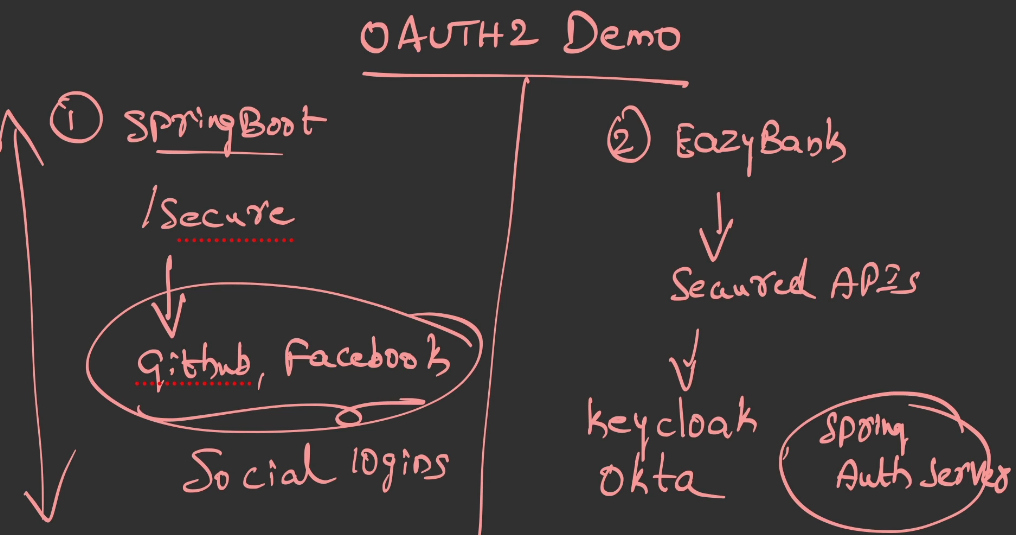
**13. OAUTH2 using spring security and social login – section14**

Introduction to the agenda of OAUTH2 implementations and demos



To show you the demo of OAUTH2, I'm going to take two different approaches.

Inside the very first approach, I'm going to build a Spring Boot application.

Inside the Spring Boot application, I'm going to have some secure pages. So to access these secure pages, I'm going to allow my end user to use the social logins like GitHub, Facebook. So using these social logins, my end user should be able to access my secure webpage inside my Spring Boot application.

The other approach is we already have the Easy Bank Spring Boot application.

Inside this Spring Boot web application, we have some secured APIs.

All these secured APIs, I'm going to protect them with our own Auth server.

So I'm going to build an Auth server with the help of Keyclock.

Apart from Keyclock, I'm also going to use the newly introduced Spring security Auth server. Using Spring Security Auth Server project, we should be able to build our own Auth server from scratch.

The reason why I'm taking these two different approaches is inside the very first approach, we are going to use the social logins. So whenever we are using the social logins in this scenario, we are never going to build our Auth Server.

We are just going to leverage the Auth server belongs to other organizations,

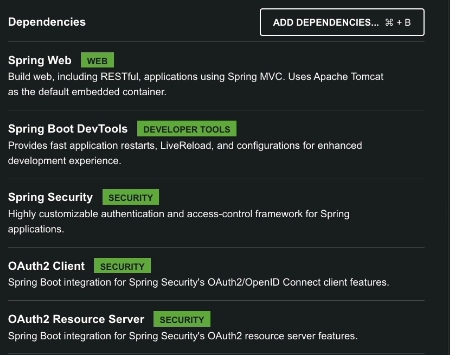
and this approach will work only for simple applications like blog applications

or some single page applications where they don't have lot of business functionality.

