**Single Sign-On(SSO)**

What is SSO?

Single sign-on (SSO) is a [session and user authentication service](https://www.techtarget.com/searchsecurity/answer/What-are-the-most-common-digital-authentication-methods) that permits a user to use one set of login credentials -- for example, a name and password -- to access multiple applications. SSO can be used by enterprises, smaller organizations and individuals to ease the management of various usernames and passwords.

In a basic web SSO service, an [agent](https://www.techtarget.com/searchenterpriseai/definition/agent-intelligent-agent) module on the application server retrieves the specific [authentication](https://www.techtarget.com/searchsecurity/definition/authentication) credentials for an individual user from a dedicated SSO policy server, while authenticating the user against a user [repository](https://www.techtarget.com/searchoracle/definition/repository), such as a Lightweight Directory Access Protocol ([LDAP](https://www.techtarget.com/searchmobilecomputing/definition/LDAP)) directory. The service authenticates the end user for all the applications the user has been given rights to and eliminates future password prompts for individual applications during the same session.

How it is Work?

Single sign-on is a federated identity management ([FIM](https://www.techtarget.com/searchsecurity/definition/federated-identity-management)) arrangement, and the use of such a system is sometimes called identity federation. [OAuth](https://www.techtarget.com/searchapparchitecture/definition/OAuth), which stands for Open Authorization and is pronounced "oh-auth," is the framework that enables an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password.

OAuth acts as an intermediary on behalf of the end user by providing the service with an access token that authorizes specific account information to be shared. When a user attempts to access an application from the service provider, the service provider will send a request to the identity provider for authentication. The service provider will then verify the authentication and log the user in.

### Social SSO

Google, LinkedIn, Twitter and Facebook offer popular SSO services that enable an end user to log in to a third-party application with their social media authentication credentials. Although social single sign-on is a convenience to users, it can present security risks because it creates a single point of failure that can be exploited by attackers.

Many security professionals recommend that end users refrain from using social SSO services altogether because, once an attacker gains control over a user's SSO credentials, they will be able to access all other applications that use the same credentials.

Apple recently unveiled its own single sign-on service and is positioning it as a [more private alternative to the SSO options](https://www.techtarget.com/searchsecurity/news/252464636/Apple-single-sign-on-option-promises-privacy-for-users) provided by Google, Facebook, LinkedIn and Twitter. The new offering, which will be called Sign in with Apple, is expected to limit what data third-party services can access. Apple's SSO will also enhance security by requiring users to use 2FA on all Apple ID accounts to support integration with Face ID and Touch ID on iOS devices.

### Enterprise SSO

Enterprise single sign-on (eSSO) software products and services are password managers with client and server components that log the user on to target applications by replaying user credentials. These credentials are almost always a username and password; target applications do not need to be modified to work with the eSSO system.

### Advantages and disadvantages of SSO

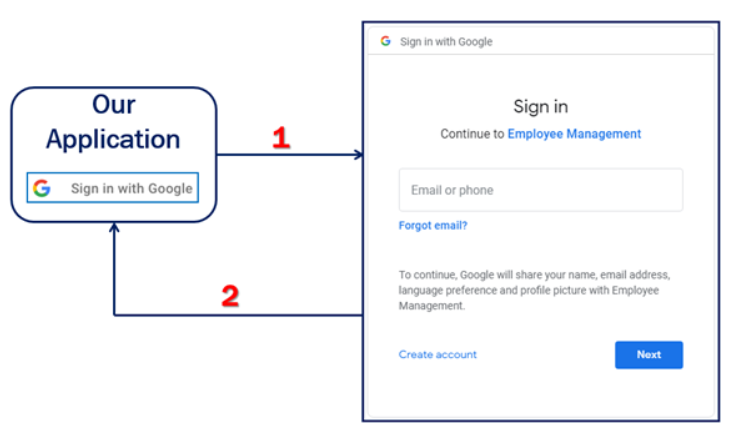
Advantages of SSO include the following:

* It enables users to remember and manage fewer passwords and usernames for each application.
* It streamlines the process of signing on and using applications -- no need to reenter passwords.
* It lessens the chance of phishing.
* It leads to fewer complaints or trouble about passwords for IT [help desks](https://www.techtarget.com/searchcustomerexperience/definition/help-desk).

