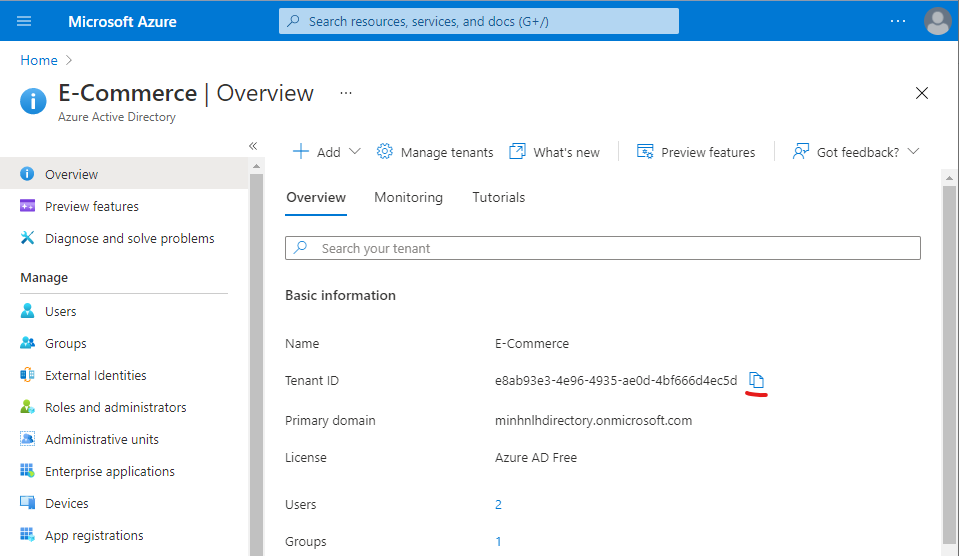
Using Azure AD in Spring Boot

# Create an app using Spring Initializr

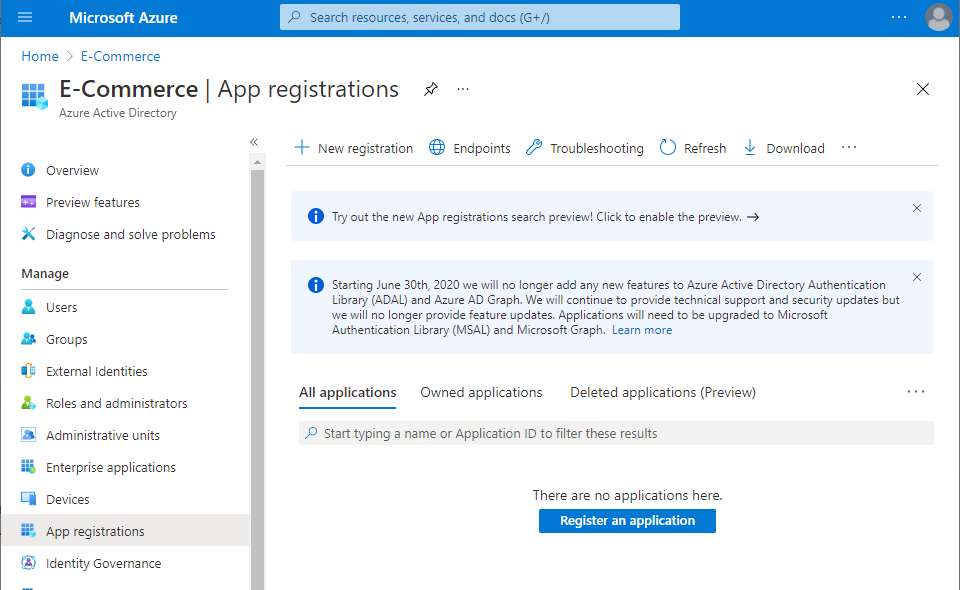
1. Browse to <https://start.spring.io/>.
2. Specify that you want to generate a Maven project with Java, enter the Group and Artifact names for your application.
3. Add Dependencies for Spring Web, Azure Active Directory, and OAuth2 Client.
4. At the bottom of the page, select the GENERATE button.
5. When prompted, download the project to a path on your local computer.

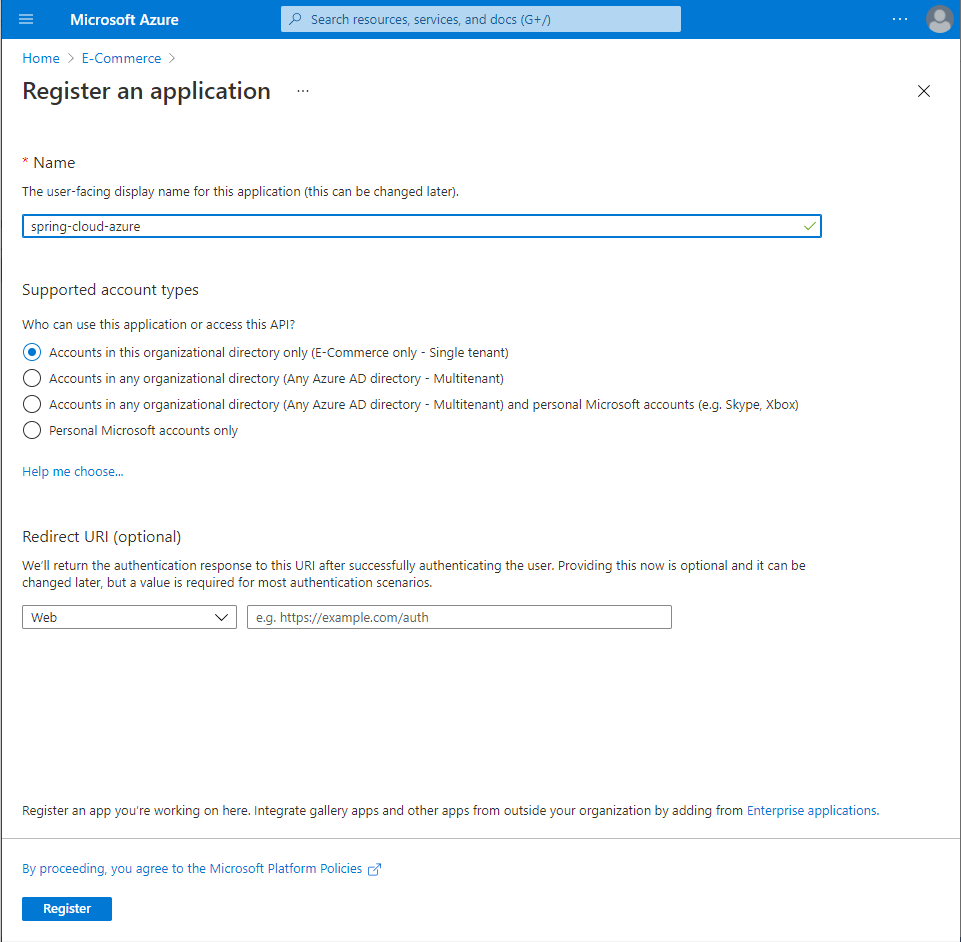
# Create Azure Active Directory instance

