# **Spring boot security + Oauth2 authentication + JWT example:**

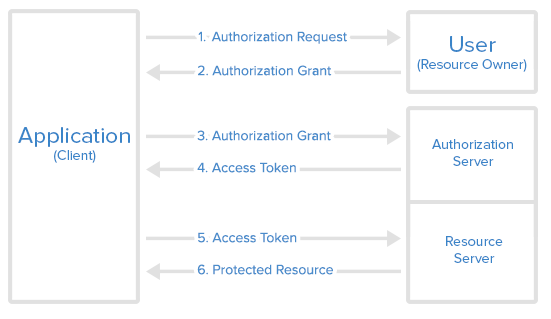
[OAuth 2](https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2) is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service. It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account. OAuth 2 provides authorization flows for web and desktop applications, and mobile devices.

#### **Roles :** OAuth defines four roles –

* **Resource Owner** – The user of the application.
* **Client** – the application (user is using) which require access to user data on the resource server.
* **Resource Server** – store user’s data and http services which can return user data to authenticated clients.
* **Authorization Server** – responsible for authenticating user’s identity and gives an authorization token. This token is accepted by resource server and validate your identity.

**Following are the 4 different grant types defined by Oauth2 :**

* **Authorization Code:** used with server-side Applications
* **Implicit:** used with Mobile Apps or Web Applications (applications that run on the user's device)
* **Resource Owner Password Credentials:** used with trusted Applications, such as those owned by the service itself
* **Client Credentials:** used with Applications API access

**Diagram :**

## **JWT Token :**

JWT Token is a JSON Web Token, used to represent the claims secured between two parties. You can learn more about the JWT token at <https://jwt.io/>.

Now, we are going to build an OAuth2 application that enables the use of Authorization Server, Resource Server with the help of a JWT Token.

JWTs include all the information within the Token. So the Resource Server needs to verify the Token’s signature to make sure the data has not been modified.

For Example :

**1.** **Get access token :**

## **2. Copy your access\_token and paste** <https://jwt.io/> **here, get all information:**

## 

## **Environment Setup :**

1. JDK 8.

2. Spring Boot.

3. Intellij eclipse.

4. Maven.

5. Spring Security : 1. AuthorizationSreverConfig.

2. ResourceServerConfig.

3. SecurityConfig.

6. JDBC DataSource for database data.

7. **A**pplication.properties file.

