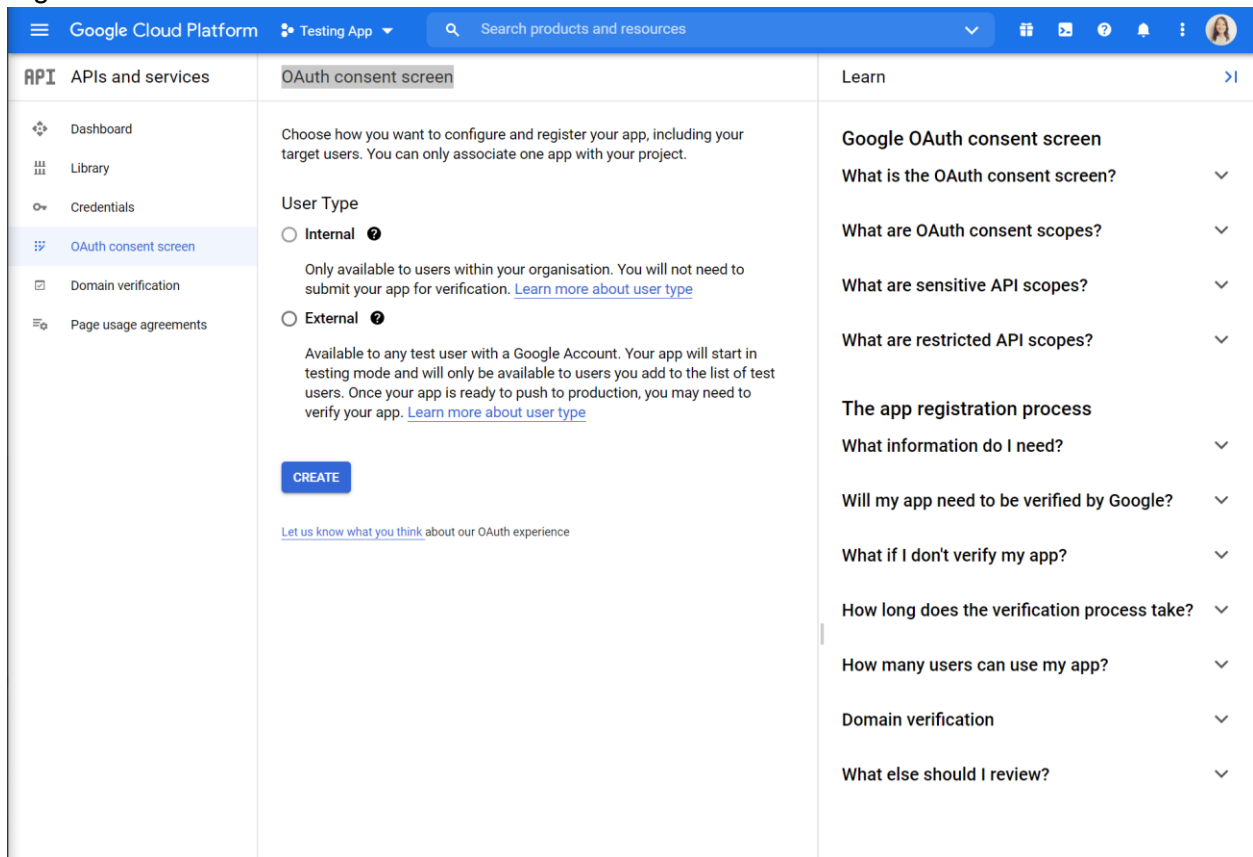


You will be directed to the following page as shown in the screenshot below. Give your App a name that you can recognize later. Insert the support email. Upload a logo if you want.

Continue to input your LMS's domain URL under "Application home page", and provide the links to the privacy policy page and terms of service link. They can be the same URL.

Add your organization's domains under "Authorised domains". In my case, I typed tut.ac.jp and tut.jp as two variations of my university.

