**OAuth 2.0 Spring Security with Okta**

**OAuth 2:**

OAuth 2.0, which stands for “Open Authorization”, is a standard designed to allow a website or application to access resources hosted by other web apps on behalf of a user.

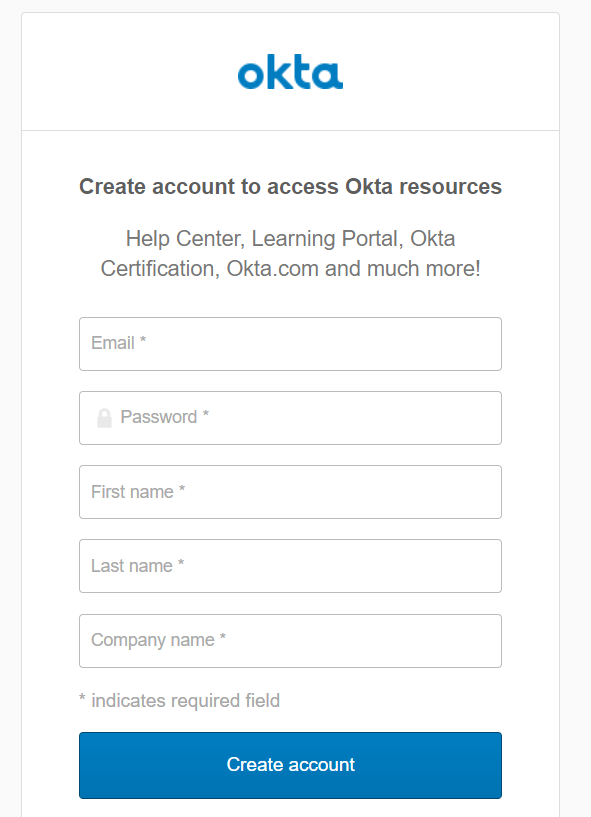
**Principles of OAuth2**:

OAuth 2.0 is an authorization protocol and NOT an authentication protocol. As such, it is designed primarily as a means of granting access to a set of resources, for example, remote APIs or user’s data.

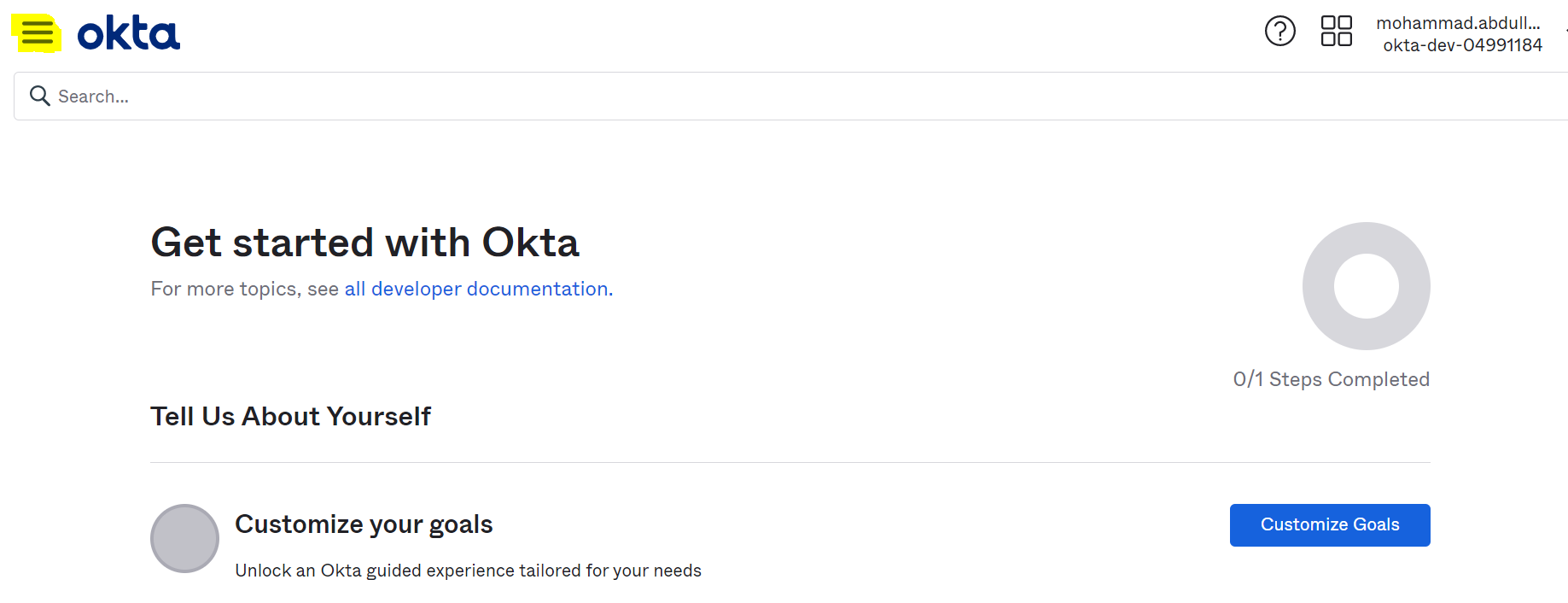
OAuth 2.0 uses Access Tokens. An Access Token is a piece of data that represents the authorization to access resources on behalf of the end-user. OAuth 2.0 doesn’t define a specific format for Access Tokens. This enables token issuers to include data in the token itself. Also, for security reasons, Access Tokens may have an expiration date.

**Get token manually:**

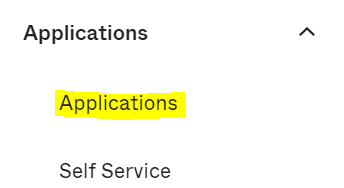
1. Create an account with email ID.



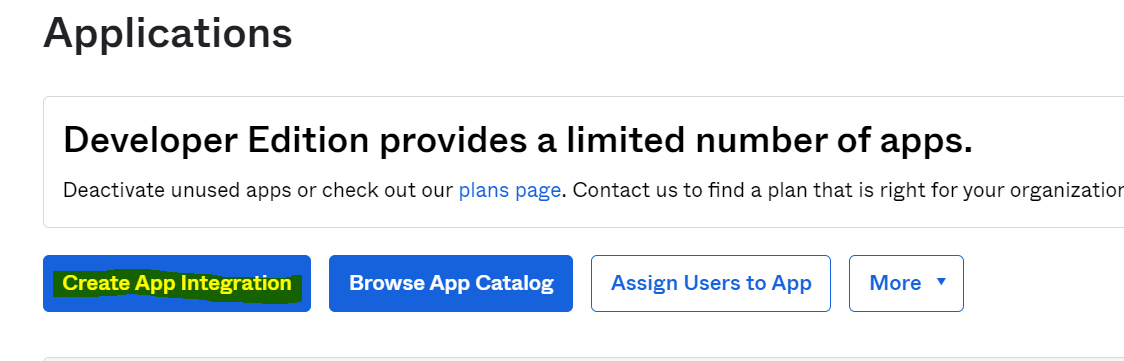
1. After login the below screen appears, and click on highlighted point:



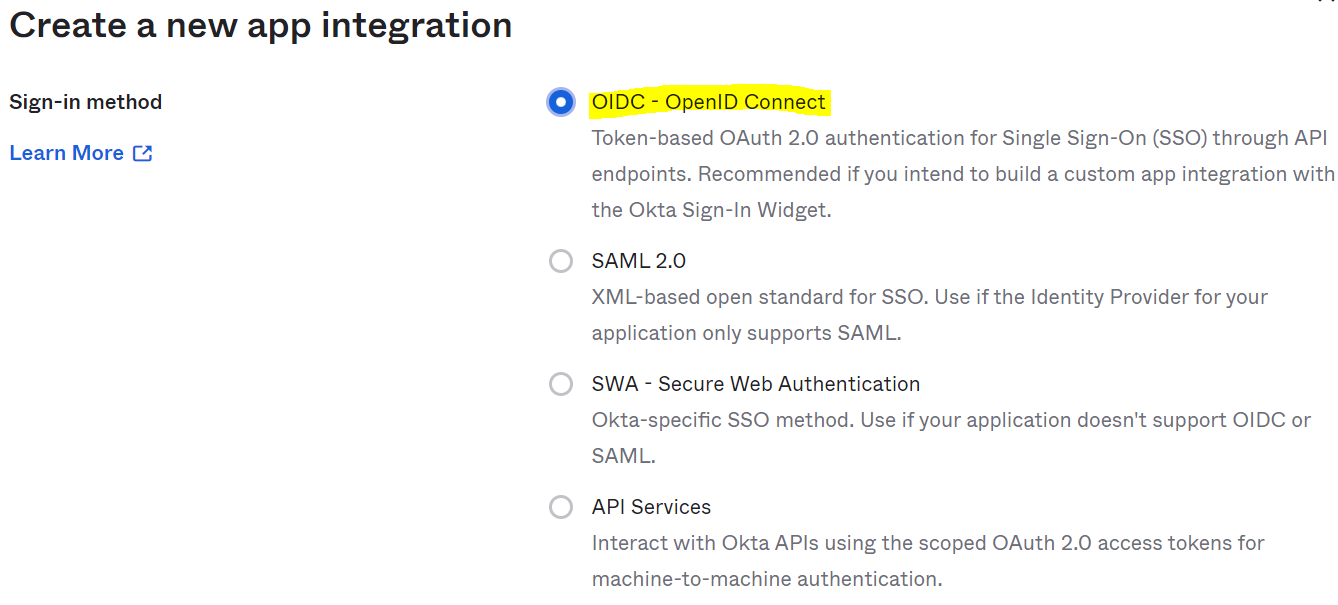
1. Go to Application -> Applications:



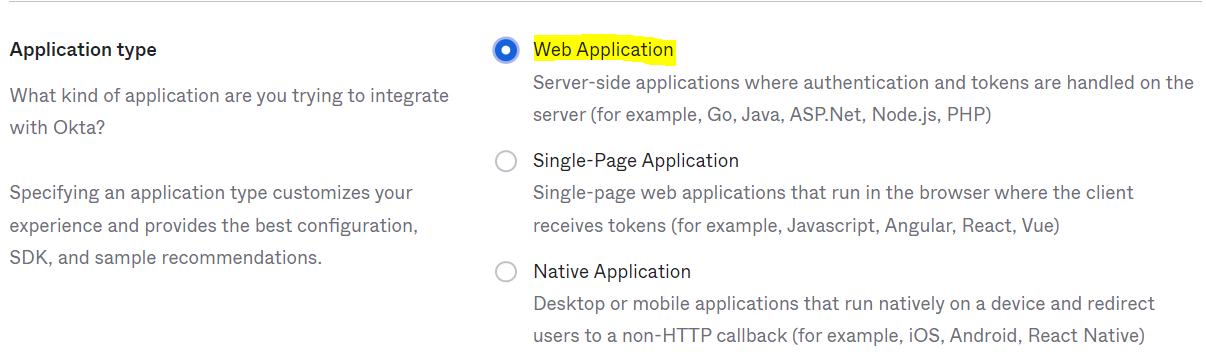
1. Create App Integration:



Select OIDC:

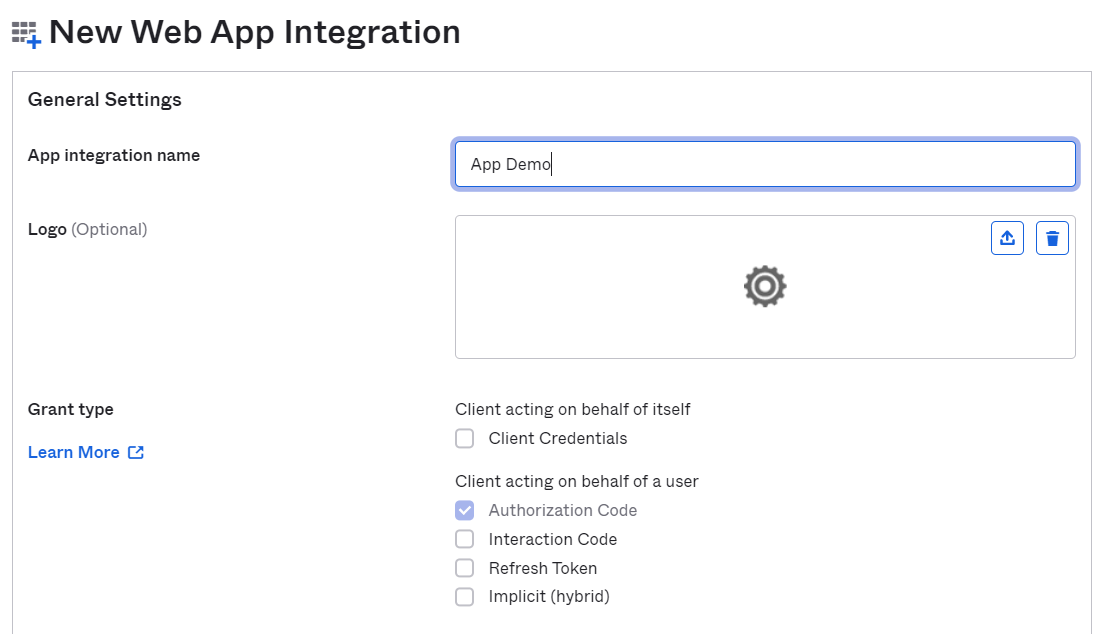


And select:



Click Next.

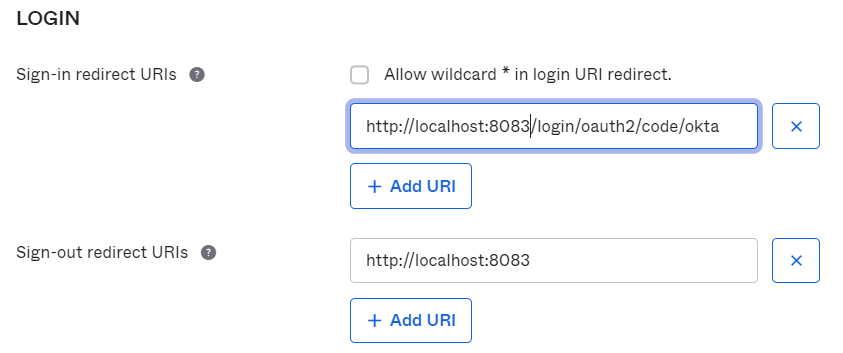
1. Provide the App name:



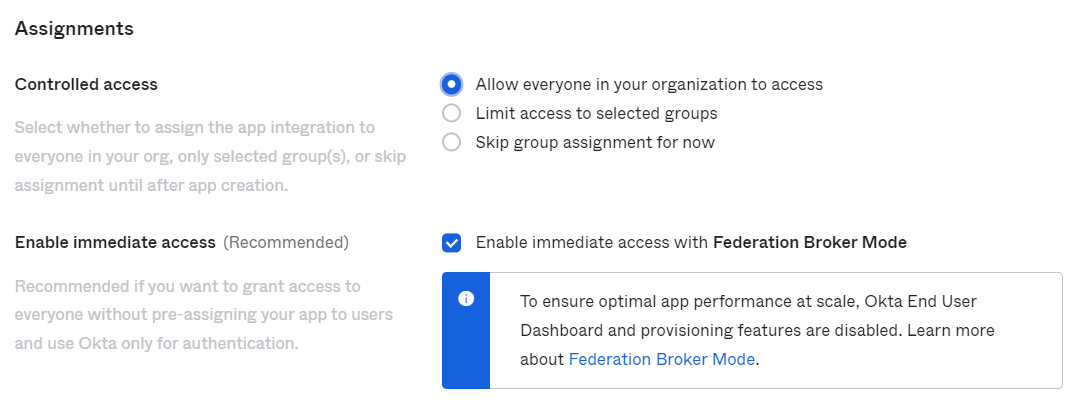
Add sign in and sign out redirect URI:

Sign In URI: - <http://localhost:8081/login/oauth2/code/okta>

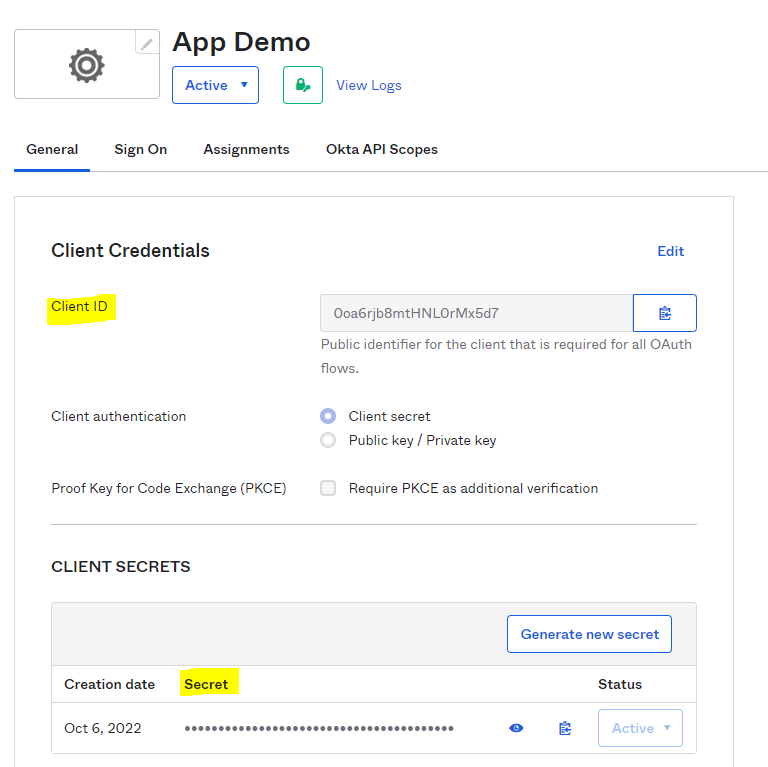
Sign out URI: - http://localhost:8083



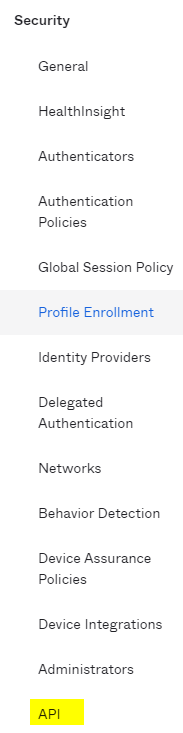
Assign the application for access based on group:



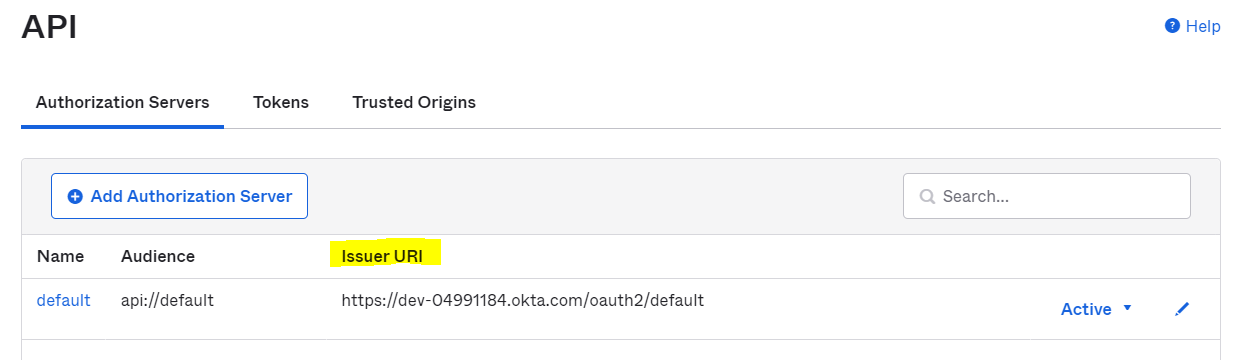
Click on save then it will generate the client Id and client Secret. See below:



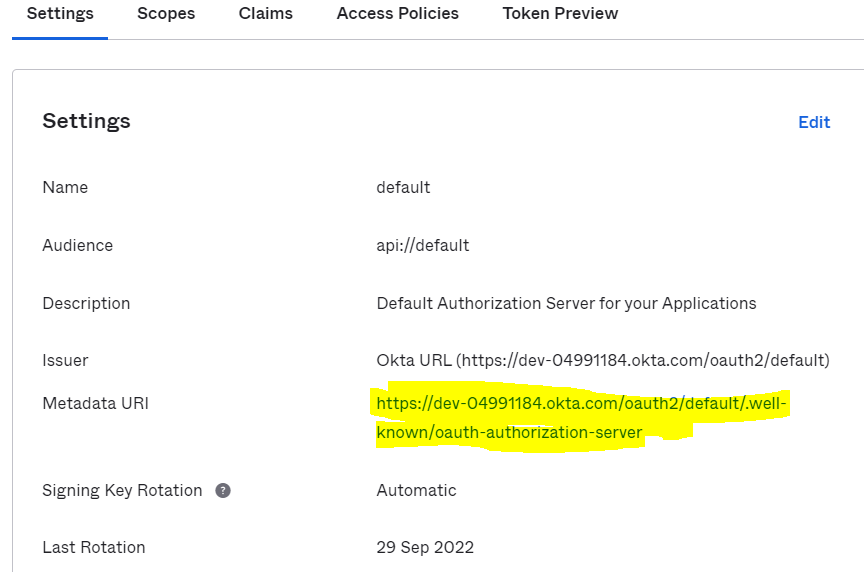
1. Go to Security and Click on API:



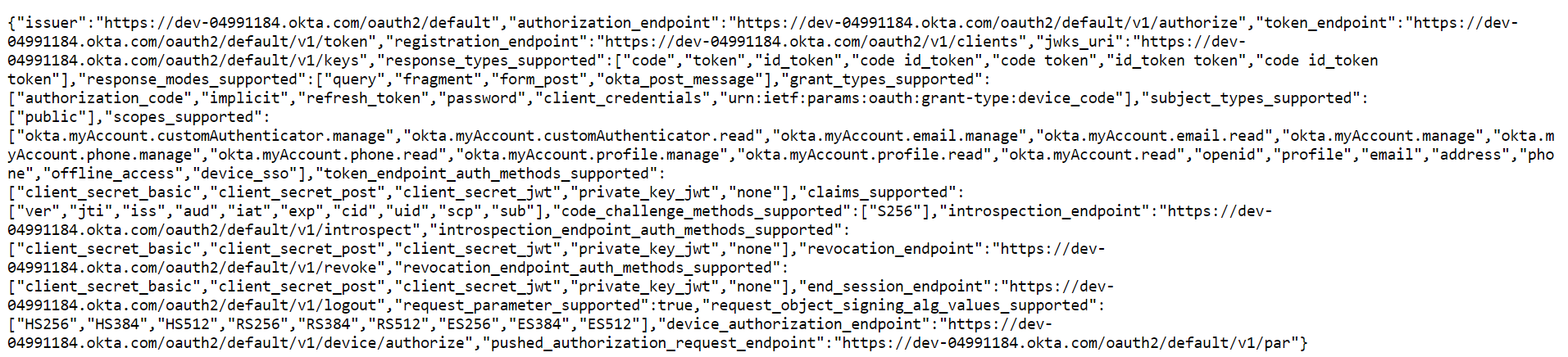
You can verify the issuer URI:



Click on Default and verify the authorization URL etc.:



Click on metadata URI then it will be redirected to the related end points in the form of JSON:



After formatting it will display like below:



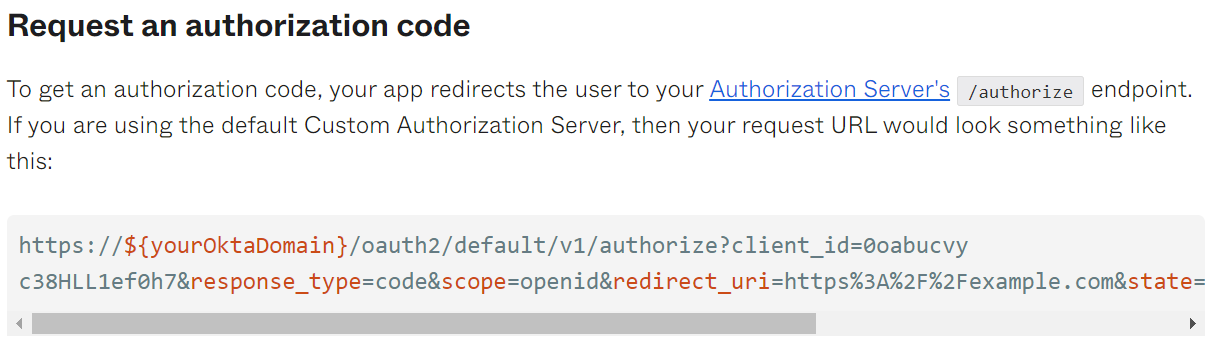
Collect all the end point for authentication reference.

1. Use Authorize URL to get the authentication code:

Authorize url: - <https://dev-04991184.okta.com/oauth2/default/v1/authorize>

Check developer document (“https://developer.okta.com/docs/guides/”).

Go to Authorization -> Implement authorization by grant type and see the below section:



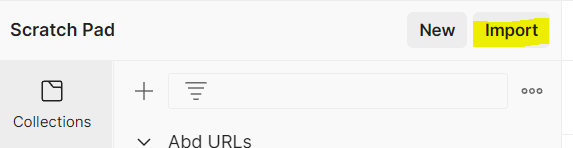
To get the code modify the URL like below:

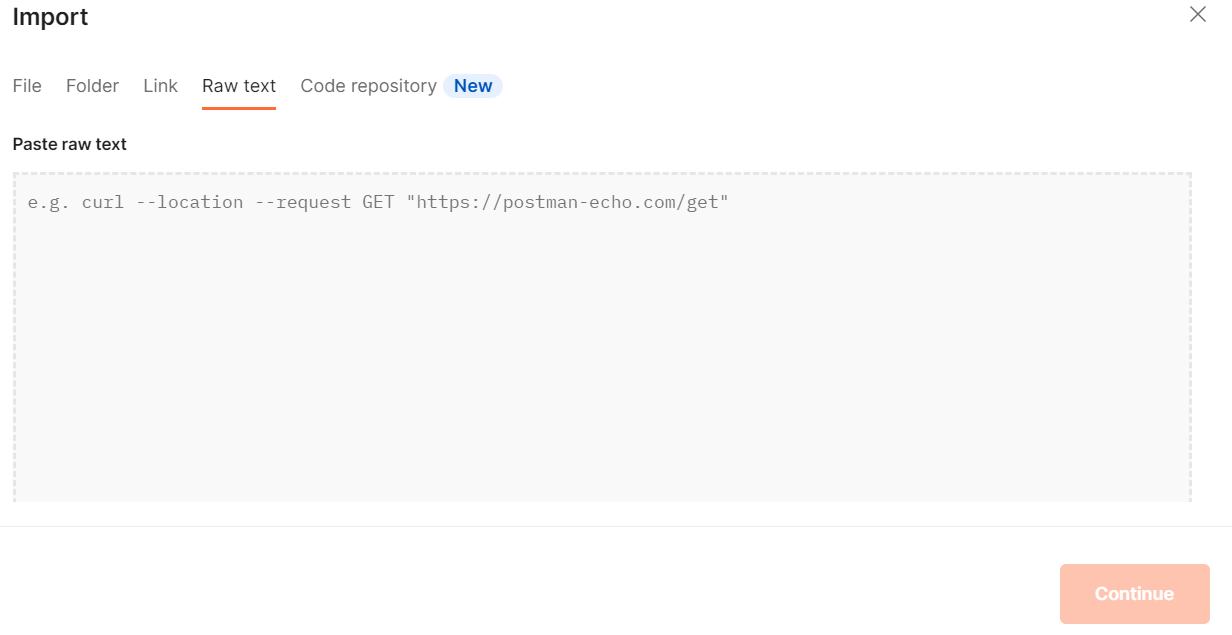
https://dev-04991184.okta.com/oauth2/default/v1/authorize?client\_id=${ClientID}&response\_type=code&scope=openid&redirect\_uri=${redirectURI}&state=state-296bc9a0-a2a2-4a57-be1a-d0e2fd9bb601

In response you will get the below URI and get the code from the URI:

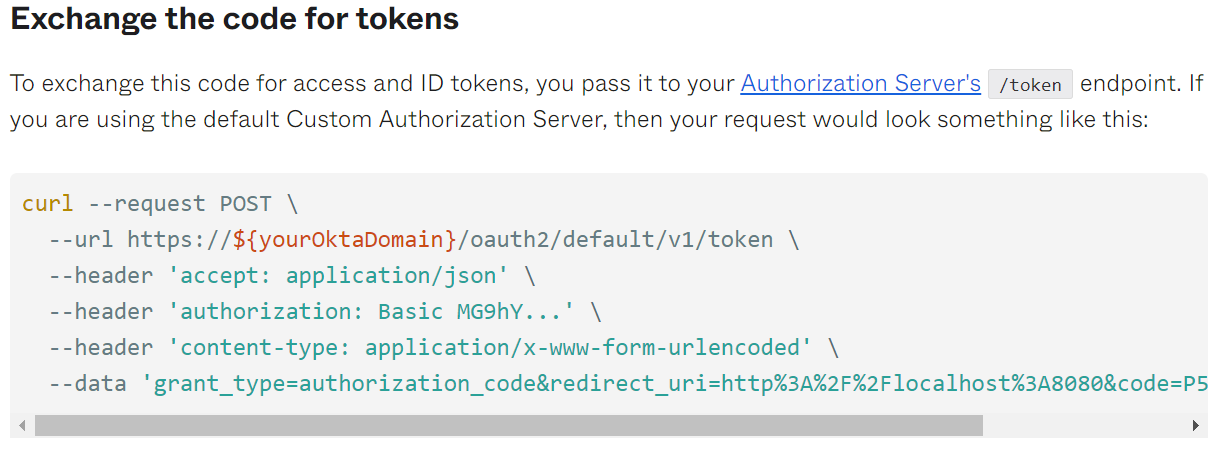
<http://localhost:8081/login/oauth2/code/okta?code=BpVDezRMWBlhKuKyrCS8_fq6xYWbbaB-vQvlGfoZogo&state=state-296bc9a0-a2a2-4a57-be1a-d0e2fd9bb601>

1. Open Postman and Click on import:





Again, go to developers document (“https://developer.okta.com/docs/guides/”) and see below section:



Use the above authorize code and modify the above code like below:

curl --request POST \

--url https://dev-04991184.okta.com/oauth2/default/v1/token \

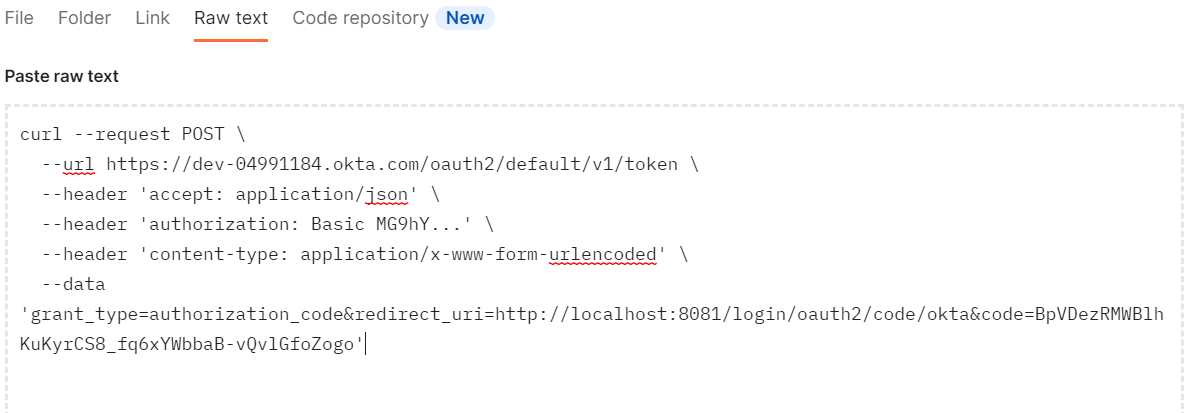
--header 'accept: application/json' \

--header 'authorization: Basic MG9hY...' \

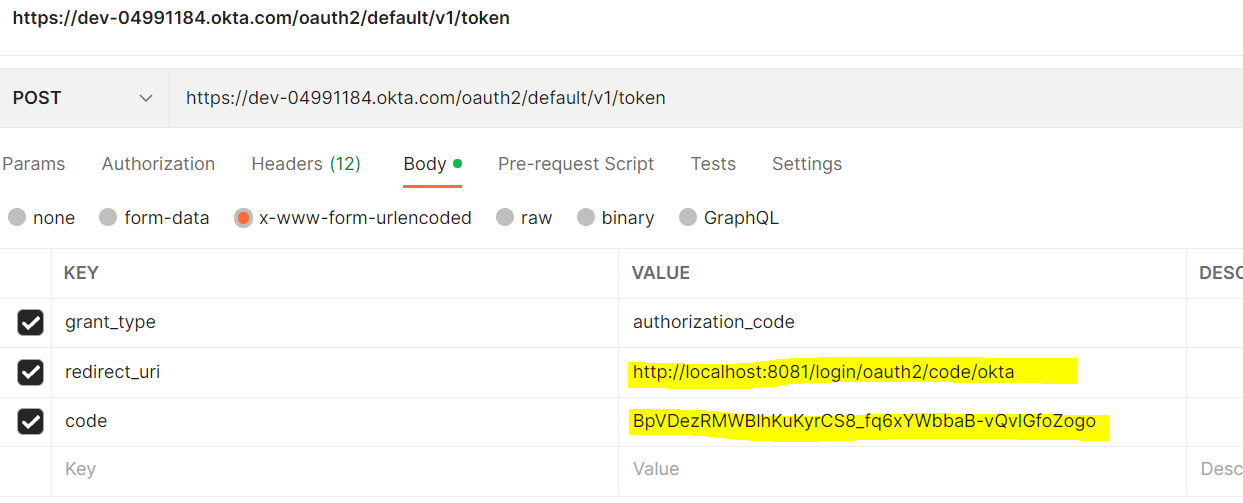
--header 'content-type: application/x-www-form-urlencoded' \

--data 'grant\_type=authorization\_code&redirect\_uri=http://localhost:8081/login/oauth2/code/okta&code=BpVDezRMWBlhKuKyrCS8\_fq6xYWbbaB-vQvlGfoZogo'

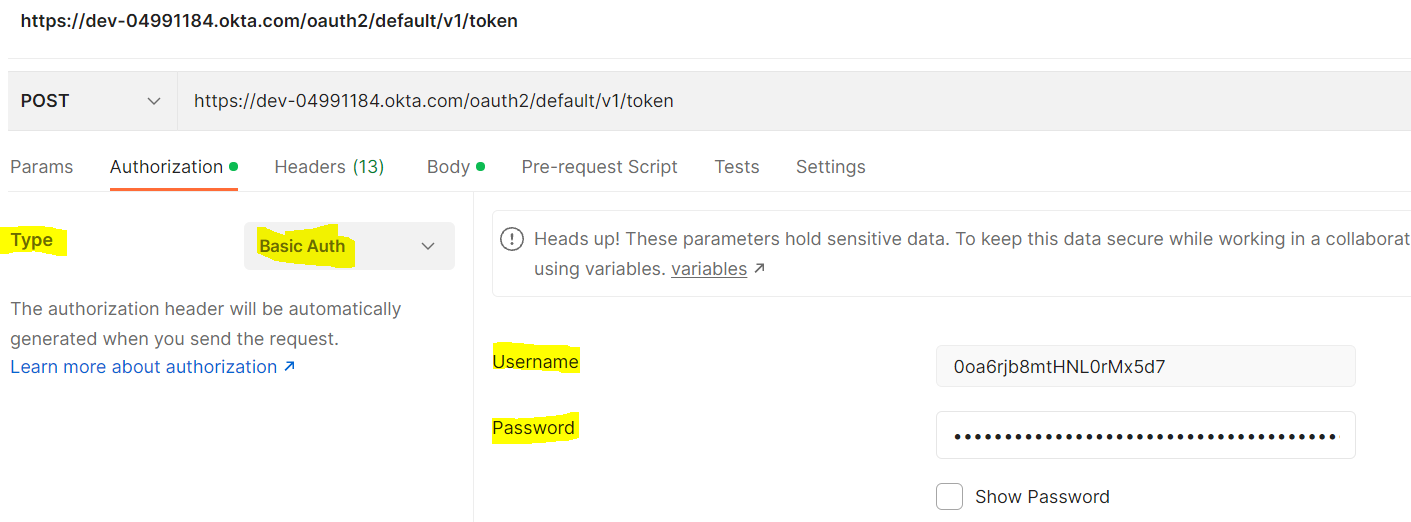
Paste this on postman:



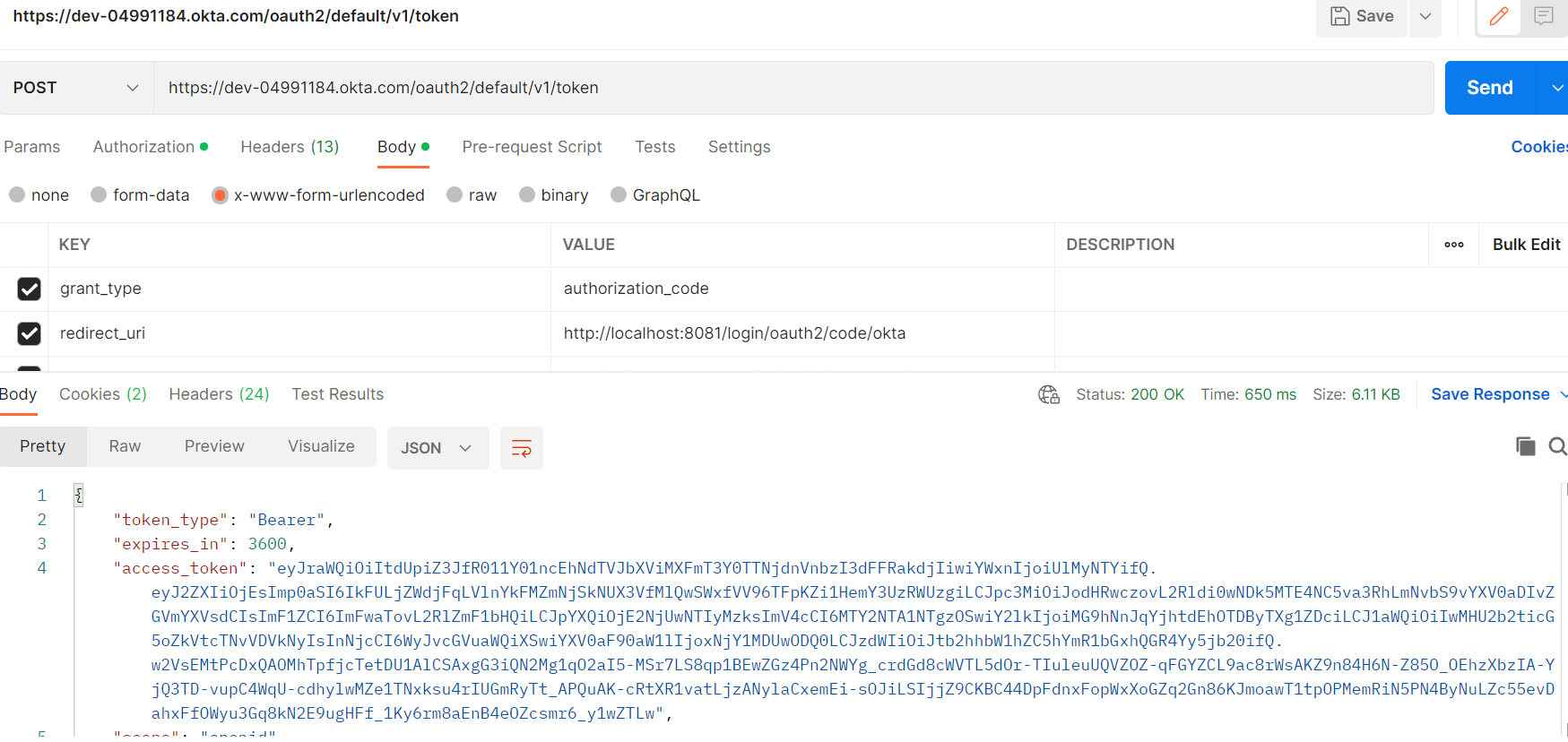
Click continue and click import and then verify the code and redirect URI:



Now, go to Authorization tab and provide the Client Id (Username) and Client secret (Password) from Okta:

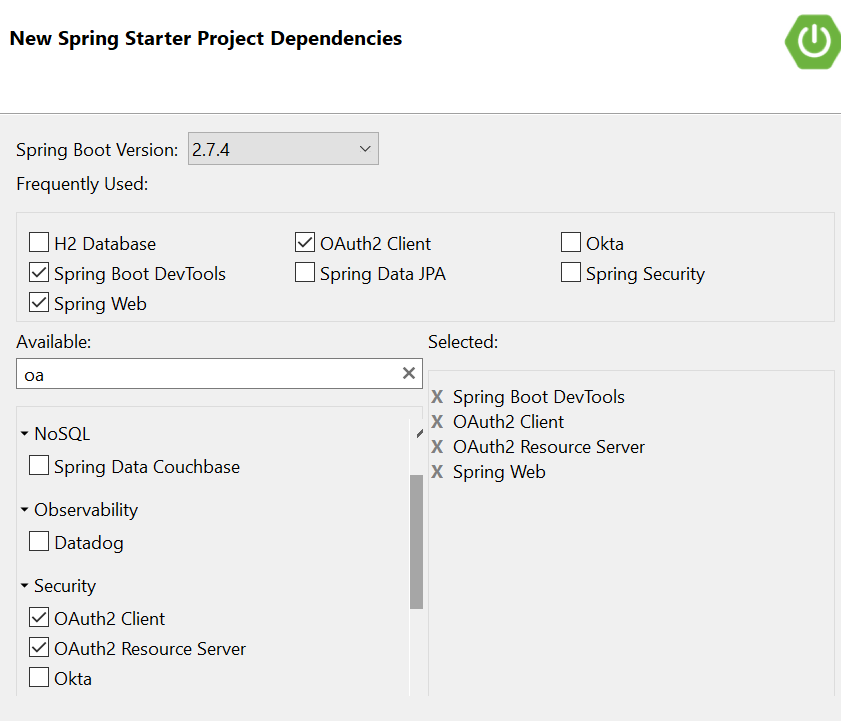


Click on send then you will get access token in response:

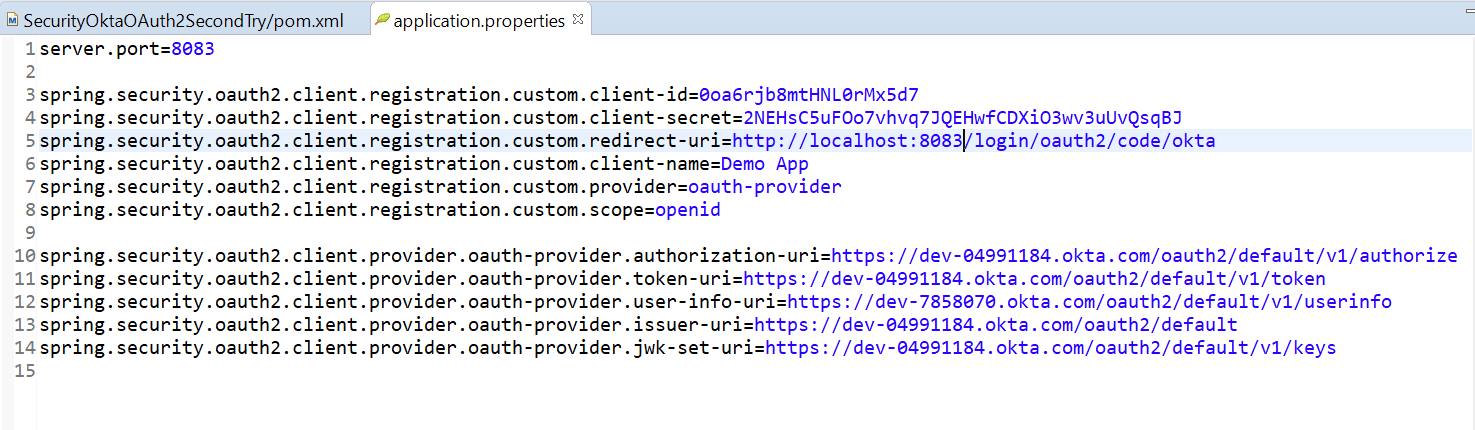


**Get token from spring boot application**

1. Create application with below dependency:



1. Add the properties like below in properties file:



1. Create the configuration file and use below code:

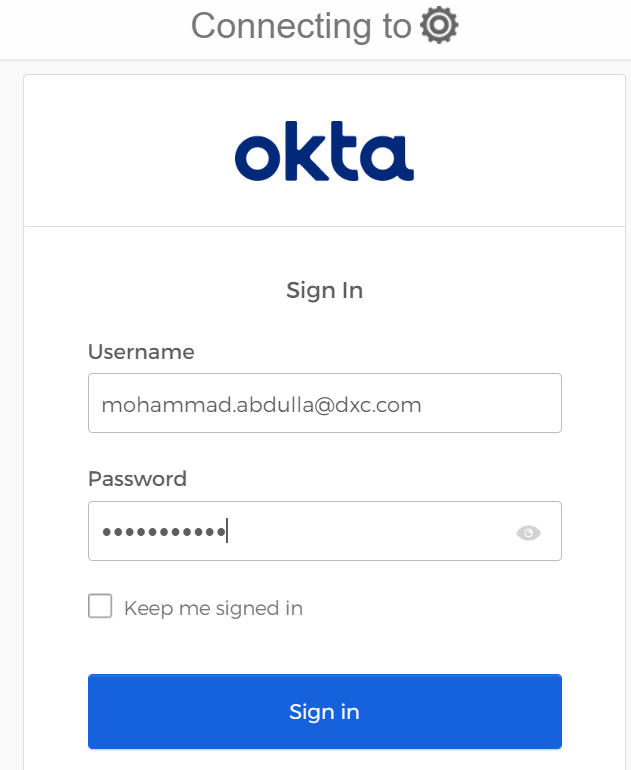


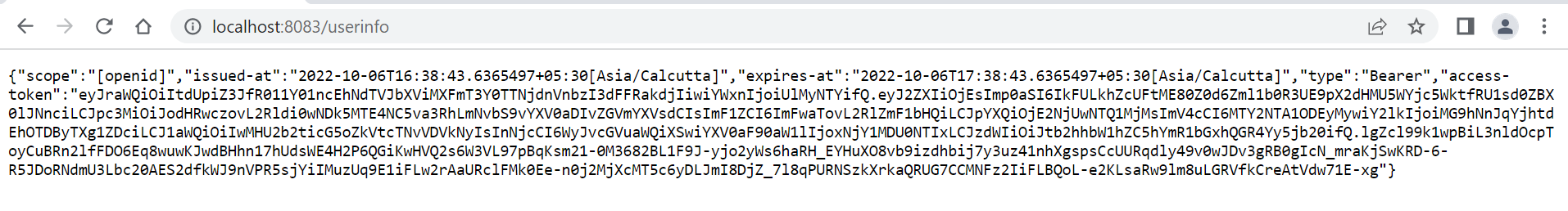
1. Create controller class:



1. Now run the application and check the user info end point on browser:

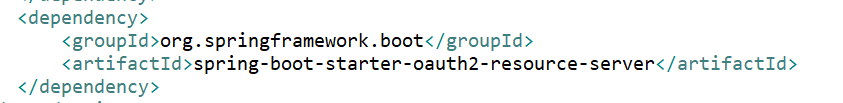
Hit the URL <http://localhost:8083/userinfo> and get the access token details after providing the credentials:



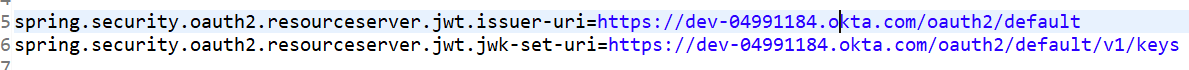


**Use token to authorize the API**

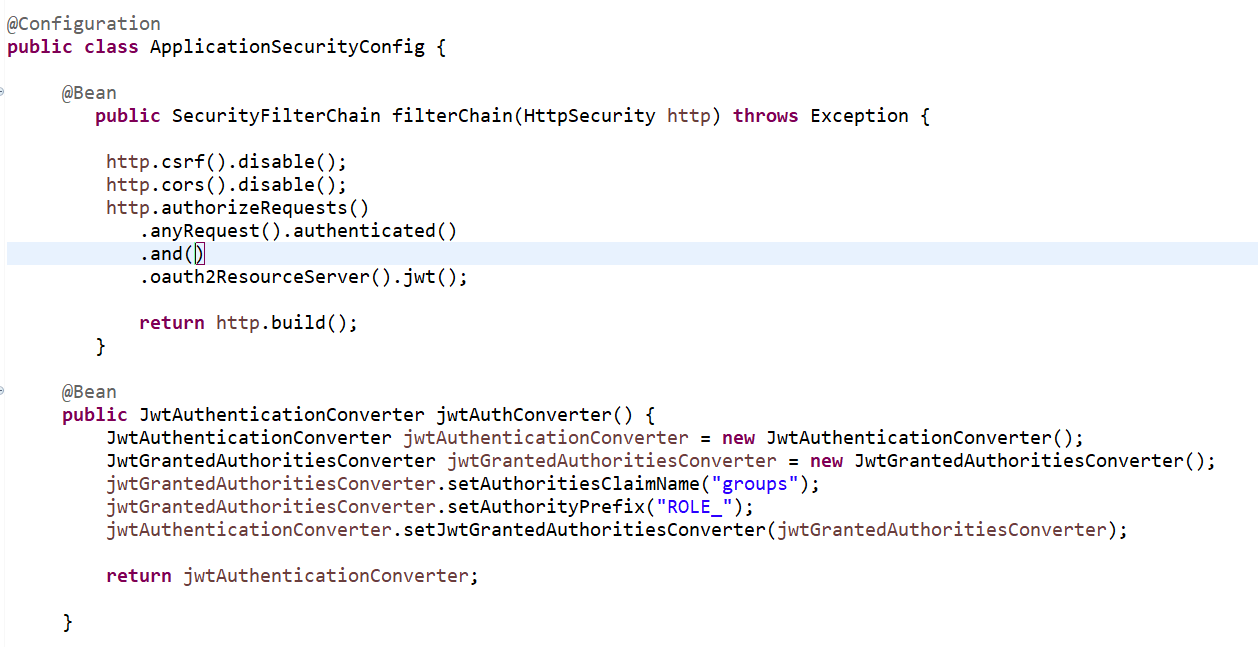
1. Create API using spring boot application
2. Add dependency of OAuth 2 resource server:



1. Add issuer URL and jwks\_uri in properties file:

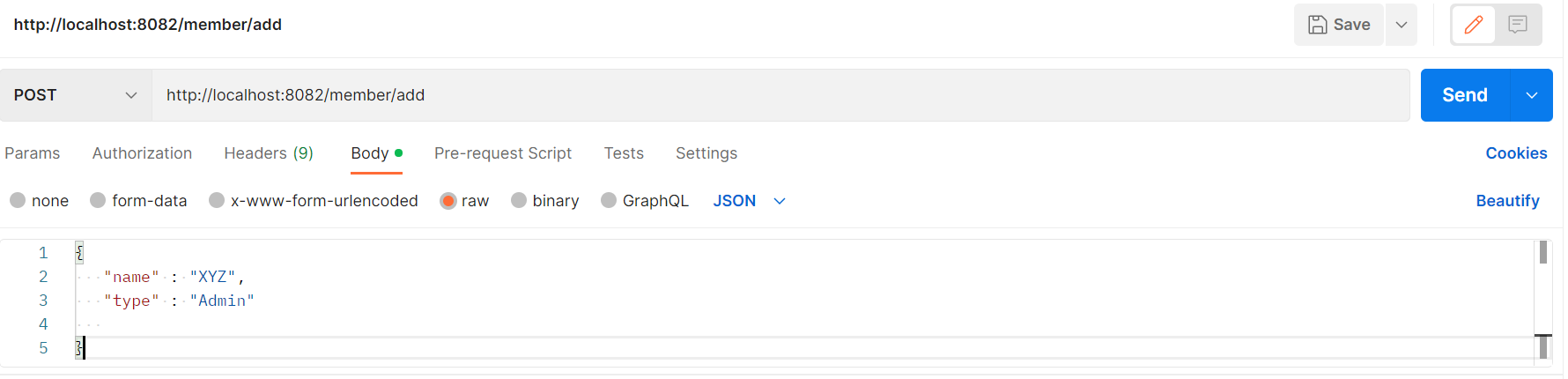


1. Create configuration file to allow URL for authorization:



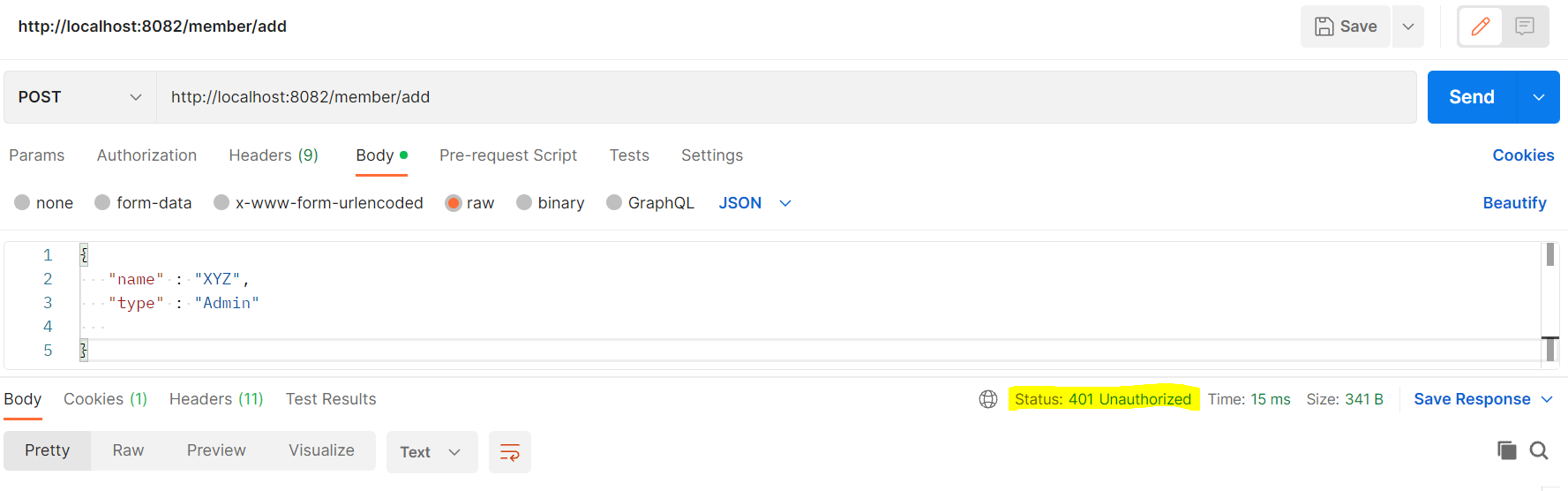
1. Open postman and add bearer token in authorization tab:

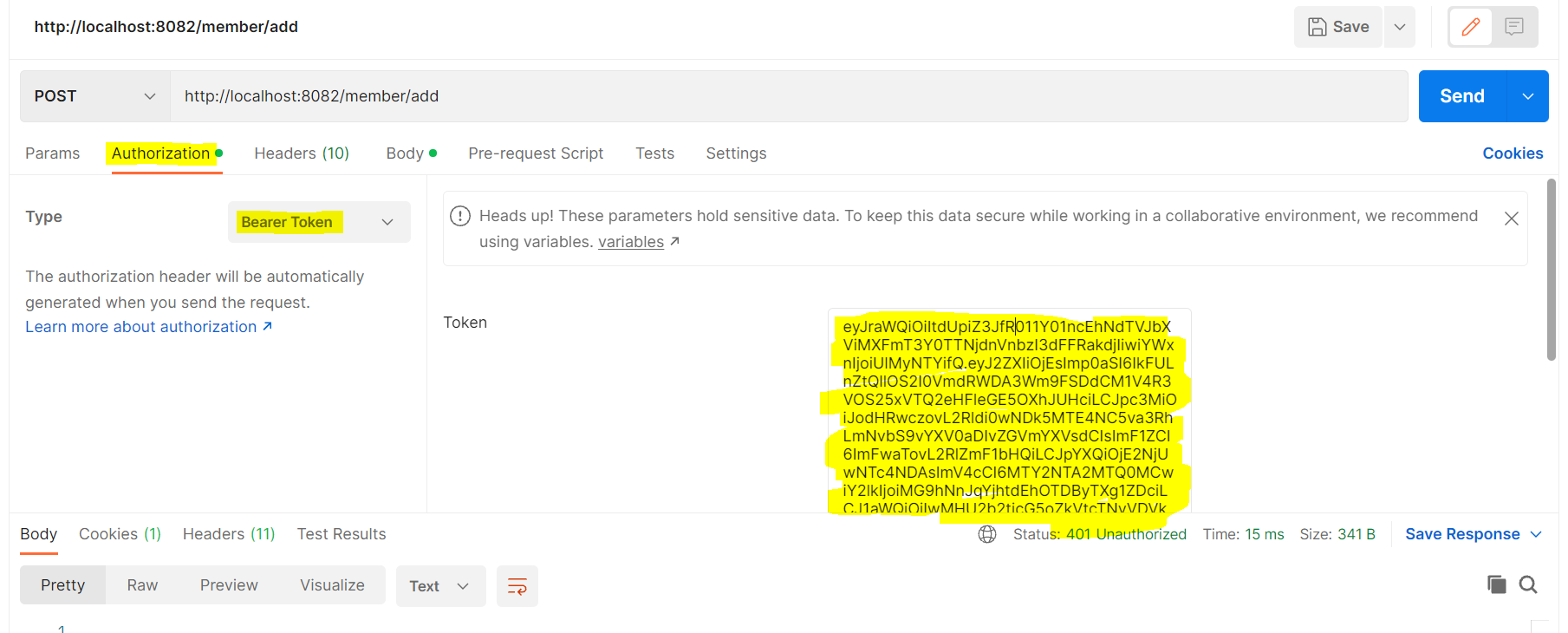
Without token:



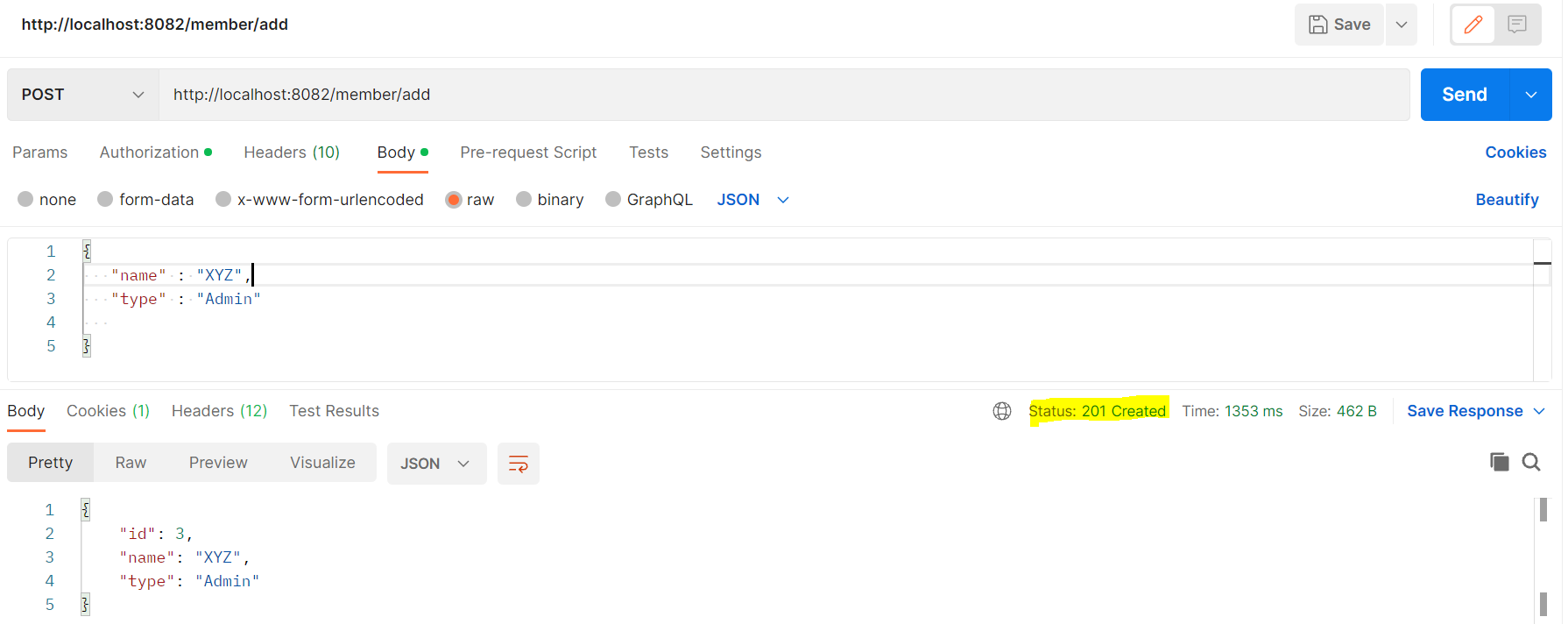
Output:

See below the API has not been authorized:



With Token (Paste the access token from userinfo end point):

Click Send:



Now it authorized and added the data into H2 DB.