Disadvantages of SSO include the following:

* It does not address certain levels of security each application sign-on may need.
* If availability is lost, then users are locked out of the multiple systems connected to the SSO.
* If unauthorized users gain access, then they could gain access to more than one application.

Overview Diagram



Environment Setup

You can download the Source code from **[here](https://github.com/drookey/Applications-with-ASP.Net).**

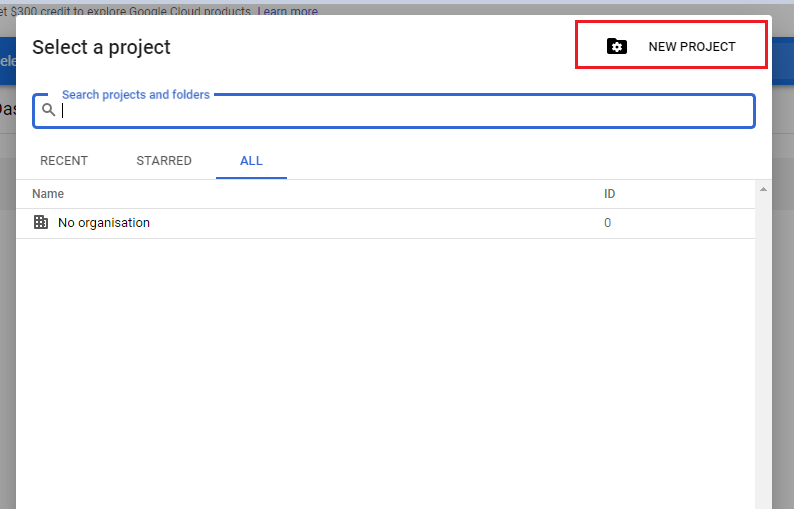
Create google OAuth credentials

we need to ClientId and Client Secret in your application for Google authentication. Navigate to the following website and login with your Google credentials [https://console.developers.google.com](https://console.developers.google.com/" \t "https://medium.com/c-sharp-progarmming/_blank) and then log in here with your G-Mail.

Once you log in, the first step creates a new project if you don’t have one already. at this moment, I don’t have any projects.so let's create a new project by clicking on Select a project.

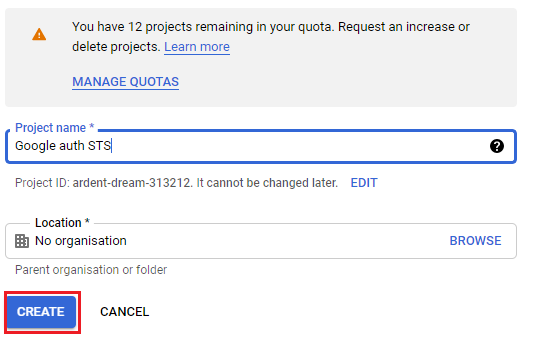


And then select the NEW PROJECT.



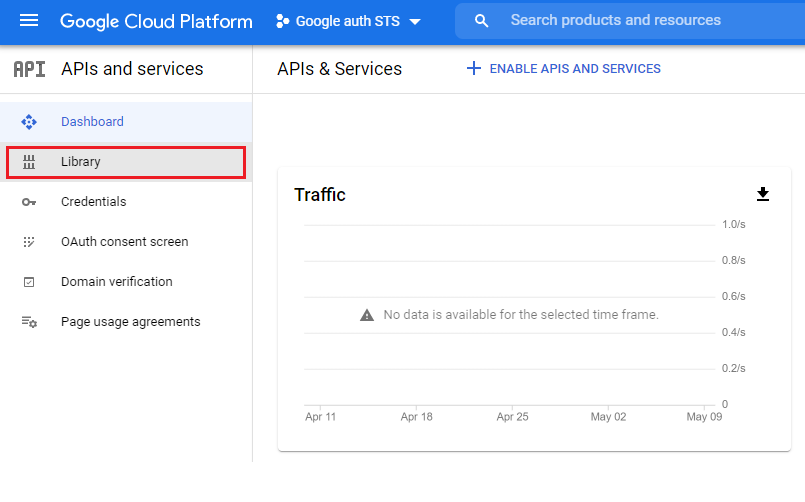
now give you a meaningful name for your project. I give my project name is Google auth STS. STS stands for Security Token Service. because we are using this project here, to provide Identity service for our .NET Core application.