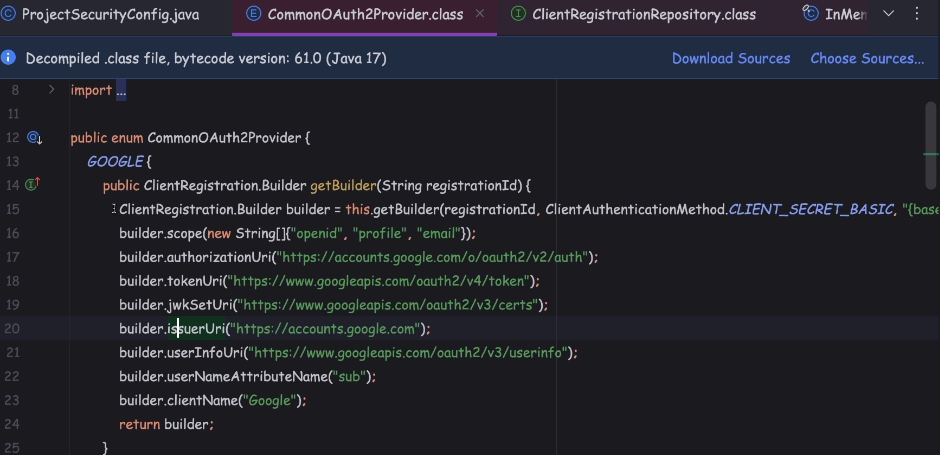
Whereas for real enterprise applications, like a bank application or a insurance application, for all such applications, the organization, they have to build their own Auth server either by leveraging the products inside the market like Keyclock,

Okta, or they can build their own Auth server with the help of Spring based Auth server library.

Demo of OAUTH2 using social logins - Part 1



Demo of OAUTH2 using social logins - Part 2



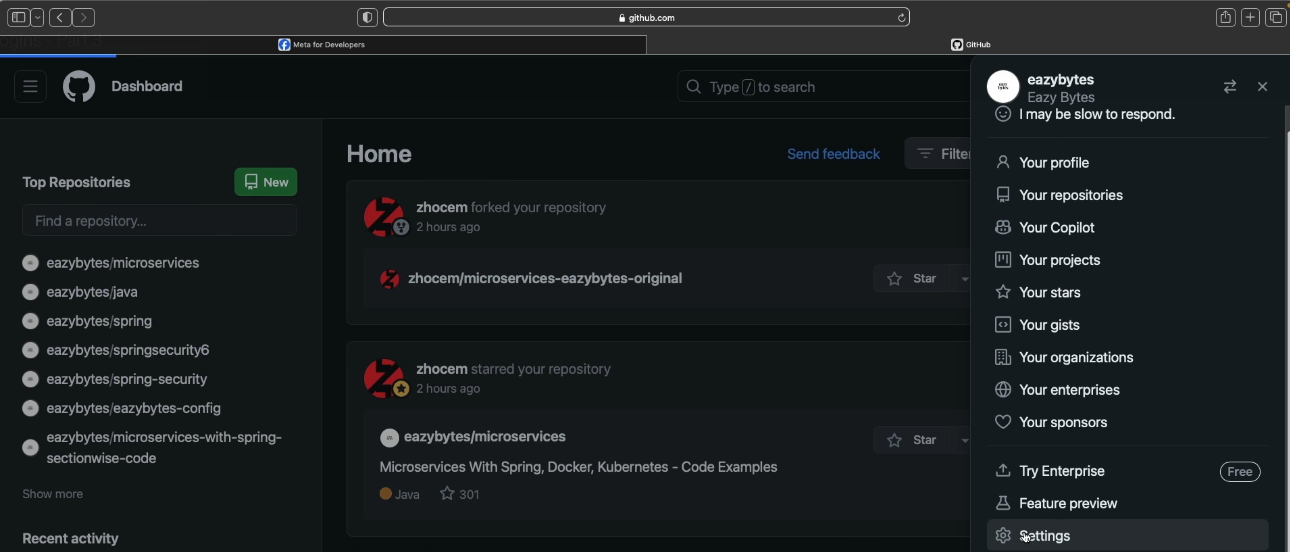
But unfortunately my application will not start, because the clientId's and client secrets are empty inside our configurations. So to use these auth servers

