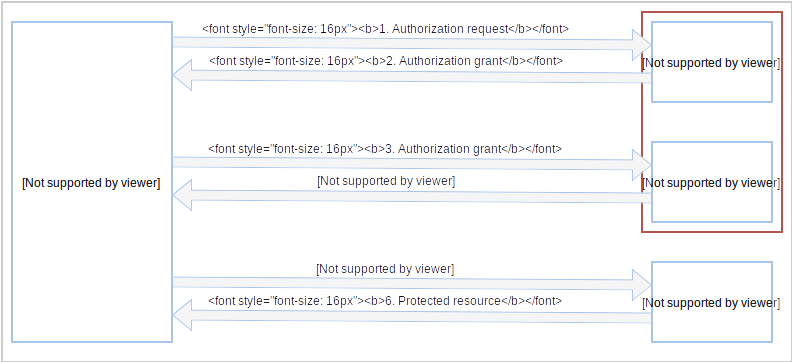
# **Spring boot security + Oauth2 authentication + in memory example:**

* The resource owner authorizes an application to access their account. You can scope this access.
* The authorization server verifies the identity of the user then issues access tokens to the application. From the developer’s point of view, services’s API covers both: resource owner and authorization server roles.
* The client is the application that wants to access the user’s account.

**Diagram :**



## **Environment Setup :**

1. JDK 8.

2. Spring Boot.

3. Intellij eclipse.

4. Maven.

5. Spring Security : 1. AuthorizationSreverConfig.

2. ResourceServerConfig

3. SecurityConfig

6.application.properties file.

**Run Project In Postman :**

**1. Login with oauth :**

**Api :** <http://localhost:9191/oauth/token>

**From-data** : grant\_type : password

username : (enter username from security config file )

password : (enter password from security config file )

