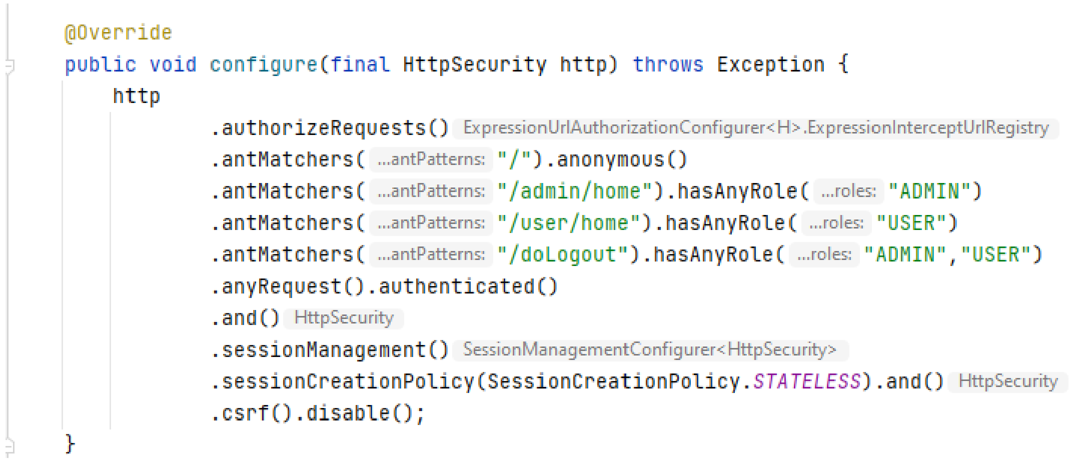
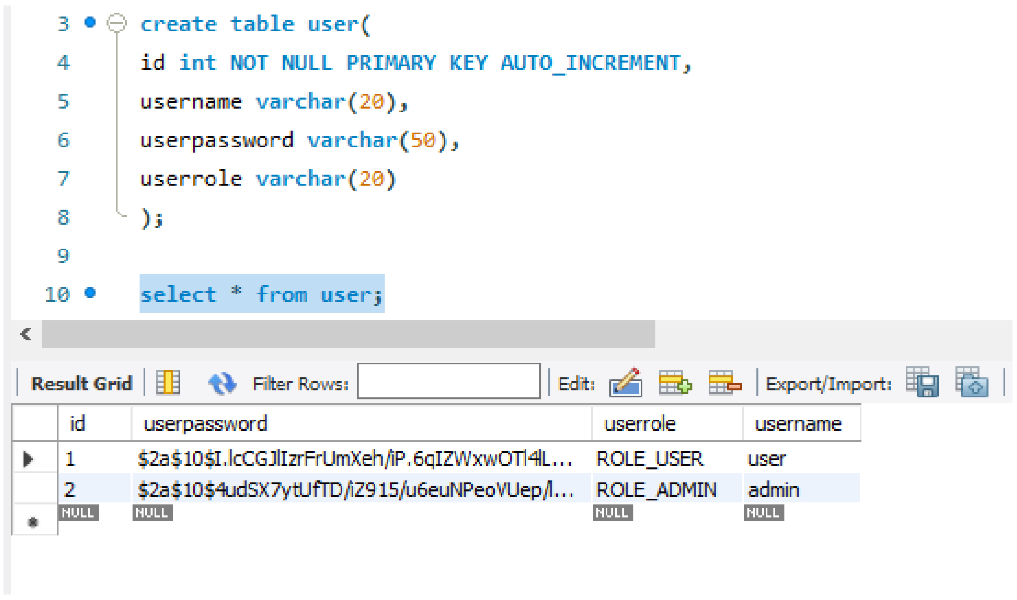
# <u>Session: Spring Security oath 2</u>