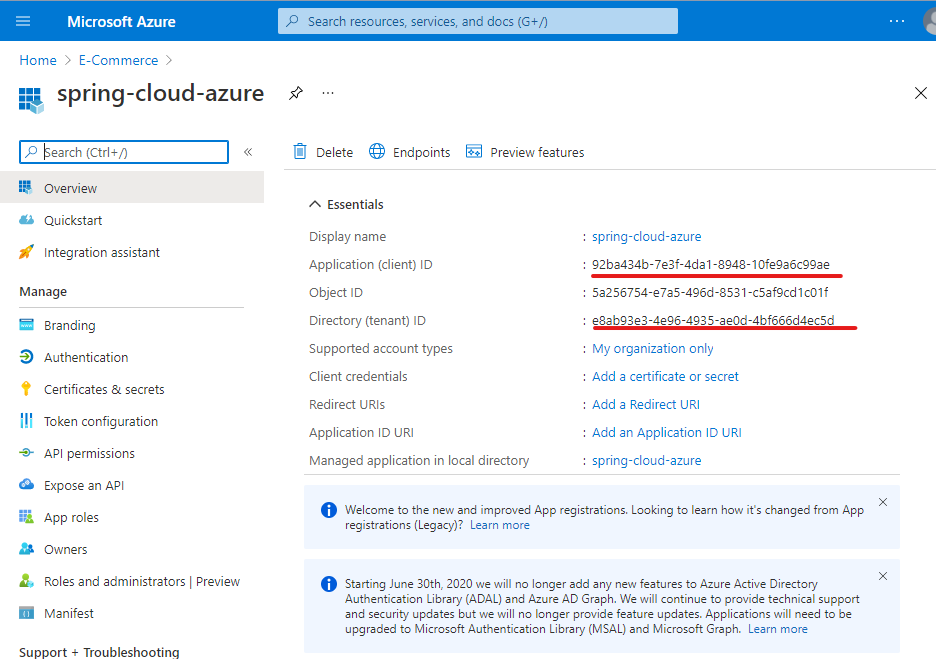
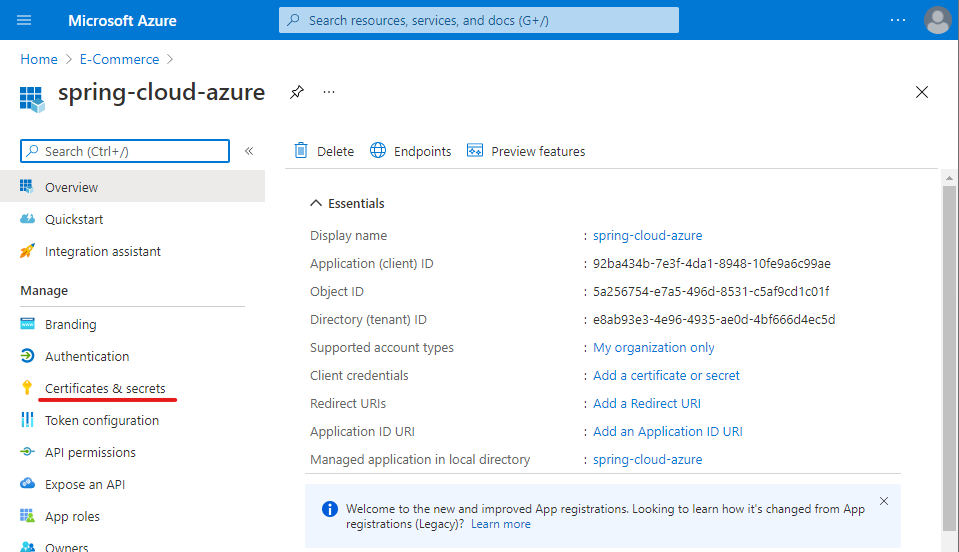
## Create the Active Directory instance