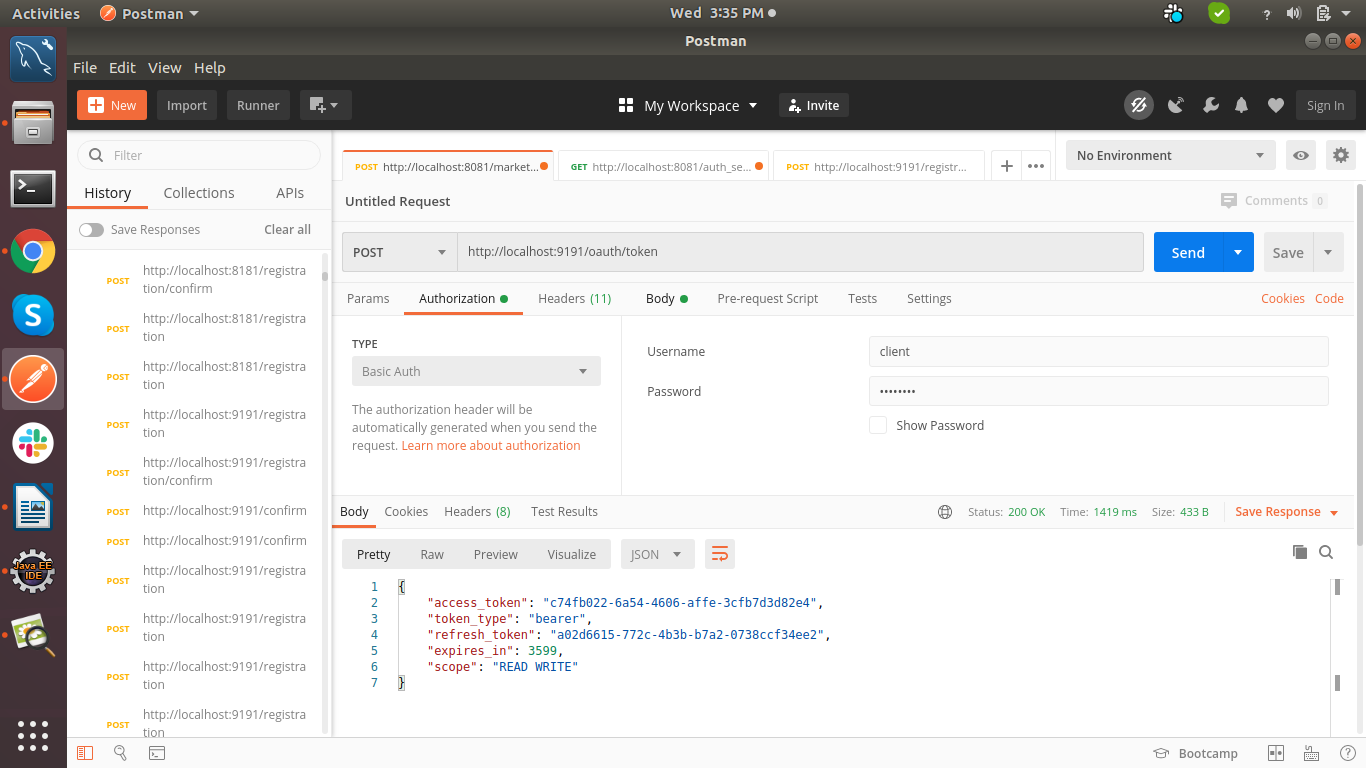
**Authorization** : Type : Basic Auth