In the end, input the developer contact email address.

Click "Save and continue" button to save all inputs.

You will continue to the "Scopes" page. I only selected non-sensitive scopes to pass non-sensitive information such as email address and first name/last name from Google account to Moodle.

Click "Save and continue" button to save all inputs.

I will leave everything empty here on the page of "Optional info".
Click "Save and continue" button to save all inputs.
You have finished creating the OAuth consent screen.

## Step 3: Create new credentials - Client ID and secret

As for how to arrive at the APIs and services page, as you see below in the screenshot, please refer to Page 5.

Click "Credentials" on the left menu of the page.

Click the "+Create credentials" button.

Choose "OAuth client ID".

The rest of the manual for this step can be found here on Github:
https://github.com/ronefel/moodle-mod_googlemeet/wiki/How-to-create-the-API-key-and-OAuth-client-ID

I will only add some additional information below.

- Leave the Authorized JavaScript origins field empty.
- In the Authorized redirect URI field, enter the URLs from your site. For me, I manage https://lms.cite.tut.ac.jp, therefore I have entered the following two URLs. For you, replace the "https://lms.cite.tut.ac.jp" with your managed domain in the following two URLs.
  - https://lms.cite.tut.ac.jp/admin/oauth2callback.php
  - https://lms.cite.tut.ac.jp



On the right side of the page, you will find the Client ID and Client secret. Copy them to a safe place. Or you can click the "Download JSON" button to keep a copy of this information somewhere safe.

# Step 4: Create new credentials - API key

As for how to arrive at the APIs and services page, as you see below in the screenshot, please refer to Page 5.
Click "Credentials" on the left menu of the page.



Click the "+Create credentials" button.

The rest of the manual for this step can be found here on Github:
https://github.com/ronefel/moodle-mod_googlemeet/wiki/How-to-create-the-API-key-and-OAuth-client-ID

You will find a screenshot showing the detail that you need to provide in the next page.

Under the "Website restrictions", I added two links:
- https://lms.cite.tut.ac.jp
- https://lms.cite.tut.ac.jp/*

You can replace the "https://lms.cite.tut.ac.jp" with your managed domain in the above two URLs.

Your API key is shown on the top right corner of the page. Keep a copy of it in a safe place.

So far, you have finished configuring on the Google side. You are ready to move to the Moodle side for further configuration.

# Part II: Moodle side

## Step 1: Create a Google service

Login as administrator.
Site administration -> Server -> OAuth 2 services (as shown below)

Click "Create Google service" button. (I have already created a Google service before. Therefore you see a record on the page)

Paste the Client ID and Client secret that you created earlier on the Google side. (Page 9-15)
Uncheck the "Require email verification".
Leave all other settings as default.
Click "Save changes".

You will see a red cross under "System account connected". There is actually no need to connect the system account to make the configuration work. Therefore I just ignored this red cross.

You can find more details about this step in the Moodle Manual:
https://docs.moodle.org/311/en/OAuth_2_services

However, if you use Google Drive document converter on your Moodle, you have to connect a system account here. Or else the Google Drive document converter will not work.

To manage the Document Converter:

Site administration -> Plugins -> Document converters -> Google Drive

If I click "Test this converter is working properly." The page will be directed to a warning message telling you that: "The OAuth service set in the document converter settings does not have a system account connected."

# Step 2: Manage authentication

Site administration -> Plugins -> Authentication -> Manage authentication

On the "Manage authentication" page, enable the OAuth 2 authentication method.



Click "Settings" to proceed.

I locked some values including: first name, last name, email address.

I changed the following values to unlocked if empty: country, ID number, Institution, Department.

Click "Save changes".

Attention: there is a warning message on the top of this page.
"*This authentication plugin allows users to create accounts on your site. You may want to enable the setting "authpreventaccountcreation" if you use this plugin.*"

This is exactly where we are going next.