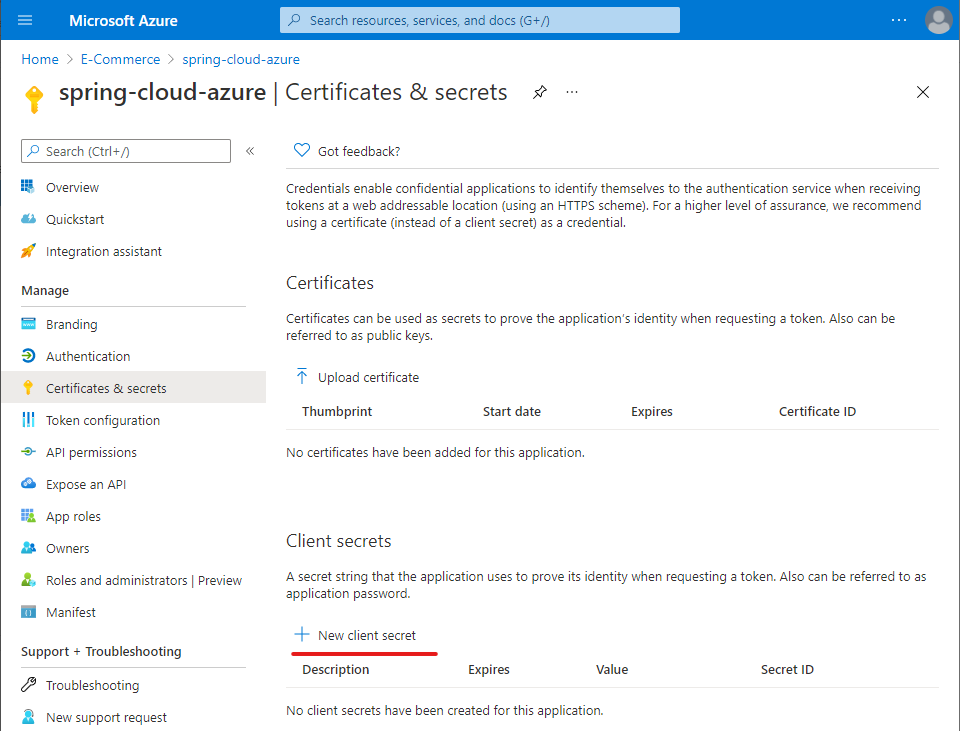
1. Log into https://portal.azure.com.
2. Select All services, then Identity, and then Azure Active Directory.
3. As default, your account will have Default Directory as the Default Tenant. If you want to create a new one, Click Manage tenants -> Create. Choose your names, click Review and Create -> Create
4. In your tenant dashboard, copy your Tenant Id and note it somewhere, we use this value later in application config

## Add an application registration for your Spring Boot app

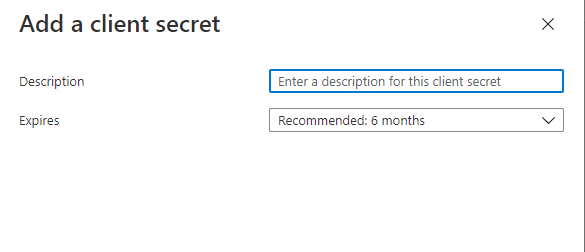
1. From the portal menu, select App registrations, and then select Register an application.
2. Choose application name -> then Register