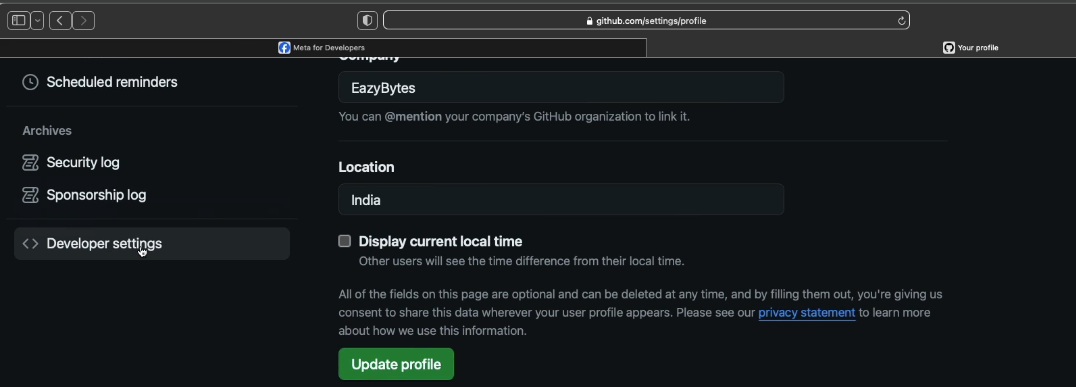
inside our OAuth 2.0 login flow, we need to visit these auth servers,

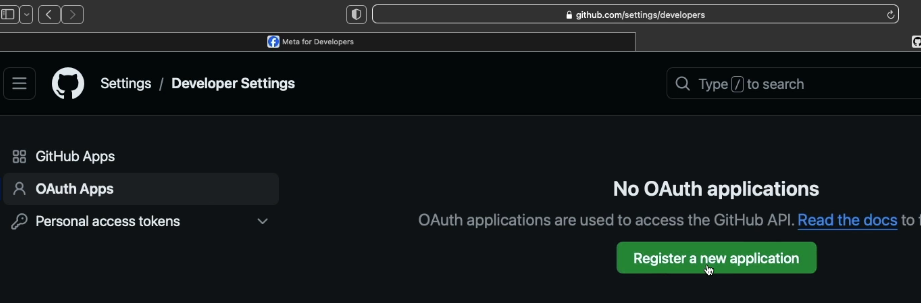
like GitHub and Facebook. And inside these auth servers, we need to register our application. And once the registration is completed, we're going to get the client ID and client secret, the same we need to mention here.

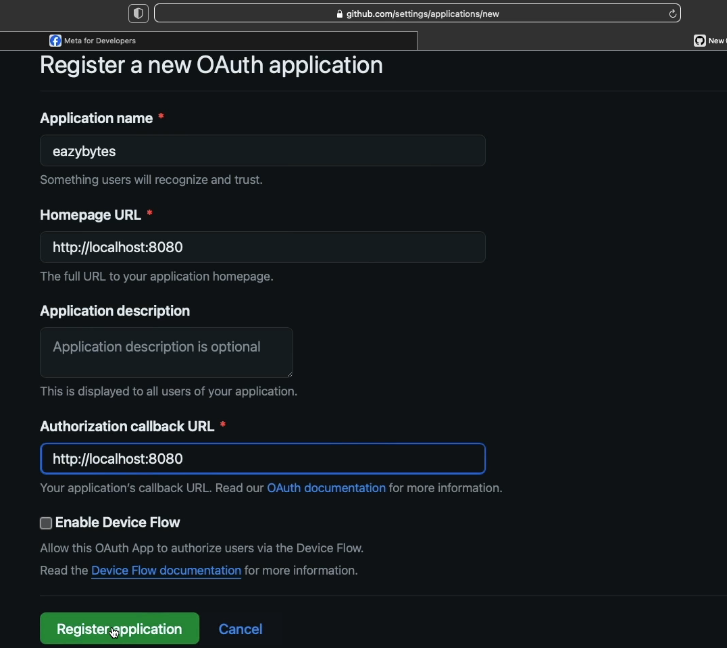
Demo of OAUTH2 using social logins - Part 3