# <u>Assignment</u>

1) Implement oauth 2 using spring security and authenticate a user which is saved in database using spring data jpa.



```
@Override
public void configure(final HttpSecurity http) throws Exception {
    http
            .authorizeRequests() ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlRegistry
            .antMatchers( ...antPatterns: "/").anonymous()
            .antMatchers( ...antPatterns: "/admin/home").hasAnyRole( ...roles: "ADMIN")
            .antMatchers( ...antPatterns: "/user/home").hasAnyRole( ...roles: "USER")
            .antMatchers( ...antPatterns: "/doLogout").hasAnyRole( ...roles: "ADMIN","USER")
            .anyRequest().authenticated()
            .and() HttpSecurity
            .sessionManagement() SessionManagementConfigurer<HttpSecurity>
            .sessionCreationPolicy(SessionCreationPolicy.STATELESS).and() HttpSecurity
            .csrf().disable();
}
```

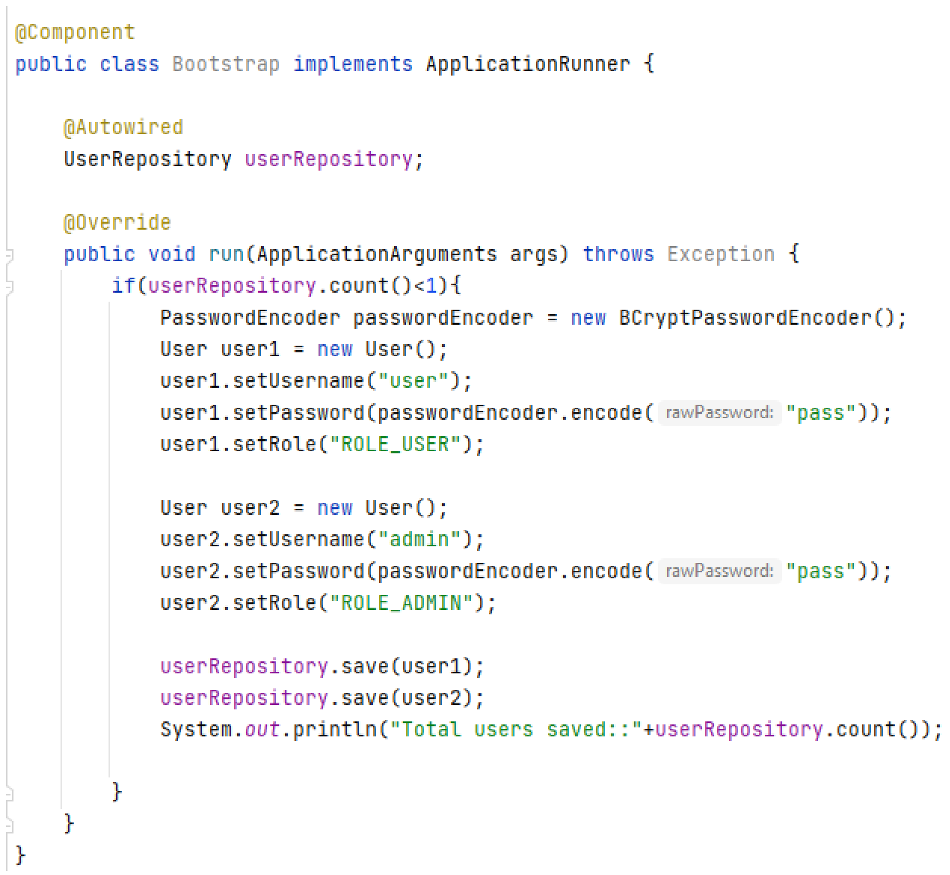


```
create table user(
id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
username varchar(20),
userpassword varchar(50),
userrole varchar(20)
);

select * from user;
```