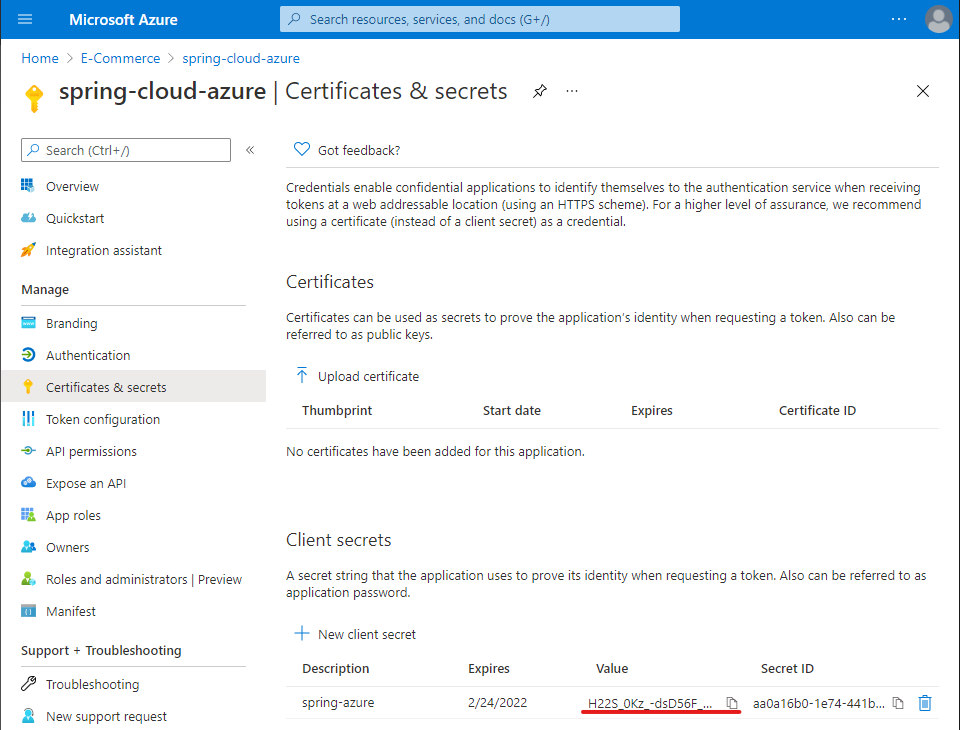
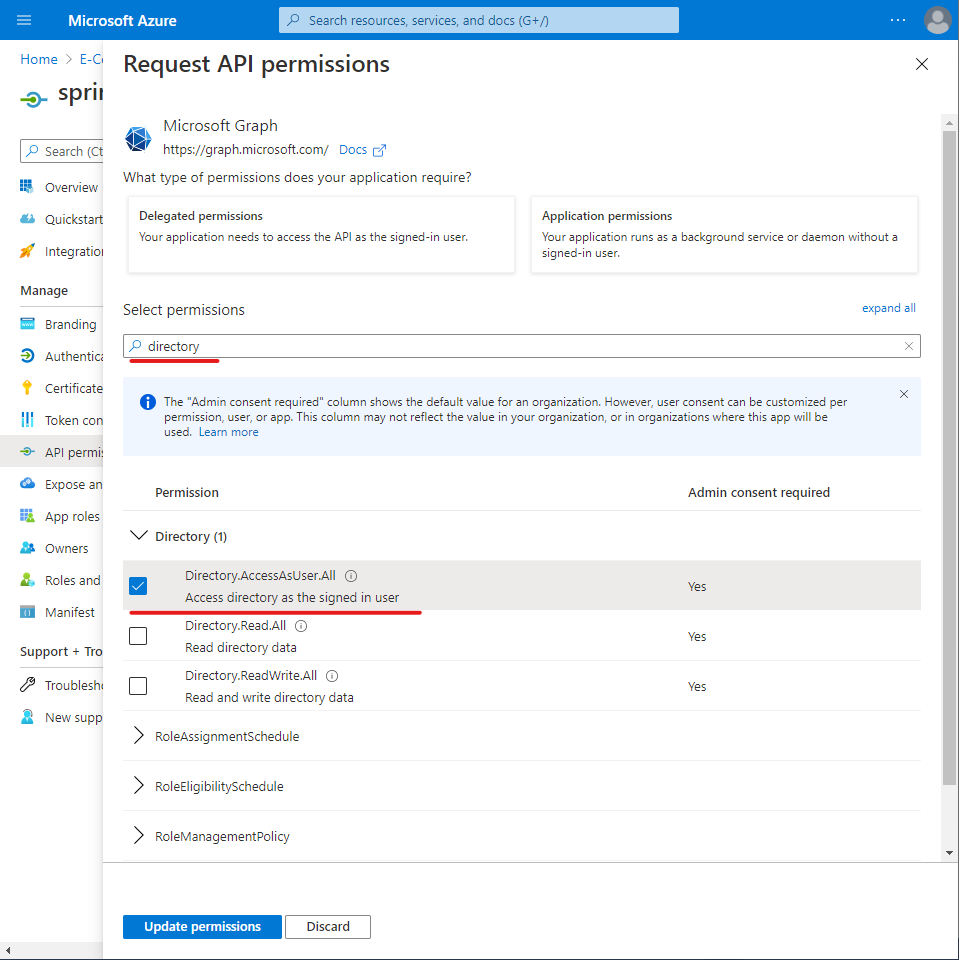
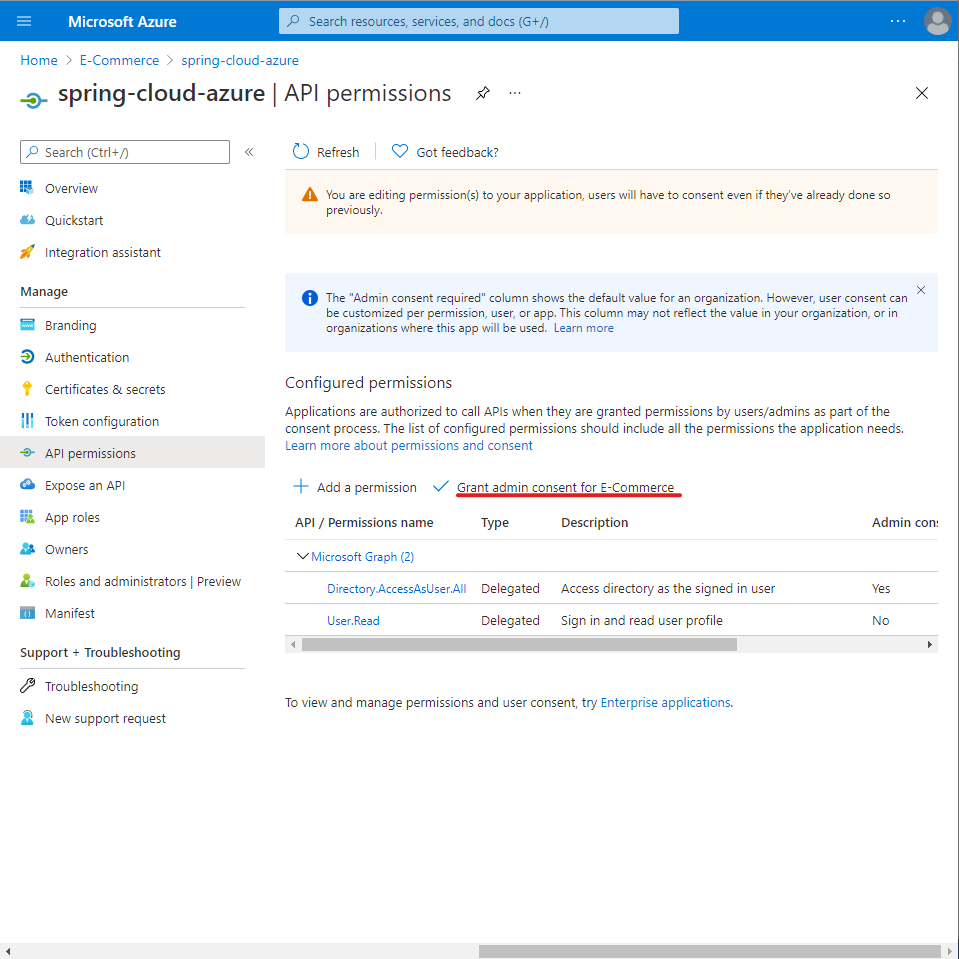