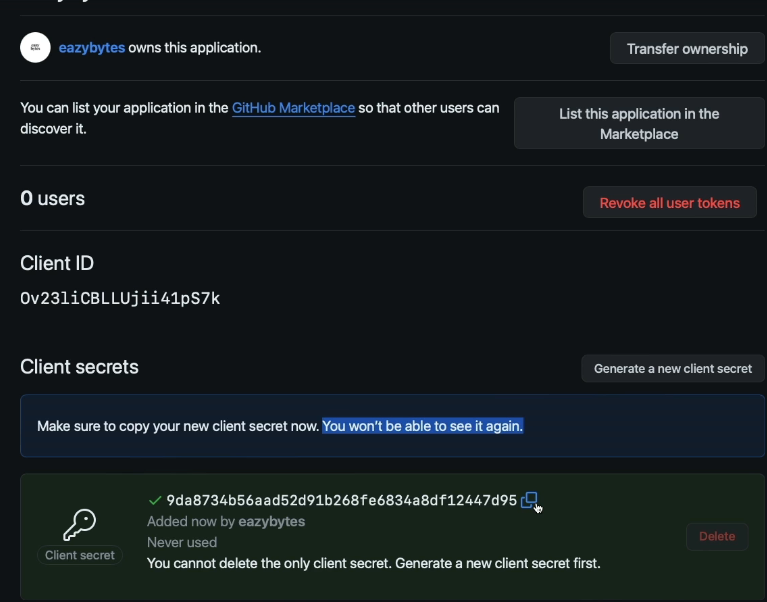
Github Oauth Registration: Login to github and under userprofile, click settings