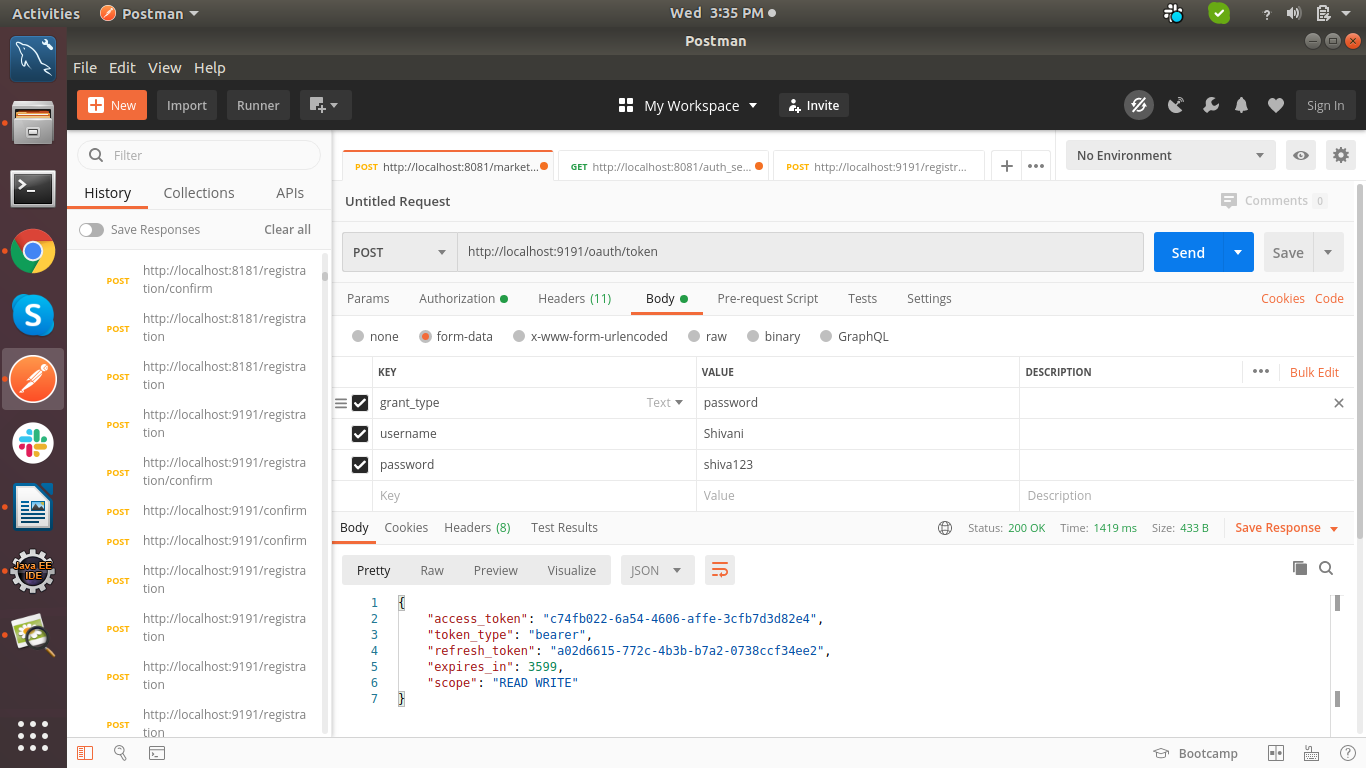
Username : (enter clientname from AuthorizationSreverConfig file)