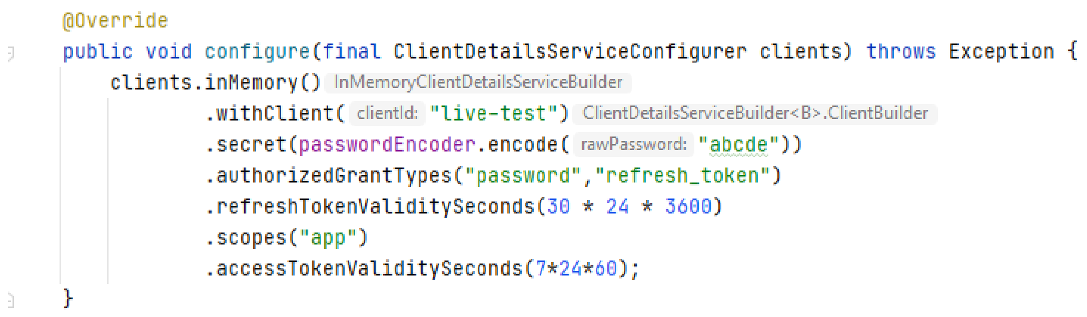
Result Grid | Filter Rows: | Edit: | Export/Import:

| | id | userpassword | userrole | username |
|---|---|---|---|---|
| ▶ | 1 | $2a$10$I.lcCGJlIzrFrUmXeh/iP.6qIZWxwOTl4lL... | ROLE_USER | user |
| | 2 | $2a$10$4udSX7ytUfTD/iZ915/u6euNPeoVUep/l... | ROLE_ADMIN | admin |
| ● | NULL | NULL | NULL | NULL |

2) Grant different Roles to different users and make sure that only authorized users of a given type can access the resource.



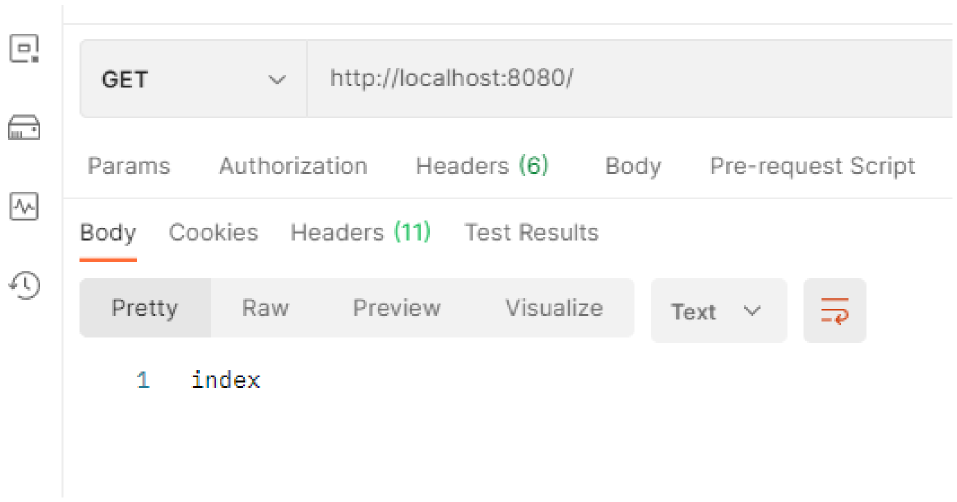
```
@Component
public class Bootstrap implements ApplicationRunner {

    @Autowired
    UserRepository userRepository;

    @Override
    public void run(ApplicationArguments args) throws Exception {
        if(userRepository.count()<1){
            PasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
            User user1 = new User();
            user1.setUsername("user");
            user1.setPassword(passwordEncoder.encode( rawPassword: "pass"));
            user1.setRole("ROLE_USER");

            User user2 = new User();
            user2.setUsername("admin");
            user2.setPassword(passwordEncoder.encode( rawPassword: "pass"));
            user2.setRole("ROLE_ADMIN");

            userRepository.save(user1);
            userRepository.save(user2);
            System.out.println("Total users saved::"+userRepository.count());

        }
    }
}
```



```
@Override
public void configure(final ClientDetailsServiceConfigurer clients) throws Exception {
    clients.inMemory() InMemoryClientDetailsServiceBuilder
            .withClient( clientId: "live-test") ClientDetailsServiceBuilder<B>.ClientBuilder
            .secret(passwordEncoder.encode( rawPassword: "abcde"))
            .authorizedGrantTypes("password","refresh_token")
            .refreshTokenValiditySeconds(30 * 24 * 3600)
            .scopes("app")
            .accessTokenValiditySeconds(7*24*60);
}
```

Checks to verify that only authorized users of a given type can access the resource.