**Run Project In Postman :**

**1. Registration :**

Api: <http://localhost:8181/registration>

Request :{ "emailId":"xyz[@gmail.com](mailto:anisha.lodhi@gmail.com)",

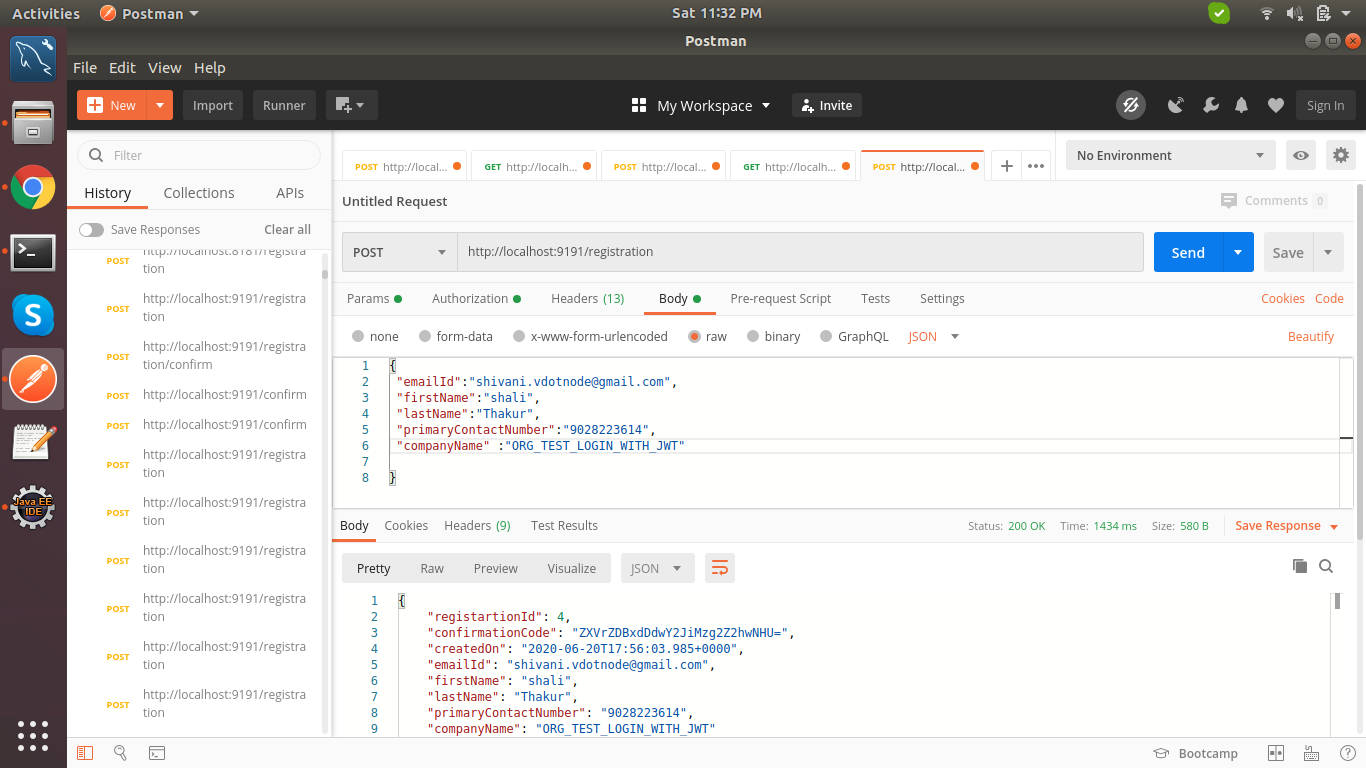
"firstName":"XYZ",

"lastName":"xyz",

"primaryContactNumber":"934563620",

"companyName" :"ORG\_TEST\_LOGIN\_ADMIN\_2" }

**Demo :**



**2. Confirm Password & Create Orgnization ,User Profile :**

**Request :**

Api : <http://localhost:8181/registration/confirm>