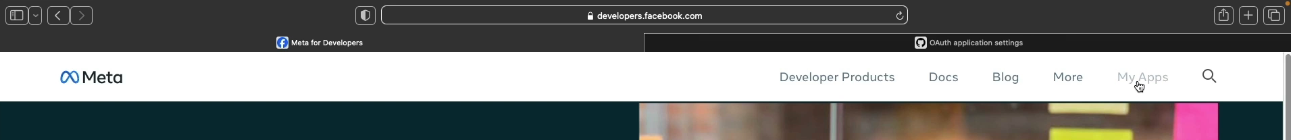


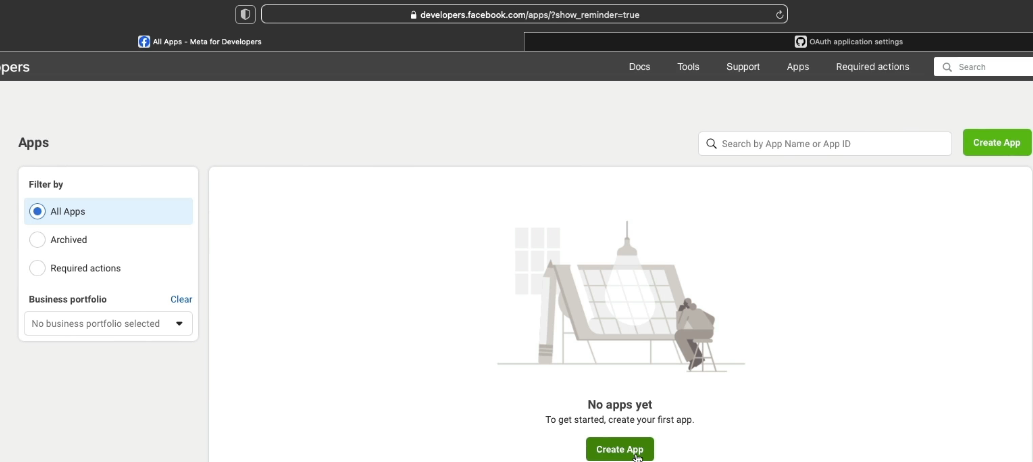


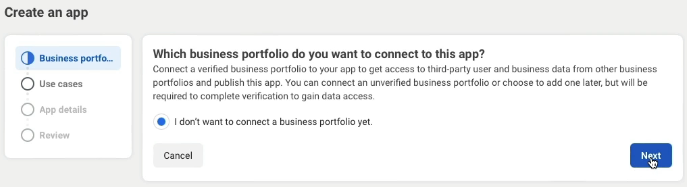


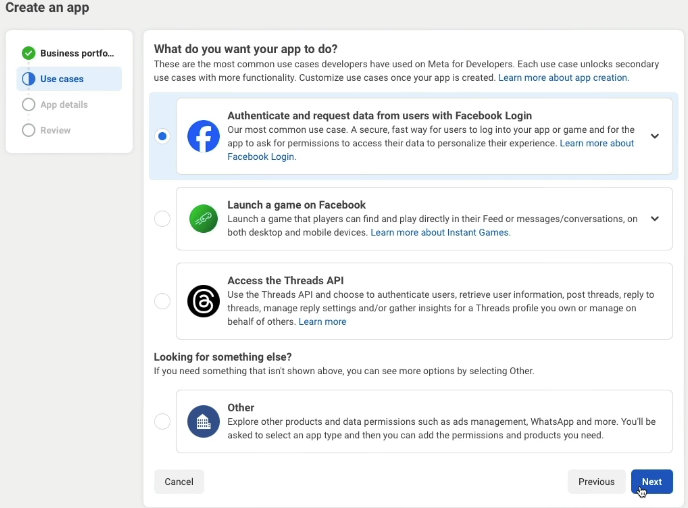


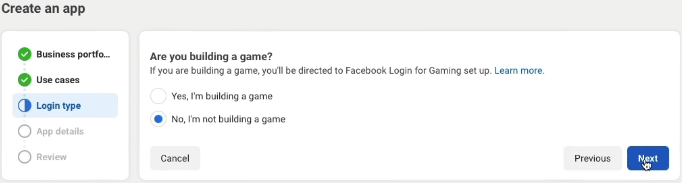
Oauth Registration with facebook:

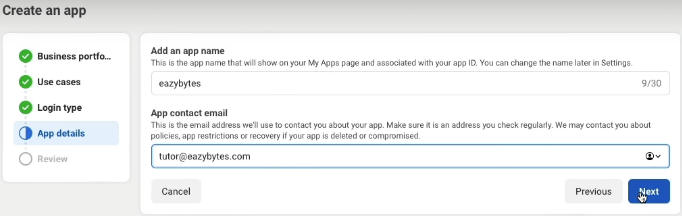


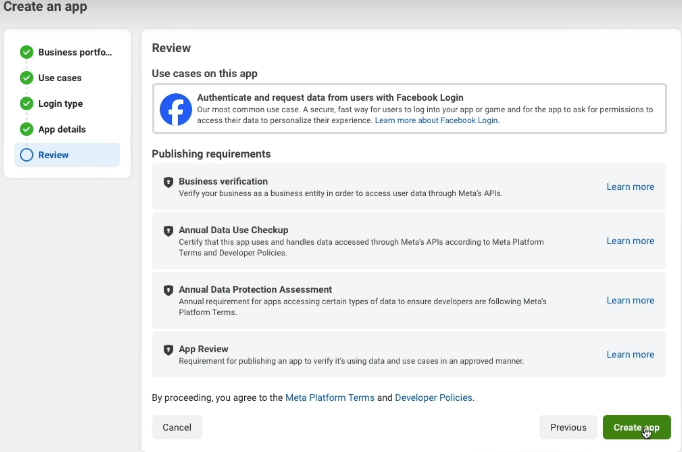




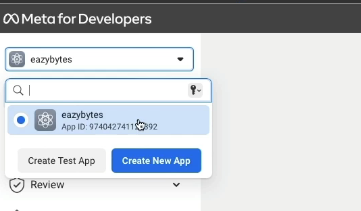


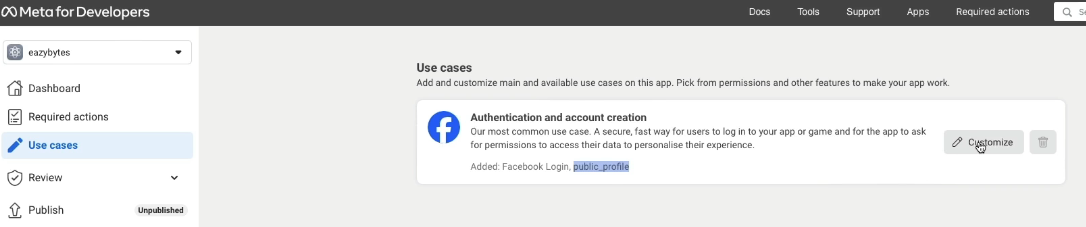






App will be created in the name of eazybytes





So here, apart from public\_profile scope, I also want to enable the email scope.

So here they're calling Permissions, but in OAuth terminology we should call it as scope. So to enable this, I need to click on this Add button. If you don't enable this email permission or email scope, then your code is not going to work.