Password : (enter secret from AuthorizationSreverConfig file)

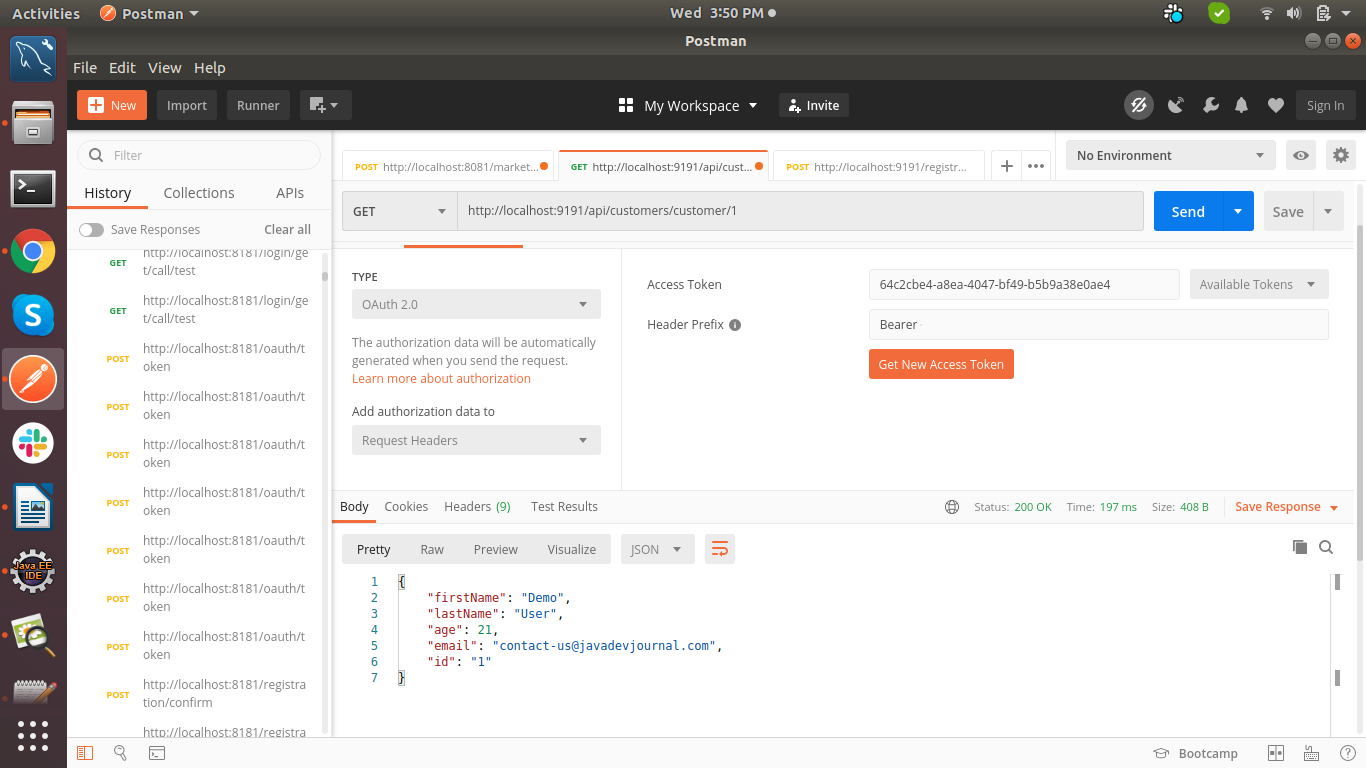
**Demo :**

**1. Client & Secret :**

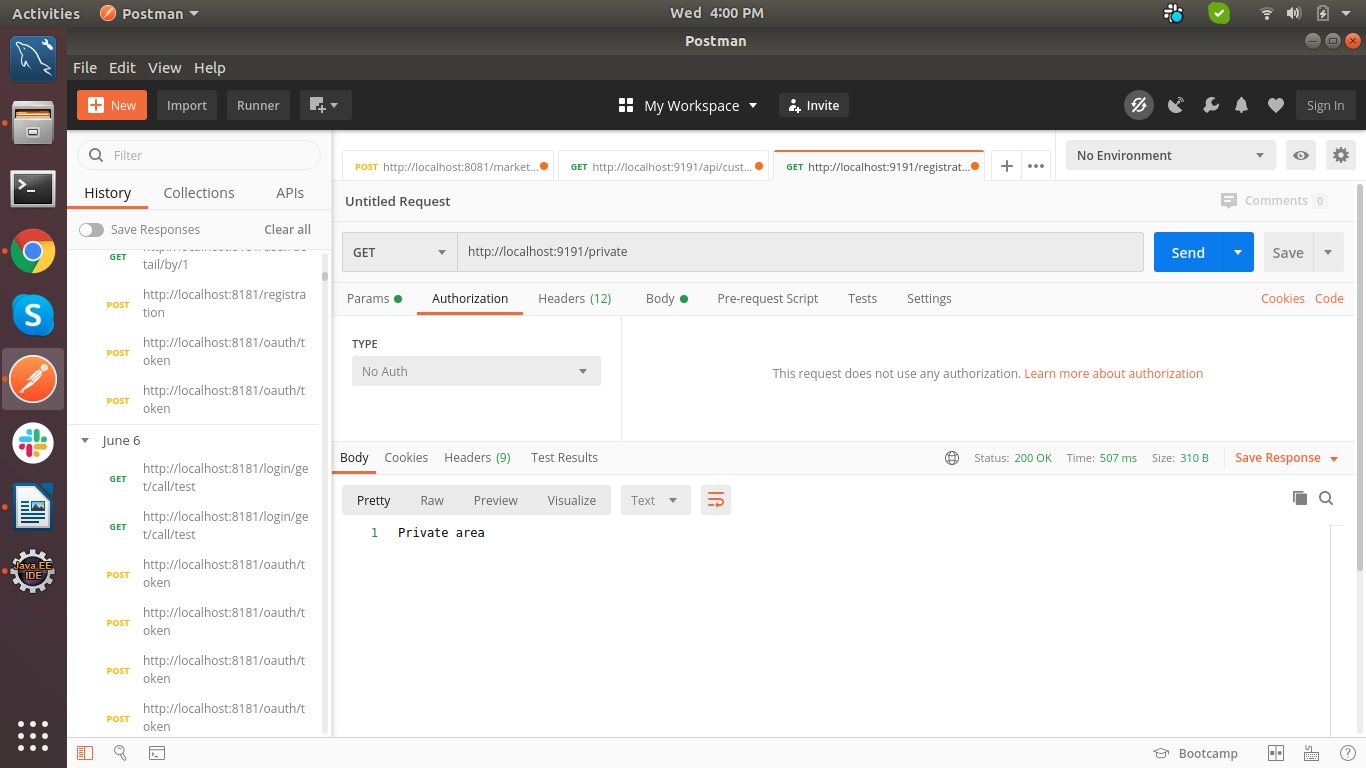




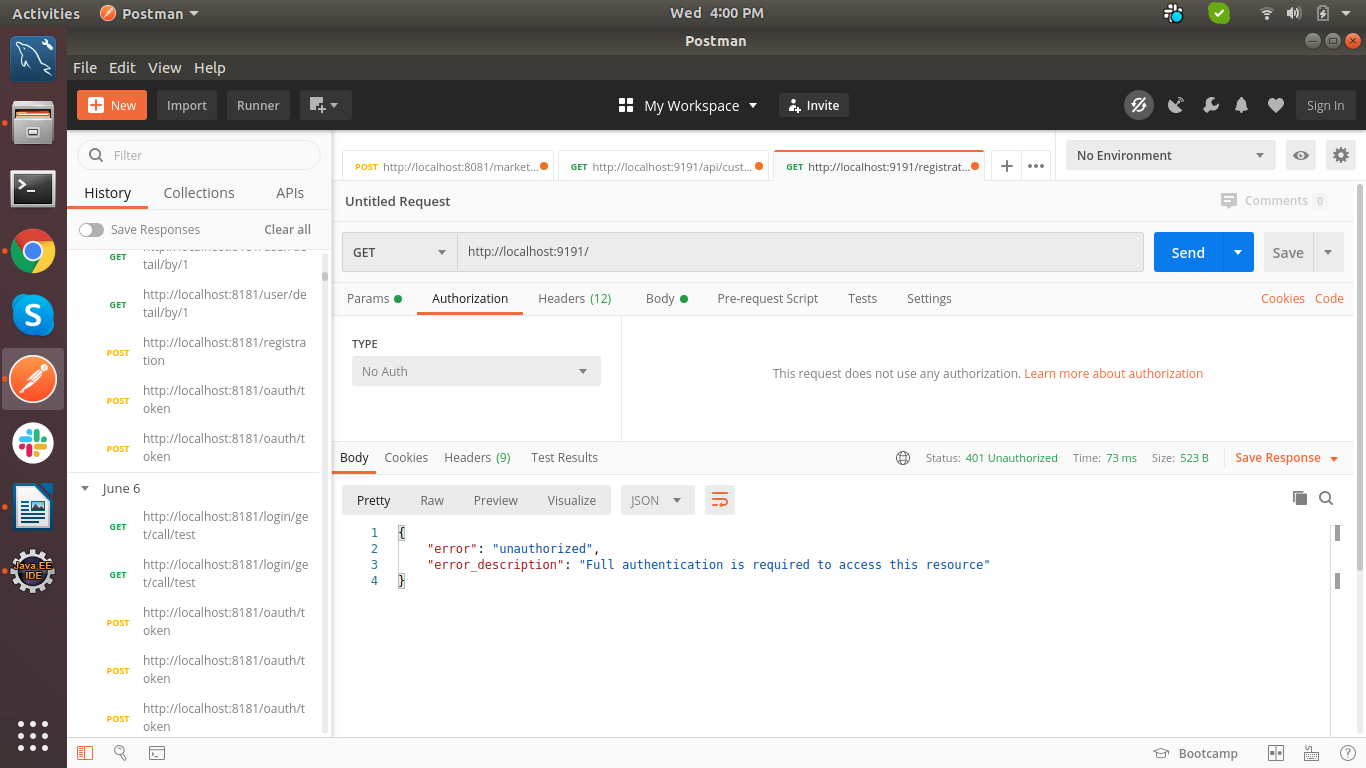