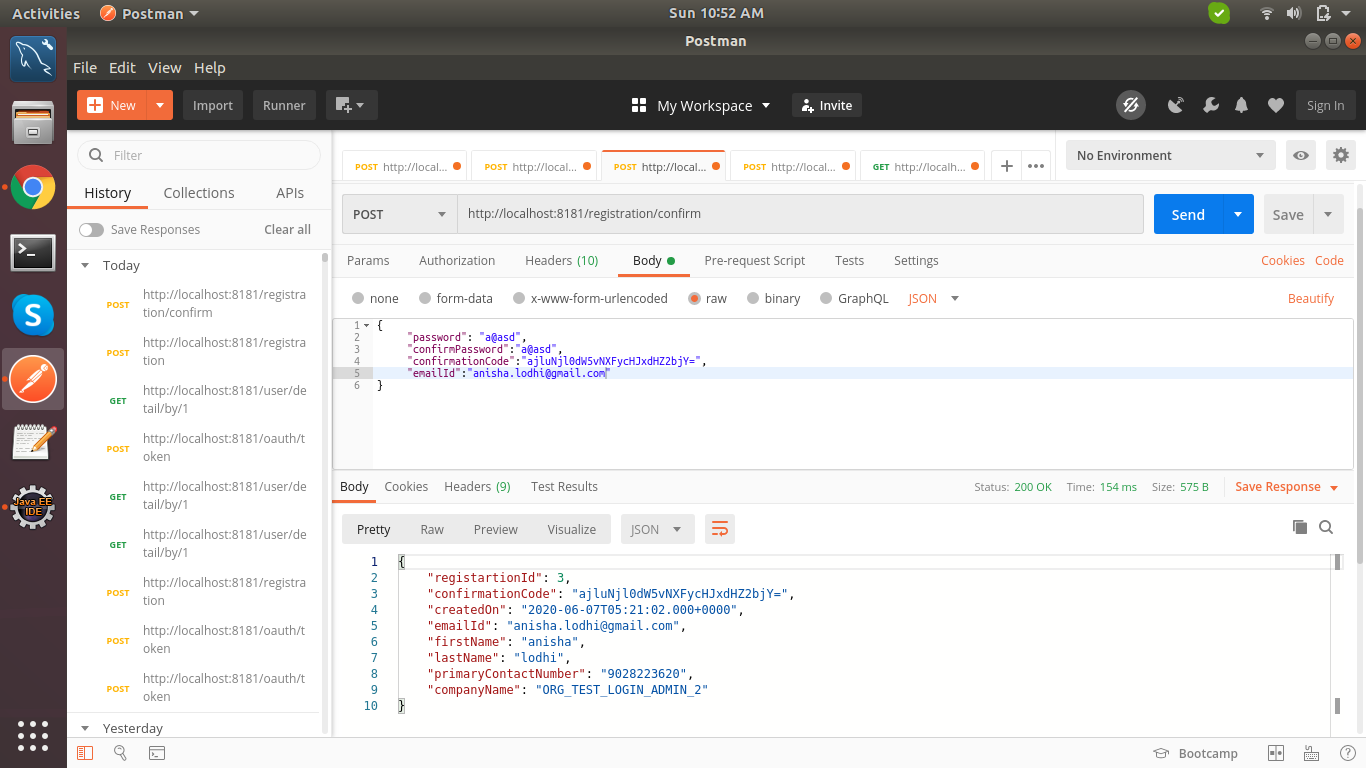
Request:{ "password": "a@asd",

"confirmPassword":"a@asd",

"confirmationCode":"ajluNjl0dW5vNXFycHJxdHZ2bjY=",

"emailId":"[anisha.lodhi@gmail.com](mailto:anisha.lodhi@gmail.com)" }

**Demo :**

****

**3. Login with oauth :**

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

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

username : (enter registor emailId )

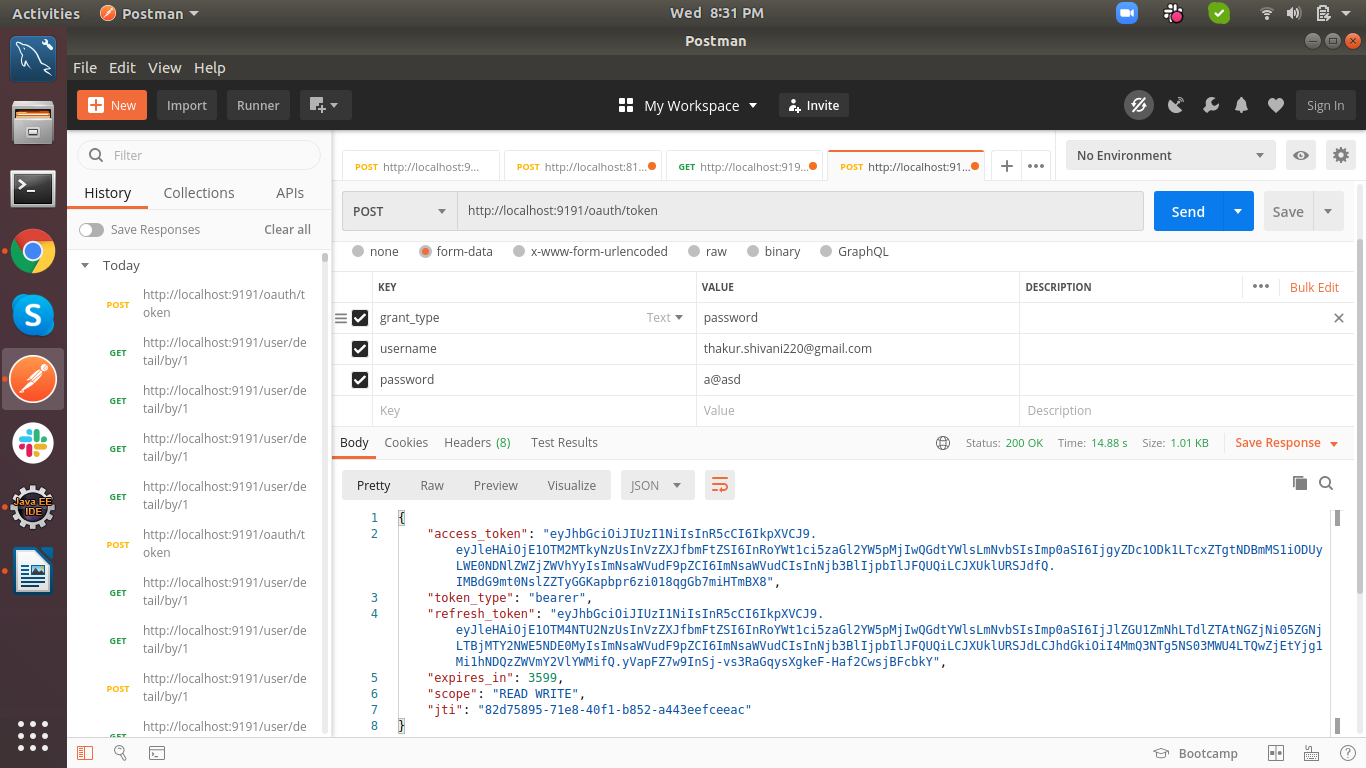
password : (enter registor password)