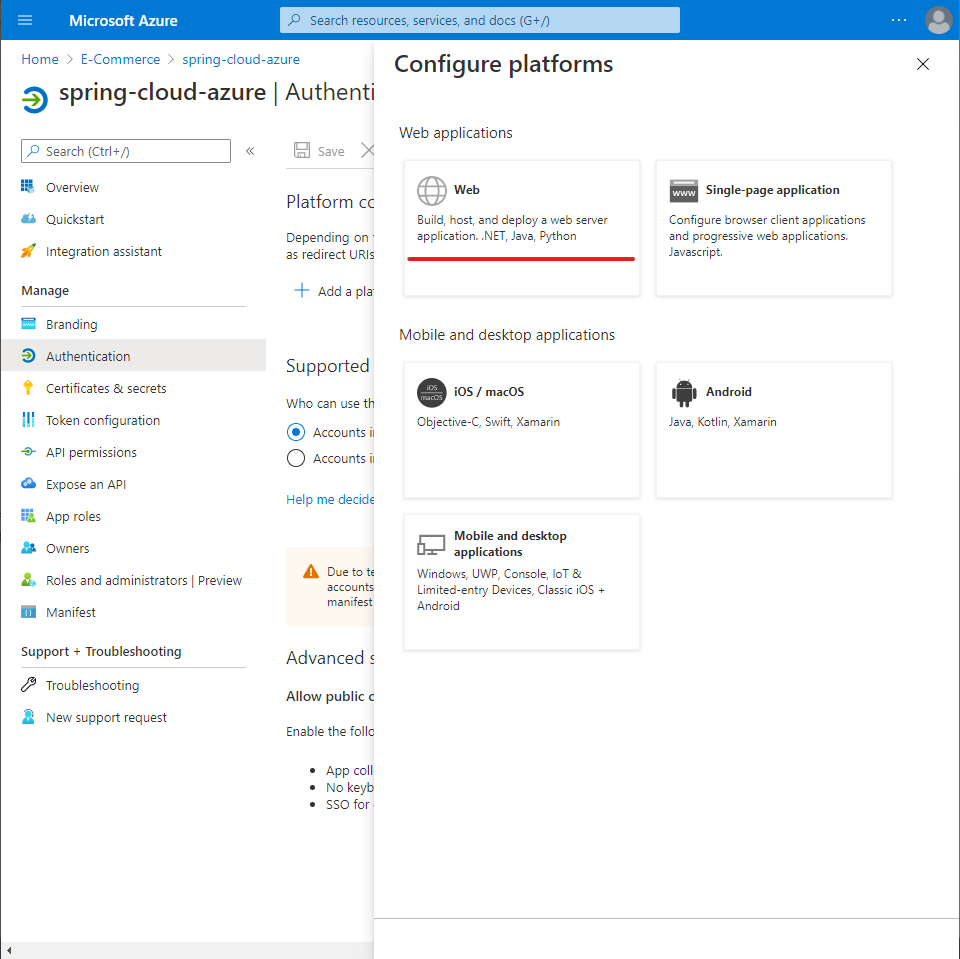
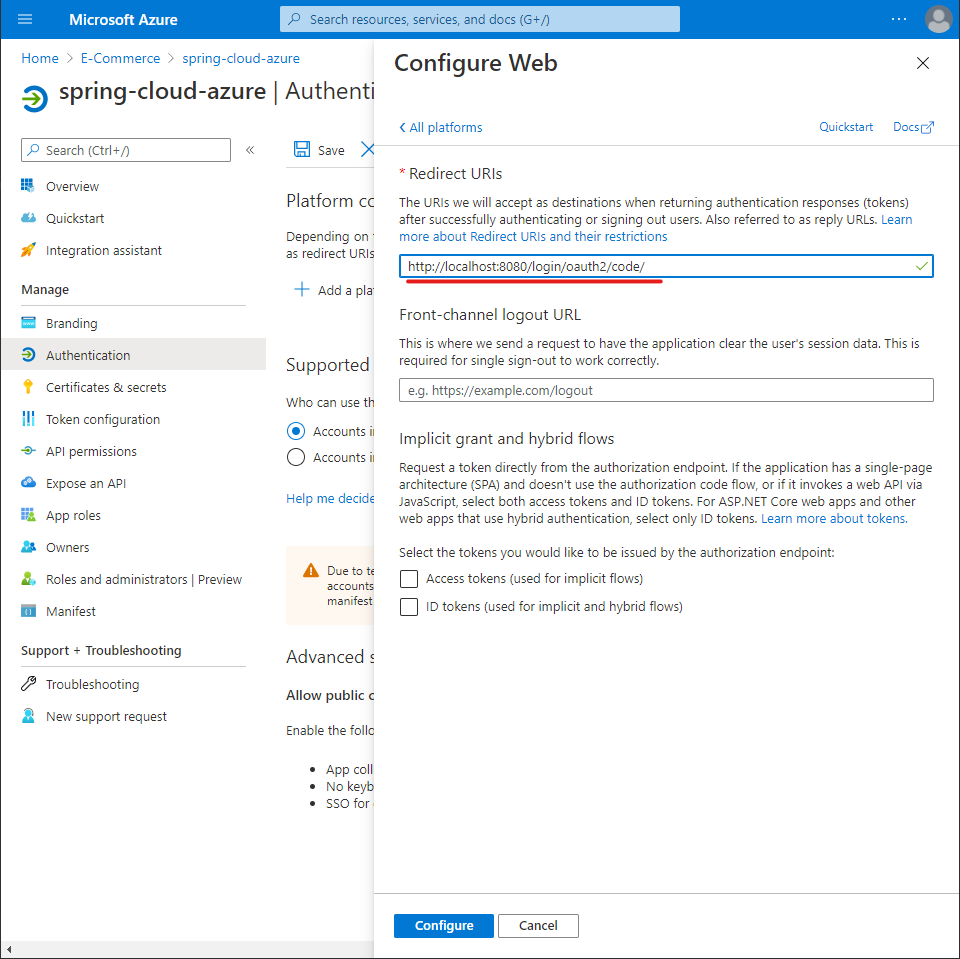