Go back to the "Manage authentication" page.
Scroll down the page.
Under the "Common settings", you will find a field called "Prevent account creation when authenticating". Make sure that you uncheck this box.
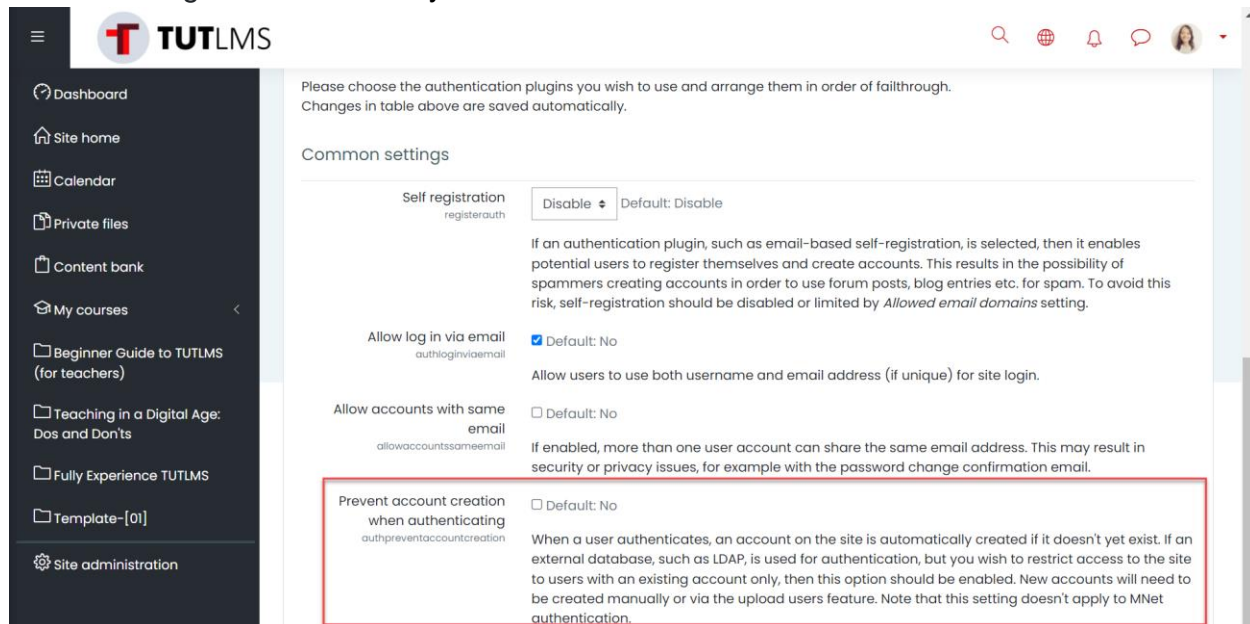


Because at the beginning, I checked it. And the configuration was awkward. It appeared that the users would need to login to their Google account first on the browser but outside this Moodle platform. If they didn't login to their Google account already and directly visited the Moodle and clicked Google logo, the page showed error when directing.

After I uncheck this box of "Prevent account creation when authenticating", the users can actually directly login with their Google account without encountering a problem. So this is a little bit different from the warning message that you can see on Page 21.

# Conclusion

The OAuth service on Moodle turned out to be an alternative login method that works for TUT email account holders. Because the TUT email account is connected to G suite service by Google. Therefore, to enable Google account login via the OAuth service actually can enable TUT email account holders to directly access to the CITE LMS without registering new accounts from scratch.

I explored this approach on my own with trial and error. The above described setup flow worked perfectly for the TUT Moodle instance. But in the end we did not adopt this approach but turned to the Shibboleth authentication. The READ ME text file serves as the guidance to installing Shibboleth on Moodle and it was implemented with the help of Information Media Center at TUT. Shibboleth turned out to be much more complicated procedure for a non system administrator role like me. Comparatively Oauth authentication method is much more friendly to master for anyone who is interested to introduce SSO in the Moodle instance.