Let's click CREATE to create this new project.

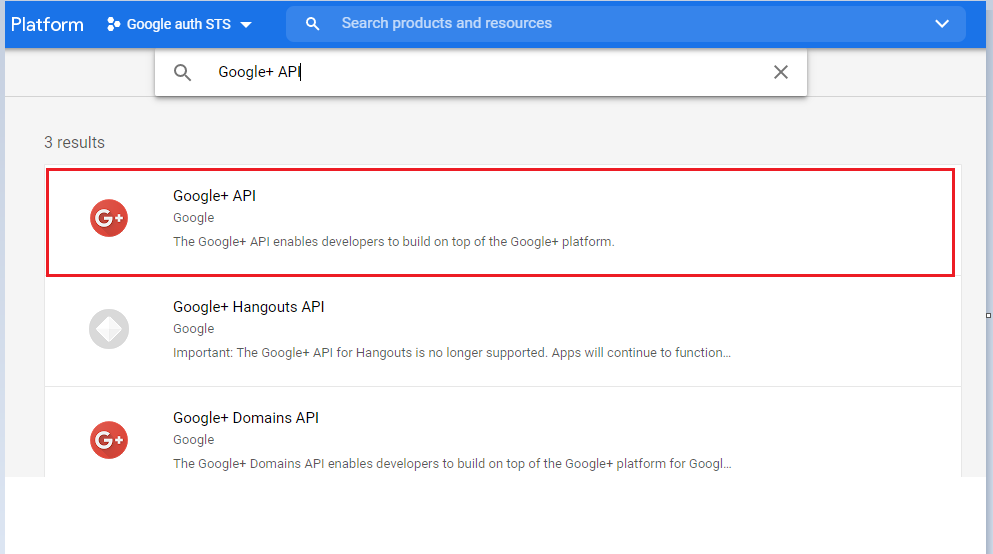


Once our new project is created, our next step is to enable Google class API.

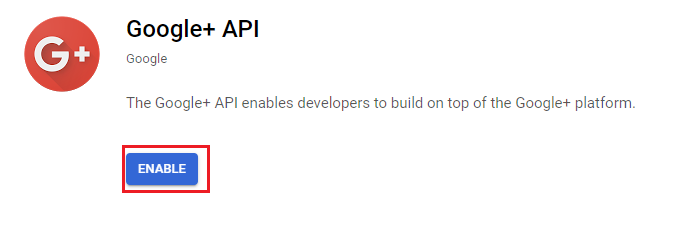
for that, click on this library navigation menu on the left. but some reason if you don’t see this navigation menu then click the Google APIs logo and then you will find this menu bar. Once you click on this library menu then.



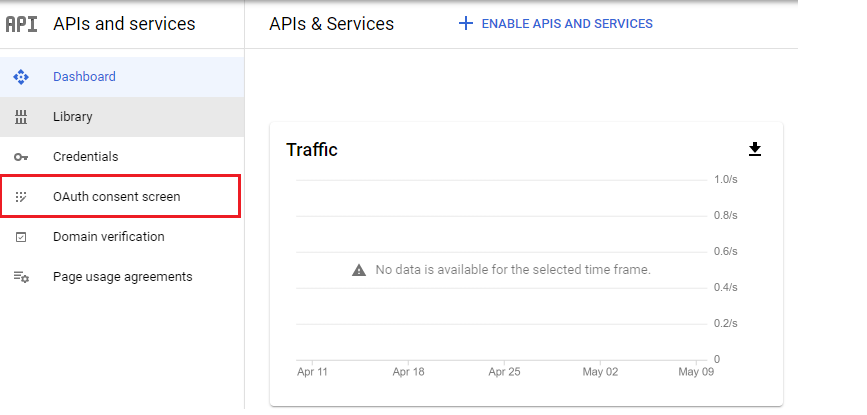
You get a search page and search here Google+ API and click on that.