**Authorization** : Type : Basic Auth

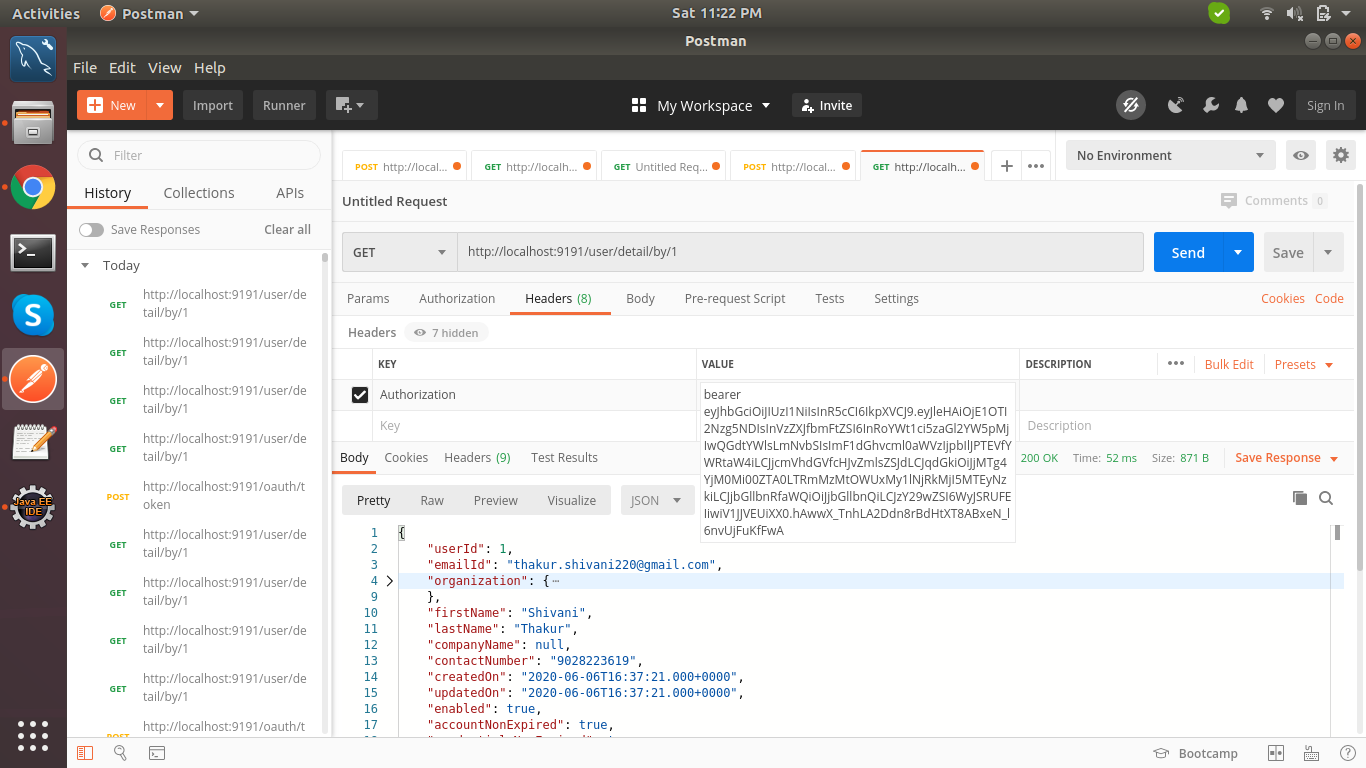
Username : (enter clientname from AuthorizationSreverConfig file)