**2. Get API with access\_token :**



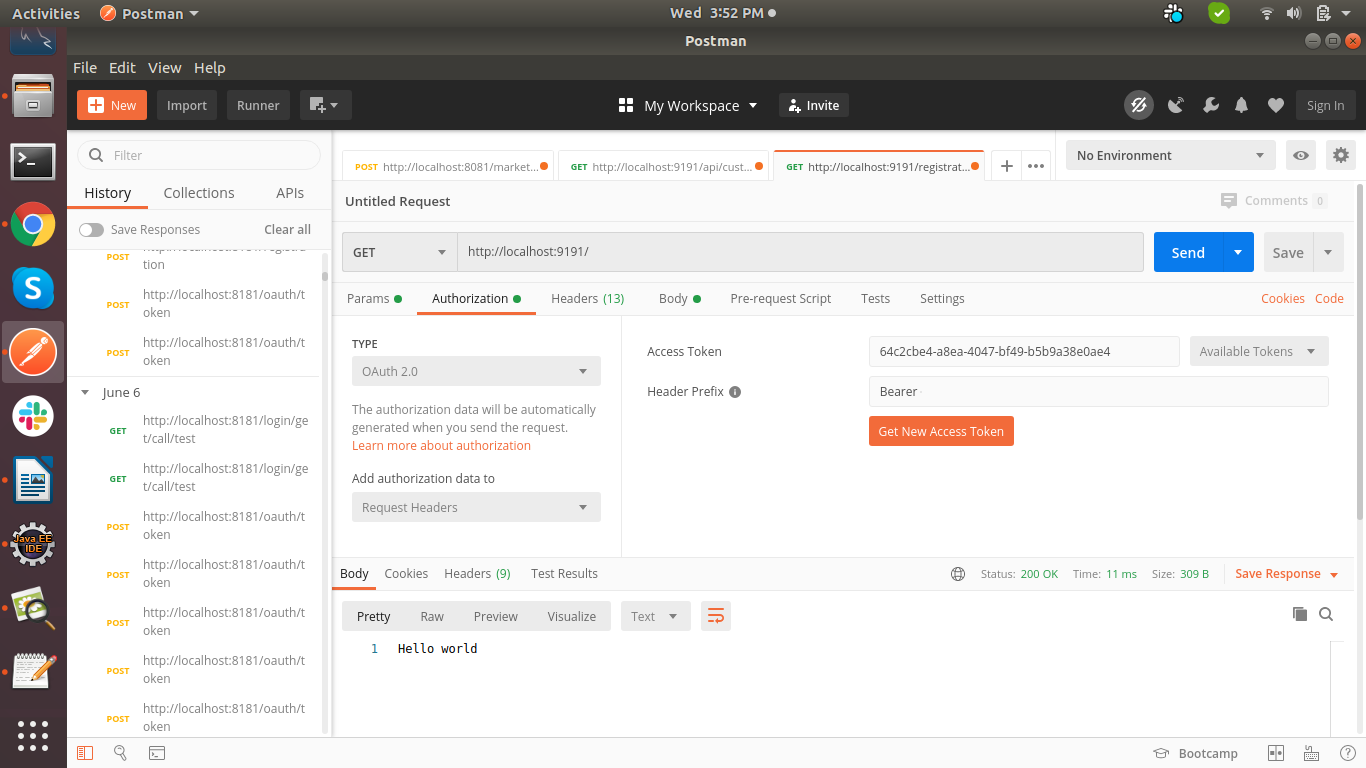
**3. This Api is allowed to execute without access\_token (check in ResourceServerConfig file):**

****

**4. This API is not allowed to be executed without authentication (access\_token) :**

****

**5. If you pass the access token, then the execution will be done properly :**

****

**Refer link:**

[**https://dzone.com/articles/securing-rest-services-with-oauth2-in-springboot-1**](https://dzone.com/articles/securing-rest-services-with-oauth2-in-springboot-1)