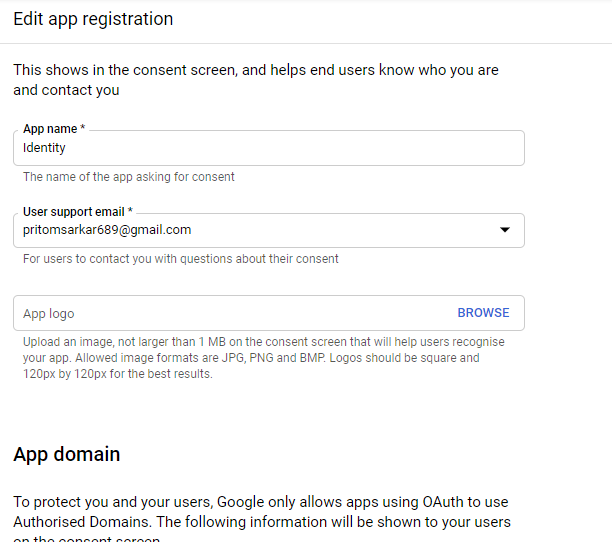
and then click the ENABLE button to enable this service.



Next, you need to configure the OAuth consent screen. If you do not see the OAuth consent screen tab, click on the Google APIs banner image on the top left-hand corner. now click on this OAuth consent screen.

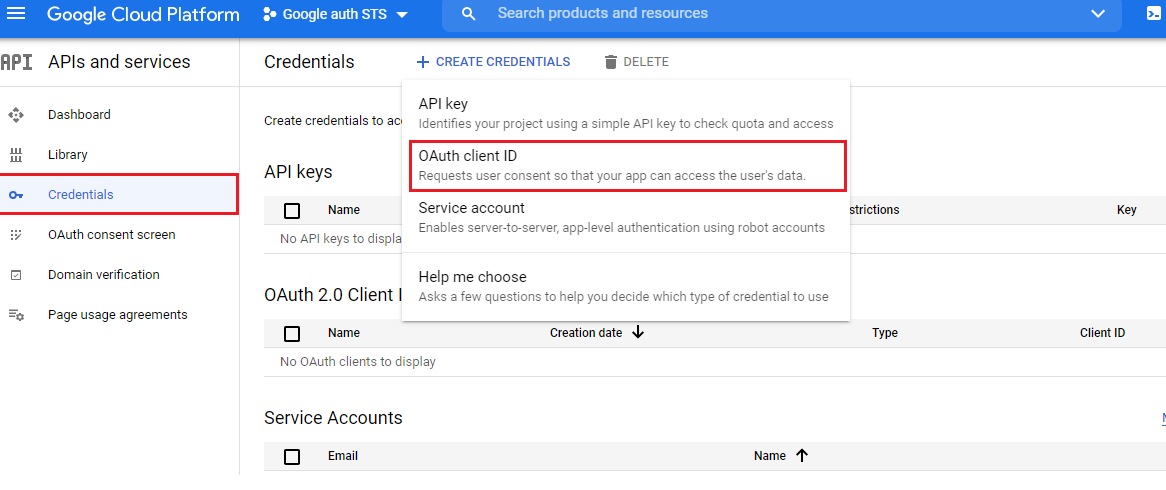


so now on this screen, the only required field is the Application name. this is the name shown to the user on the consent screen. at the moment we are creating these OAuth credentials for our .NET Core application.



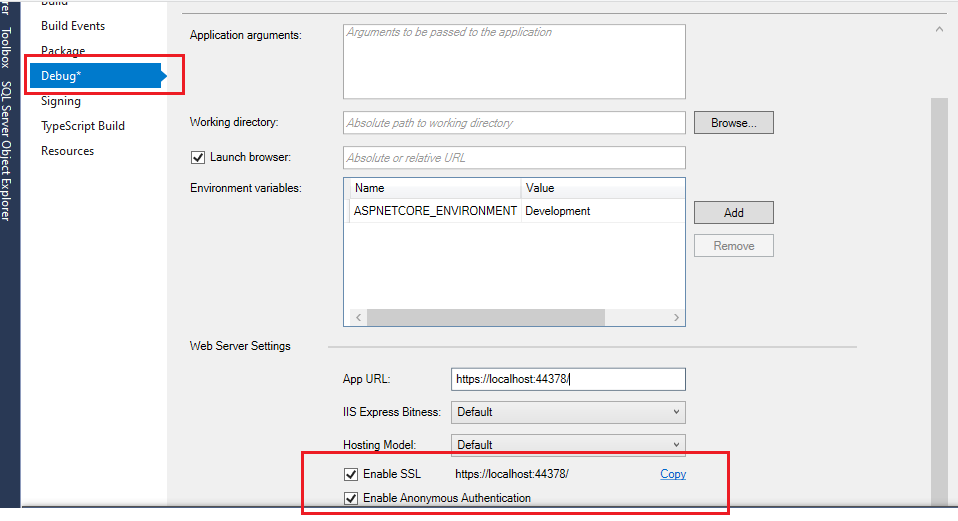
I have given above the app name, user support mail, and Developer contact information email address. Now scroll down and then you have to click Save & Continue.