1. Note down Application ID and tenant ID:
2. Select Certificates & secrets in the left navigation pane. Then select New client secret.

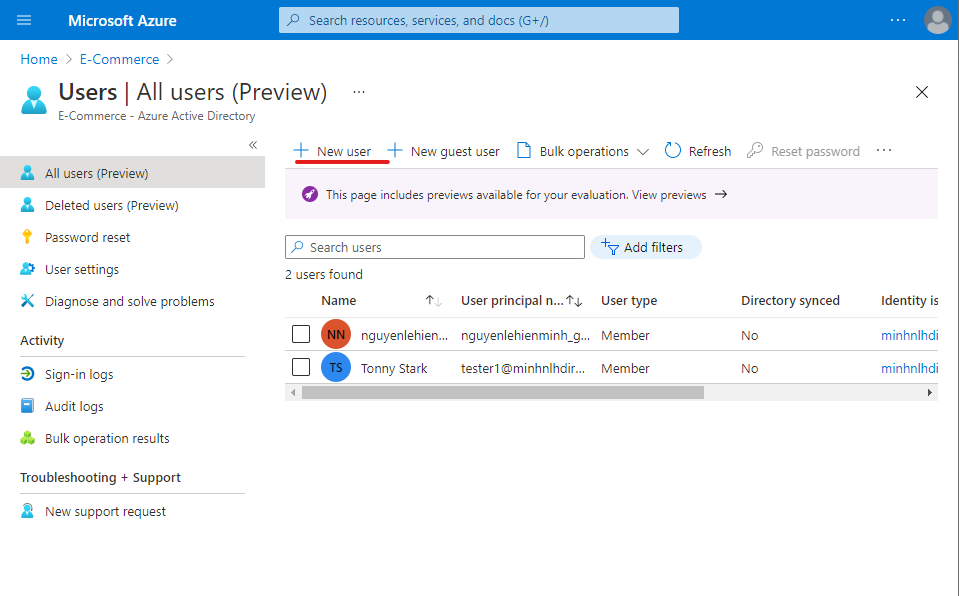
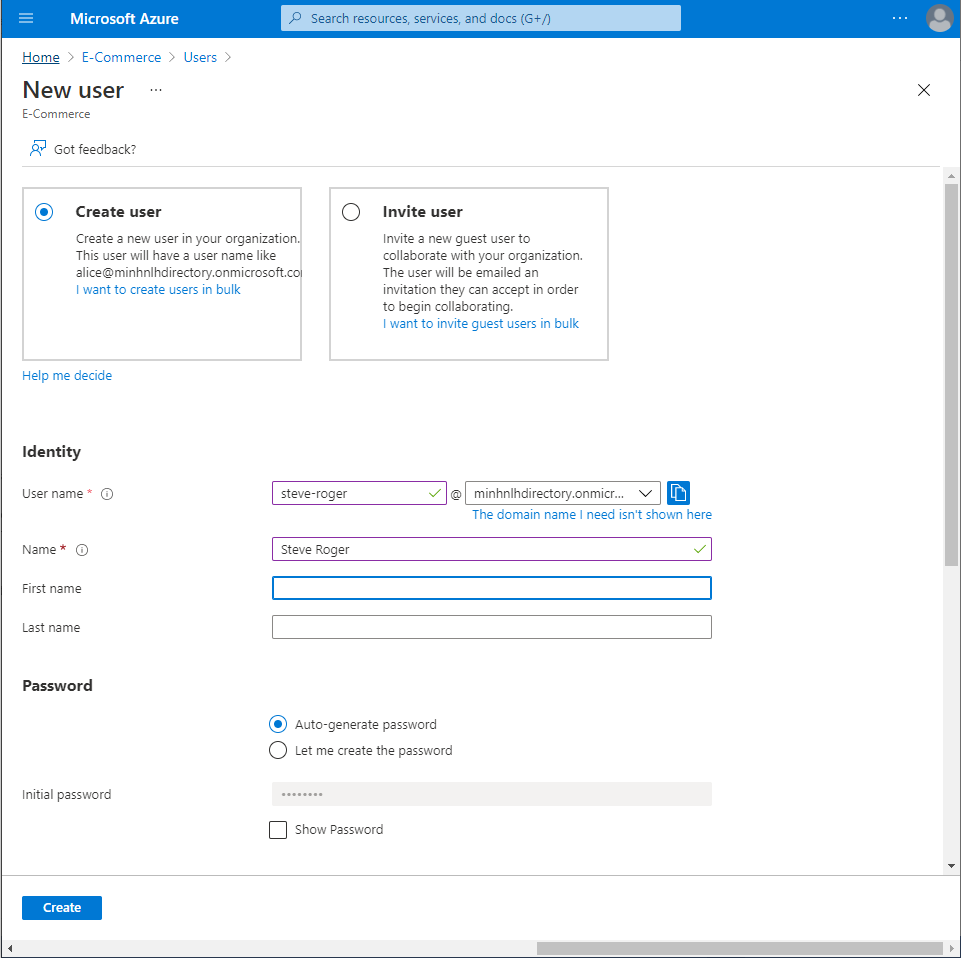
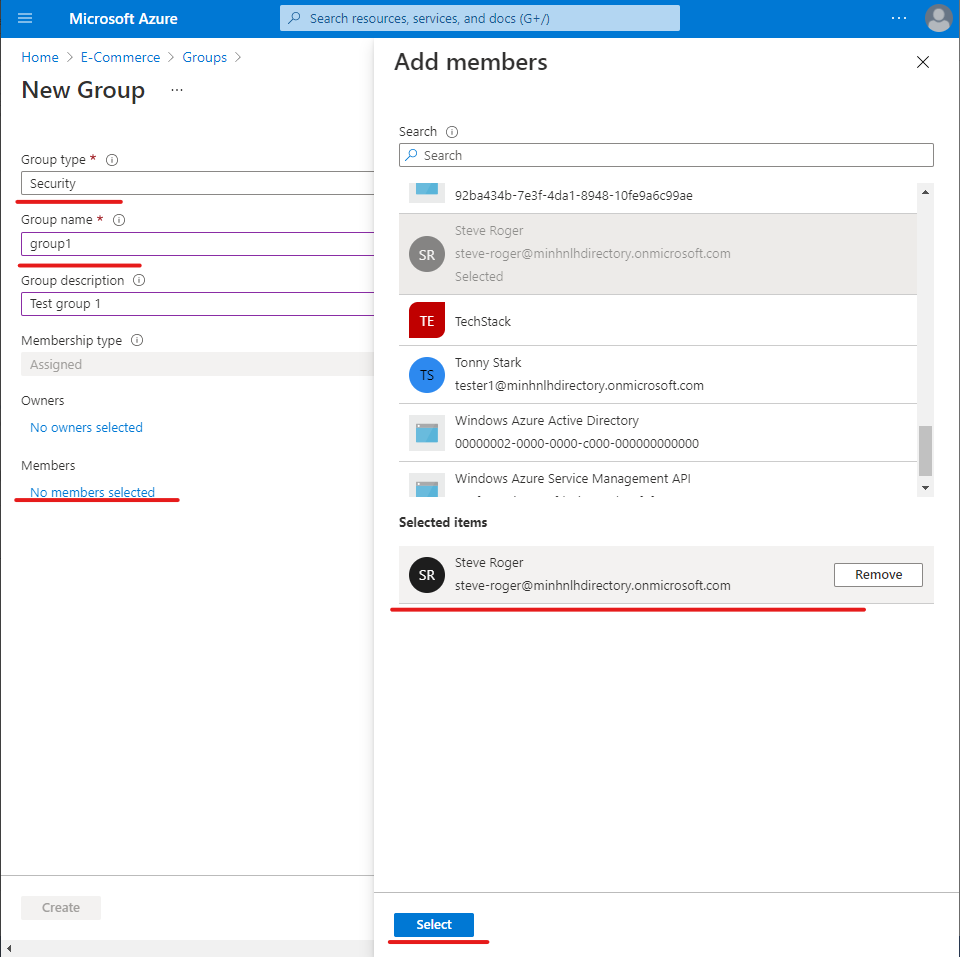
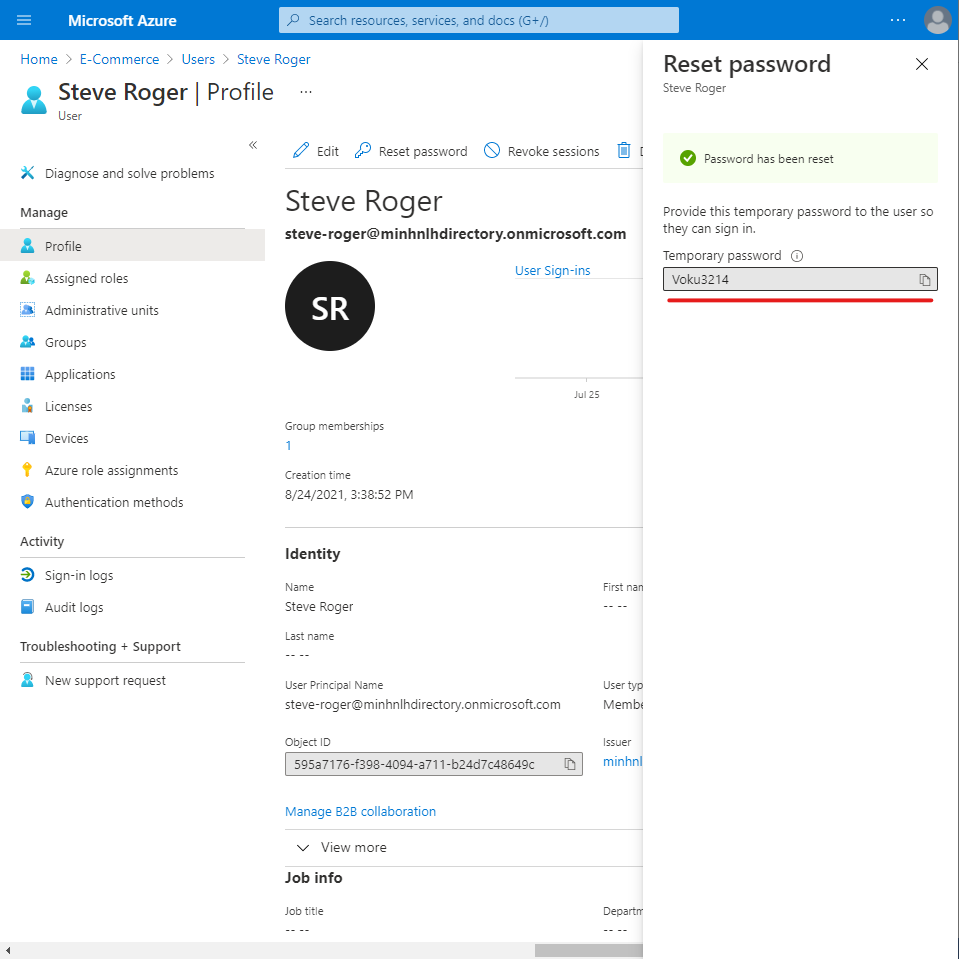