The reason is very simple. If you go to these FACEBOOK enum, here inside the framework, they have configured scope as public\_profile and email.

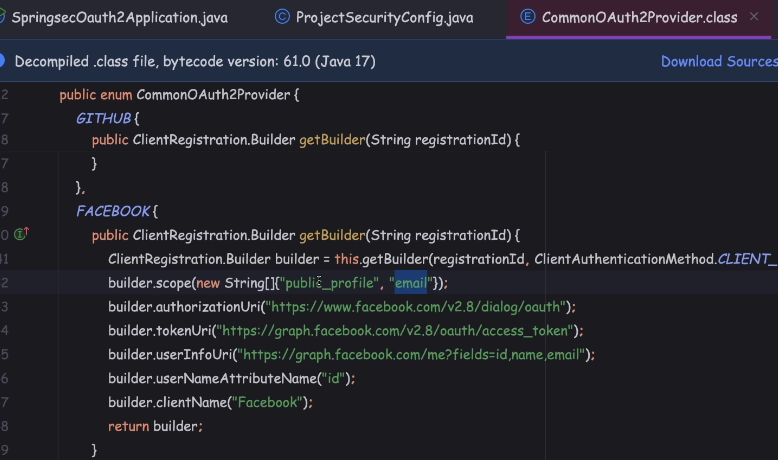
So whenever a request is going from your application to the Facebook auth-server,

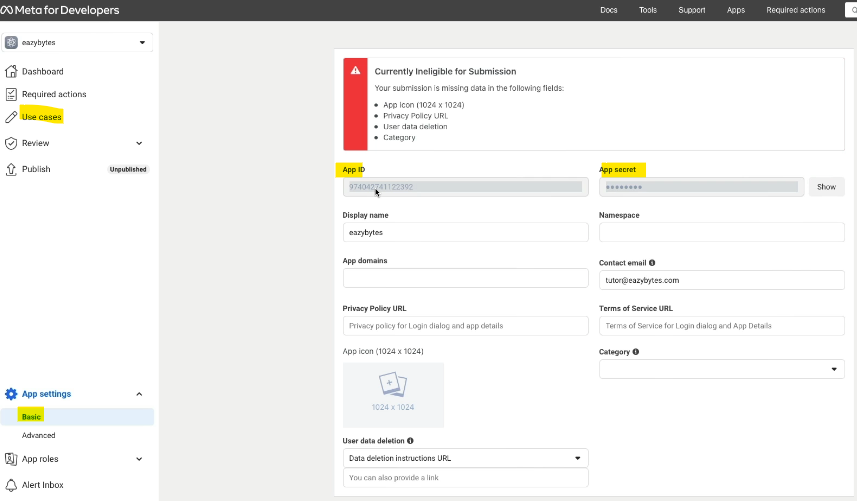
your application is going to request for both of these scopes.

So if you don't enable both these scopes, you're going to get an error.

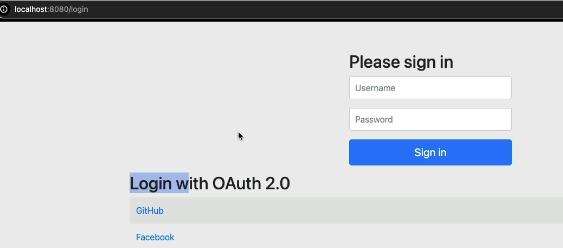
That's why please make sure to enable the email permission as well.

By default, public\_profile permission is anyway going to be enabled by Facebook.





So whatever application ID we got from the Facebook, it is going to act as a clientId here.



Inside the browser, I'm going to access the page, which is localhost:8080/secure.

So this is a secure path. As soon as I refresh this page, you'll be able to see I'm going to get a login page. Inside the login page, I have a normal login option.

Apart from normal login option, I also have an option, which is login with OAuth 2.0.

Inside these OAuth 2.0 options, I have two different options that I can select.

