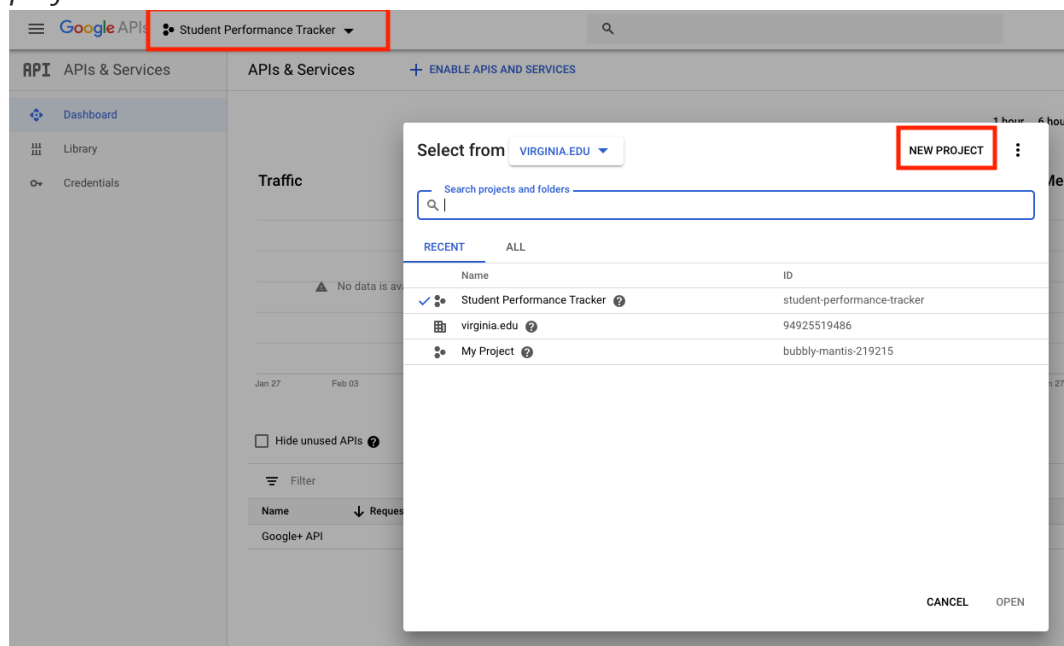


Setting up Google OAuth

For OAuth in development, use the domain `spt-acas.com` and add it to your `/etc/hosts` file so that it directs to localhost (instructions in development install instructions). For OAuth in production, use the domain that your server is accessible at.

1. Navigate to the [Google API Console](#)
 - o Login using your gmail
2. Create a new project by clicking on the box next to *Google APIs* then selecting *new project*



3. Name your new project and click create

New Project

Project Name *
SPT Demo ?

Project ID: extreme-quasar-232804. It cannot be changed later. [EDIT](#)

Organization
virginia.edu ?

This project will be attached to virginia.edu.

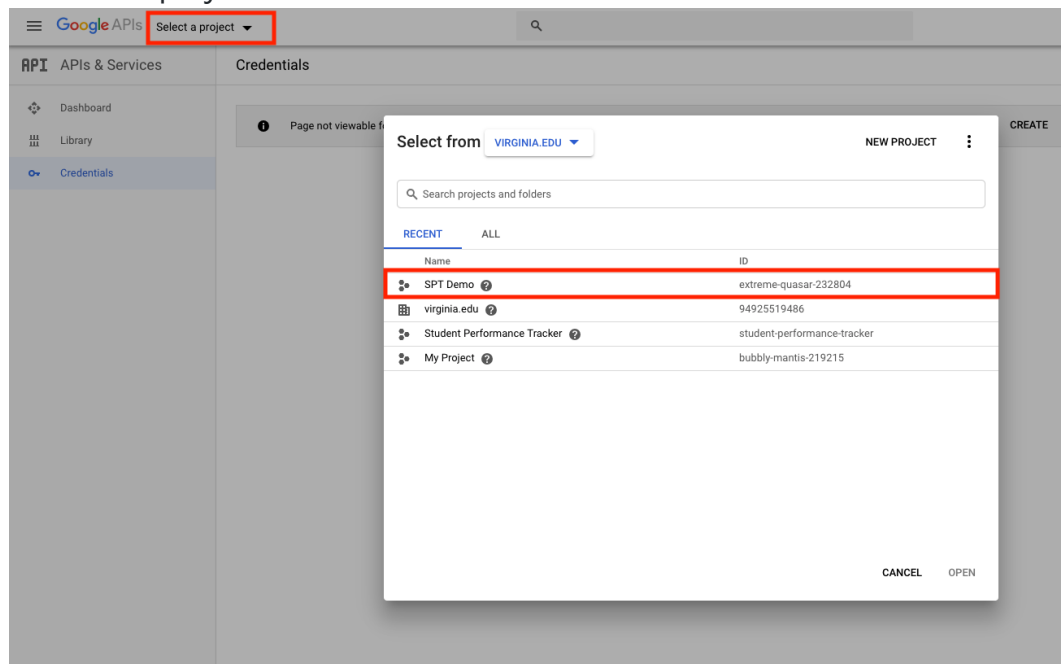
Location *
virginia.edu [BROWSE](#)

Parent organization or folder

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

○

4. Select the new project



○


5. Select the *OAuth Consent Screen* tab on the left menu

- Select *External*.


OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

☐ Internal 

Only available to users within your organization. You will not need to submit your app for verification.

☒ External 

Available to any user with a Google Account.



-
- Give the application a name
- Add the domain to the authorized domains list.
- Click save

7. Select the *Credentials* tab

- Click on *Create Credentials*
 - Select *OAuth client ID*
 - Select *Web Application*
 - Name it *SPT* or another name if desired

← Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

- ☒ Web application
- ☐ Android [Learn more](#)
- ☐ Chrome App [Learn more](#)
- ☐ iOS [Learn more](#)
- ☐ Other

Name ?

spt demo

Restrictions

Enter JavaScript origins, redirect URIs, or both [Learn More](#)

Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (<https://example.com/subdir>). If you're using a nonstandard port, you must include it in the origin URI.

<https://www.example.com>

Type in the domain and press Enter to add it

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

<https://www.example.com>

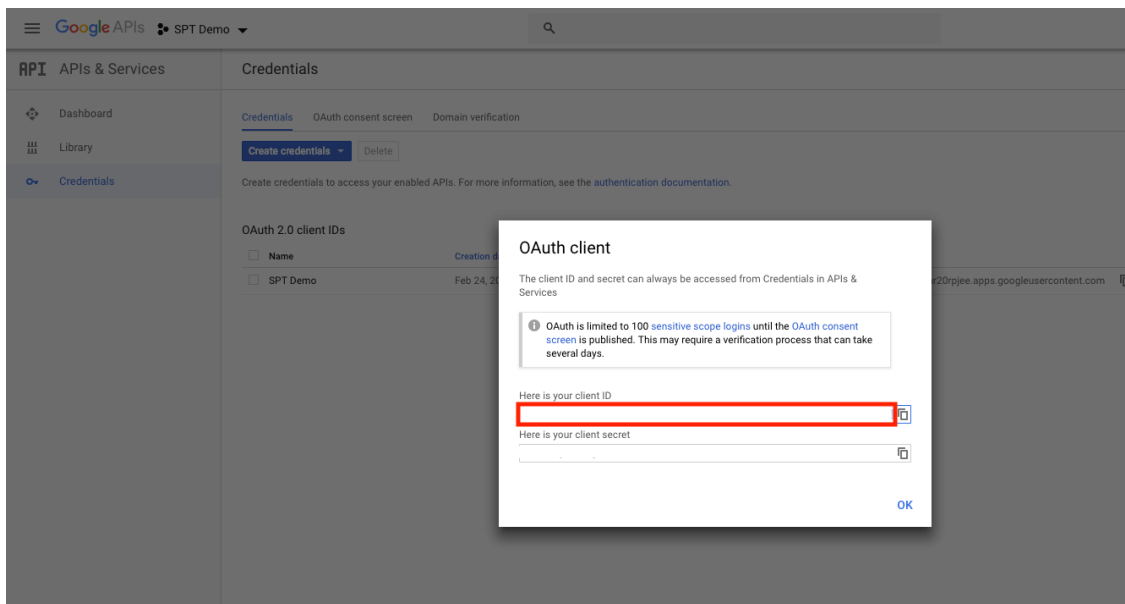
Type in the domain and press Enter to add it

Create

Cancel

-
- Add your domain to the *Authorized JavaScript Origins* and the *Redirect URLs*
 - If using the production deployment, use https, otherwise use http
- Click Save

8. You will be given a pop-up dialog, copy the *Client ID*



-
- Open the project in an IDE
 - navigate to [Auth.js](#)
 - `src/frontend/src/vuex/modules/Auth.js`
 - Edit the show line to include your Client ID

```
Auth.js
1 import api from '../api';
2 import store from '../index';
3 import axios from 'axios';
4 import { API_URL } from '@constants/';
5 import { DEBUG_TOKEN } from '@constants/';
6 //import { IN_PROD } from '@constants/';
7
8 const state = {
9   signedIn: false,
10  profile: null,
11 };
12
13 const actions = {
14   // Contact the google API
15   initGapi({ commit }) {
16     return new Promise((resolve, reject) => {
17       gapi.load('auth2', {
18         callback: () => {
19           gapi.auth2
20             .init({
21               client_id:
22                 '<paste client id here>',
23             })
24             .then(() => {
25               resolve();
26             });
27         },
28       });
29     });
30   },

```

- navigate to (../src/backend/sptApp/auth.py)
 - Edit the SERVER_CLIENT_ID to also match your Client ID