Password : (enter secret from AuthorizationSreverConfig file)

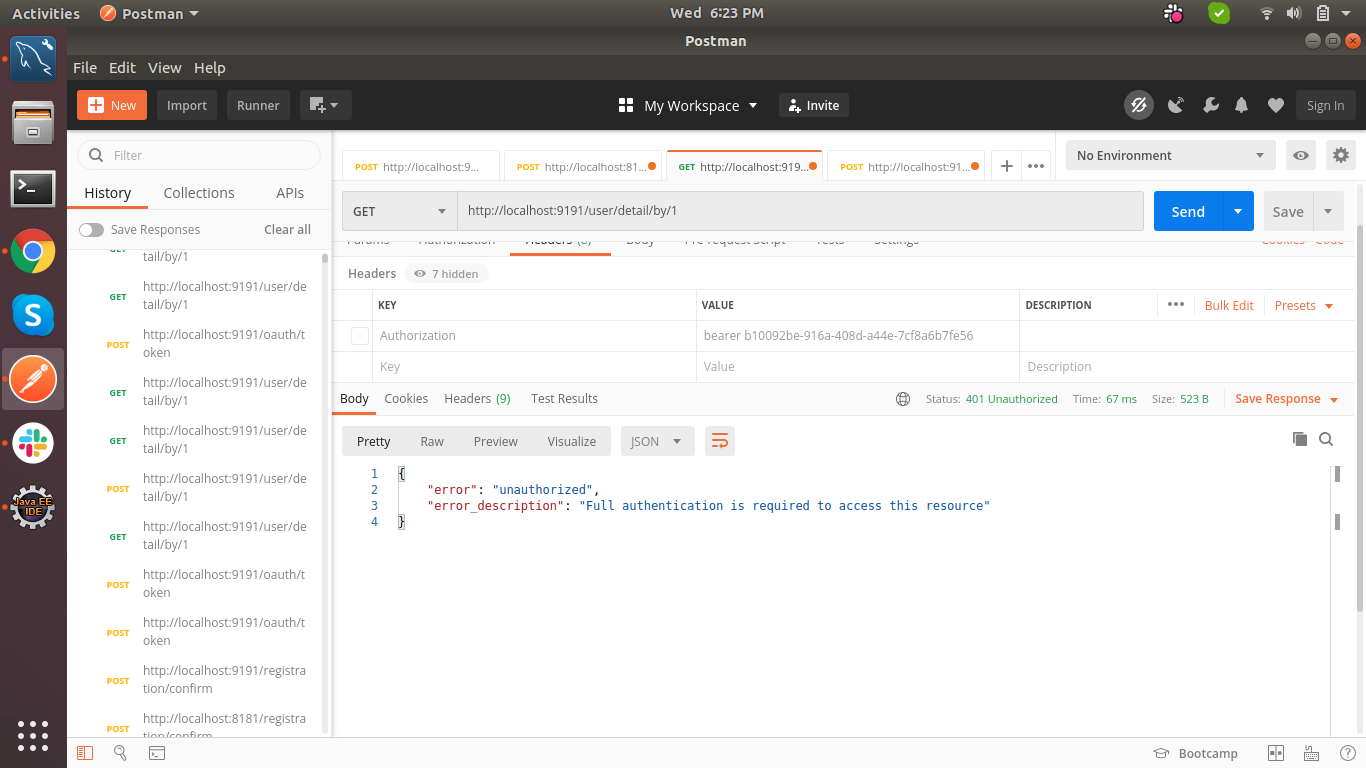
**4. Client & Secret & user credentials:**

****

**5. Get user by Id API with access\_token :**

****

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

****