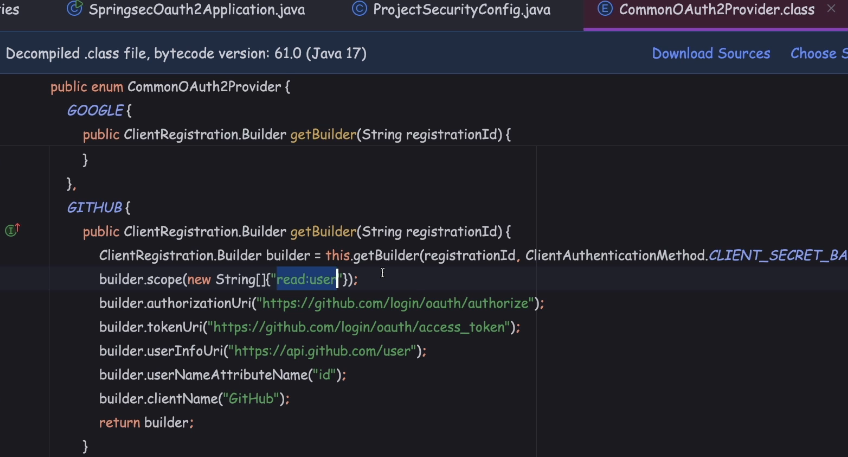
The very first one is GitHub, followed by Facebook.

So here I'm trying to rely on this default login page. But if you're building a enterprise application, you need to build your own custom login page with the similar kind of buttons. And whatever action or URL is going to be invoked from this button,

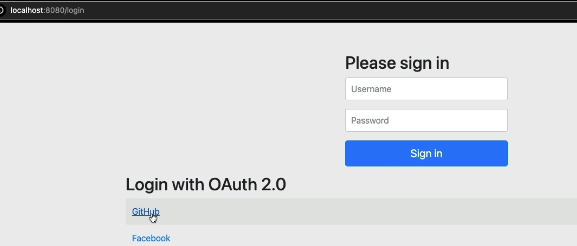
the same, you need to configure inside your custom HTML buttons as well.

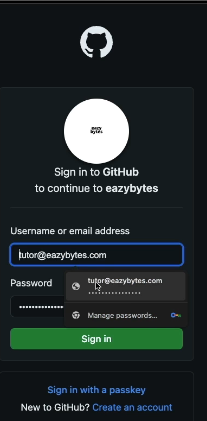
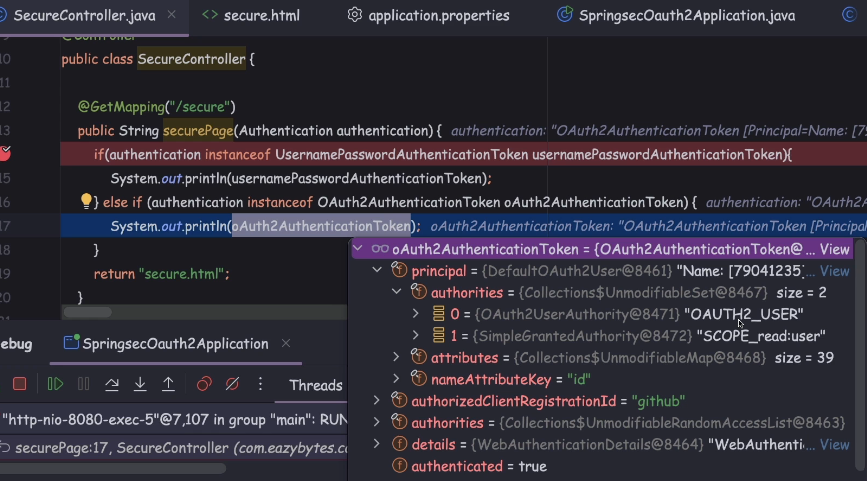
So if you try to see the source code of this HTML page, you'll get a clue around this.