and after that, you have to click the credentials navigation bar. and then CREATE CREDENTIALS=>OAuth client ID.



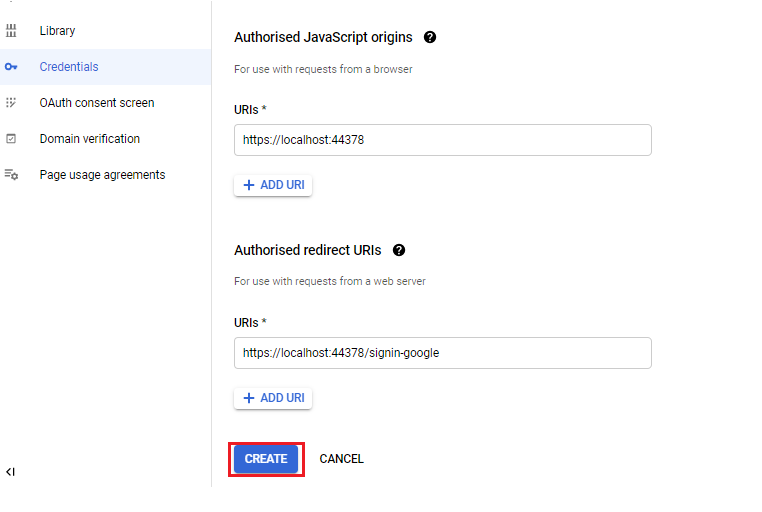
so now on this screen select Web Application. because that is the type of our client application. next, provide a name for the client. I gave this name Identity client. next Authorised JavaScript origins, which is the URI at which the client application is hosted.

at the moment our project still running on our local machine. so to get the URI at which our project hosted in the local machine, right-click in project name in the solution explorer and go to the properties window.https is more secure than http.at first, enable SSL.then copy this URL and paste it in the App URL text box. so that is the URL, at which our application is hosted.

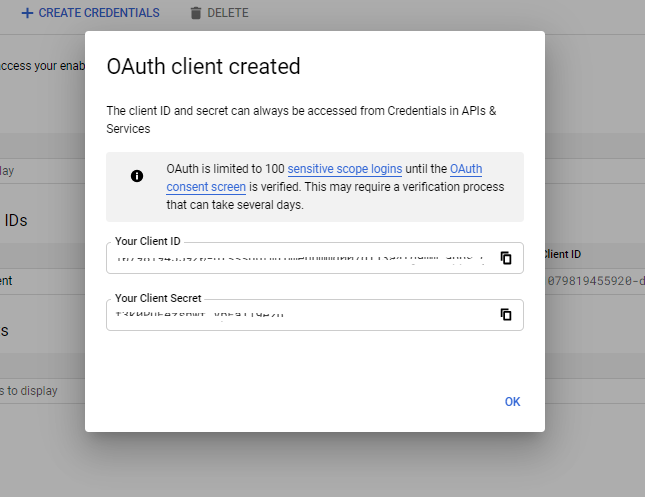


so let’s copy this URL and paste it Authorised JavaScript origins’ text box.

finally, you have to configure Authorised redirect URIs.this is the URI to which user is redirected to after they are authenticated by Google. and the default is the URI at which our application is hosted and this path signin-google and finally click the CREATE button.



So now you get the Client\_Id and Client\_Secret.



we need both of these IDs to integrate google authentication in our .NET Core project.

Setting up the UI for Google authentication.

so now you have to configure Google authentication for your .NET Core application. we do that in our startup class.