1. Add a Description and select duration in the Expires list. Select Add. The value for the key will be automatically filled in

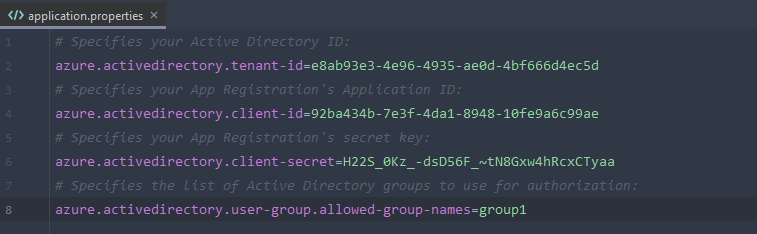
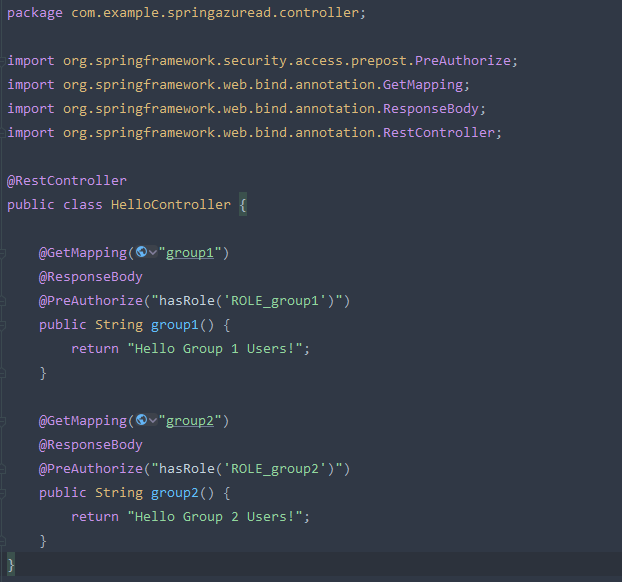


1. Copy and save the value of the client secret to configure your application.properties file later in this tutorial. (You won't be able to retrieve this value later.)
2. Select API permissions in the left navigation pane.
3. Select Microsoft Graph and tick Access the directory as the signed-in user and Sign in and read user profile. Select Grant Permissions... and Yes when prompted.
4. Select Grant admin consent for Azure Sample and select Yes. 
5. From the main page for your app registration, select Authentication, and select Add a platform. Then select Web applications.
6. Enter http://localhost:8080/login/oauth2/code/ as a new Redirect URI, and then select Configure.