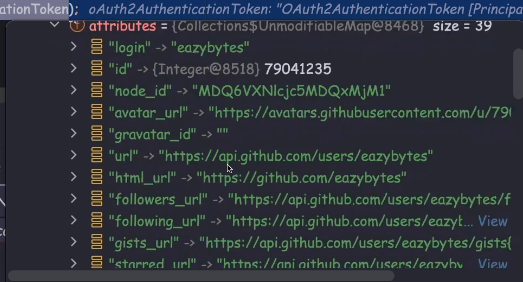
Demo of OAUTH2 using social logins - Part 4



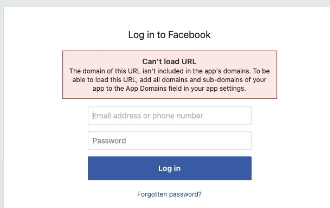
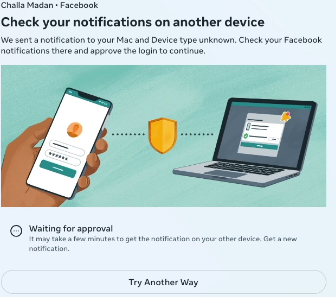
Demo github: try to access the path which is localhost:8080/secure

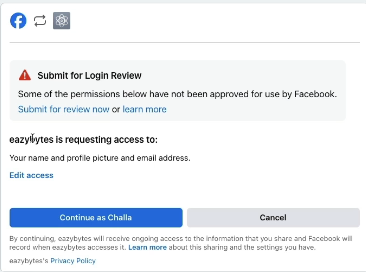




Demo facebook



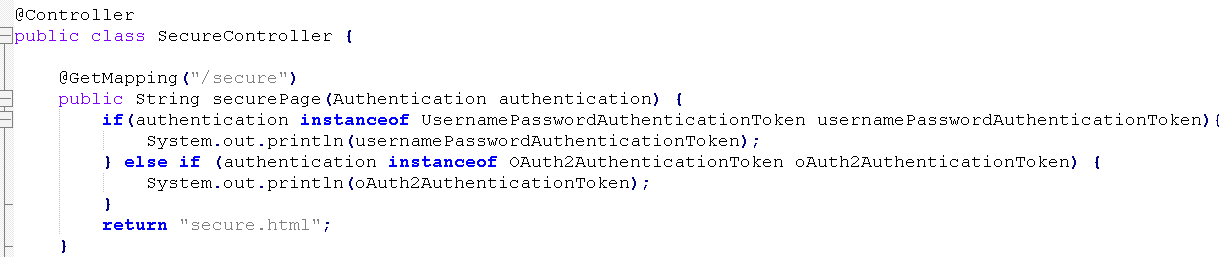
Demo of OAUTH2 using social logins - Part 5

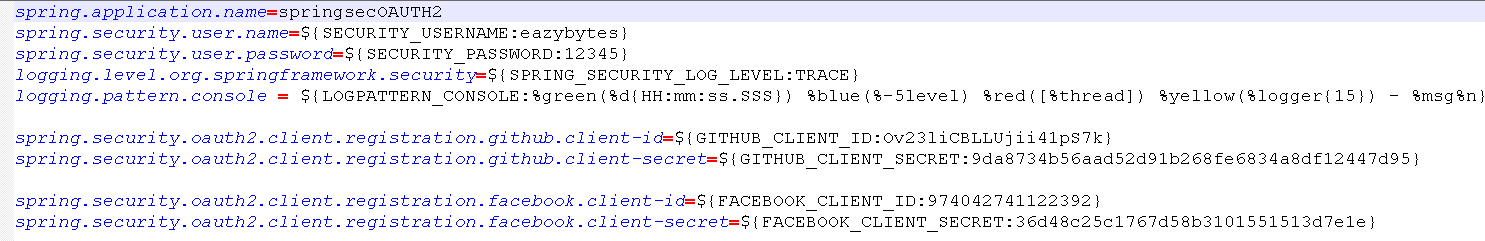
inside our Spring Boot web application, we tried to configure the OAUTH2 related configurations by creating the ClientRegistrationRepository. And to this being, we are trying to provide the configurations of GitHub Auth Server and Facebook Auth Server.

So this is one style of configuration.

There is also another easy configuration we can do in application.properties







Top of Form

Which of the following SpringBoot starter project dependency should be used to integrate with OAuth2 as a client ?

<dependency>

* 1. <groupId>org.springframework.boot</groupId>
  2. <artifactId>spring-boot-starter-oauth2-client</artifactId>
  3. </dependency>

Top of Form

Which of the following enum class can be used to configure most famous social login Auth providers like Google, GitHub, FaceBook etc ?

**CommonOAuth2Provider**