## Add a user account to your directory, and add that account to a group

1. From the Overview page of your Active Directory, select Users, and then select New user.
2. When the User panel is displayed, enter the User name and Name. Then select Create.
3. From the Overview page of your Active Directory, select Groups, then New group, which you'll use for authorization in your application.
4. Select No members selected. (For the purposes of this tutorial, we'll create a group named group1.) Search for the user created in the previous step. Choose Select to add the user to the group. Then select Create to create the new group.
5. Go back to the Users panel, select your test user, and select Reset password, and copy the password. You'll use the password when you log into your application later in this tutorial.

# Configure and compile your app

1. Extract the files from the project archive you created and downloaded earlier in this tutorial into a directory.
2. Navigate to the src/main/resources folder in your project, then open the application.properties file in a text editor.
3. Specify the settings for your app registration using the values you created earlier. For example:
4. Save and close the application.properties file.
5. Create a folder named controller in the Java source folder for your application. For example: src/main/java/com/example/springazuread/controller.
6. Create a new Java file named HelloController.java in the controller folder and open it in a text editor.
7. Enter the following code, then save and close the file:
8. Open your application class in a text editor.
9. Add @EnableWebSecurity and @EnableGlobalMethodSecurity(prePostEnabled = true) in your application class as shown in the following example, then save and close the file:

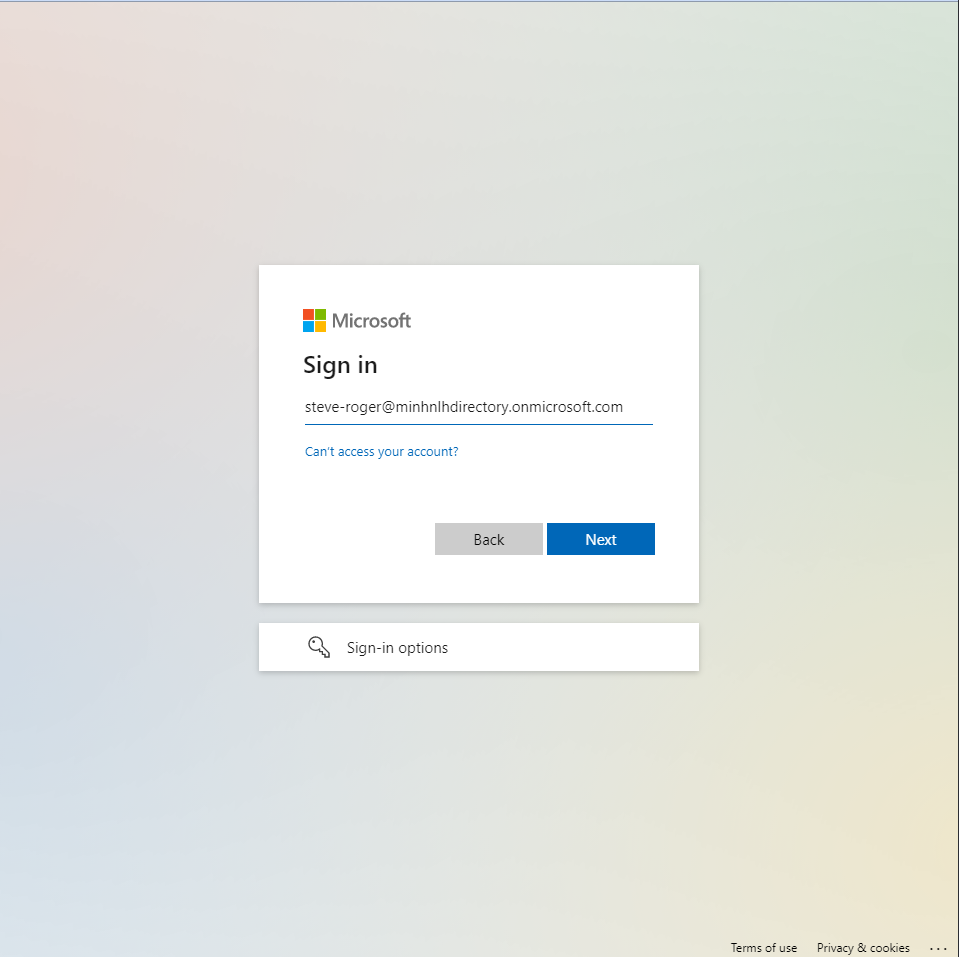
# Build and test your app

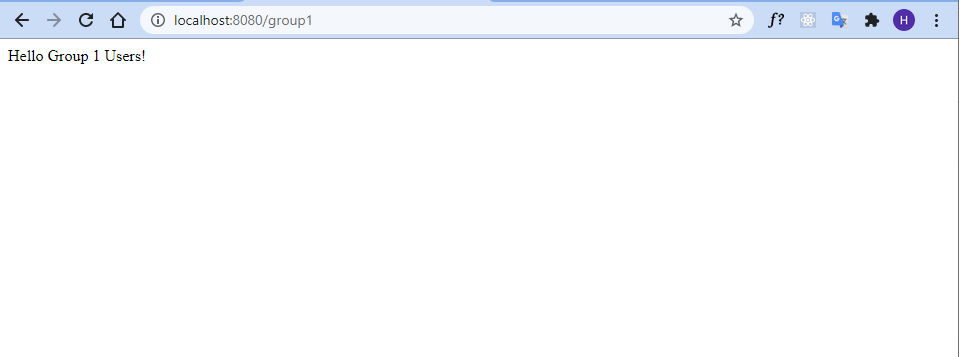
1. Open a command prompt and change directory to the folder where your app's pom.xml file is located.
2. Build your Spring Boot application with Maven and run it. For example:

**mvn clean package**

**mvn spring-boot:run**

1. After your application is built and started by Maven, open http://localhost:8080/group1 in a web browser. You should be prompted for a user name and password.



1. After you've logged in successfully, you should see the sample "Hello Group 1 Users!" text from the controller.