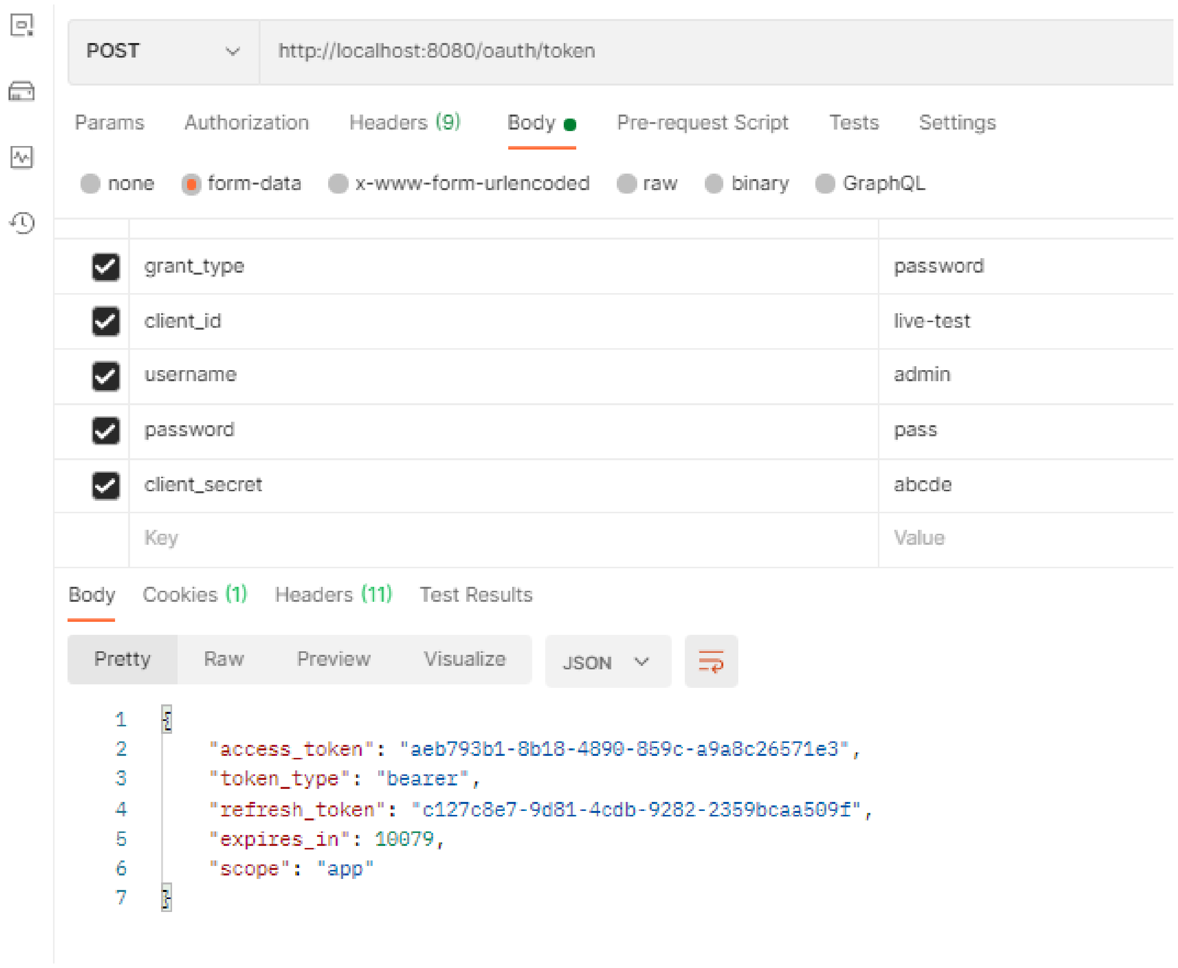
- Request call for Non Logged in / anonymous users



http://localhost:8080/

```
index
```

- Access token request call for admin

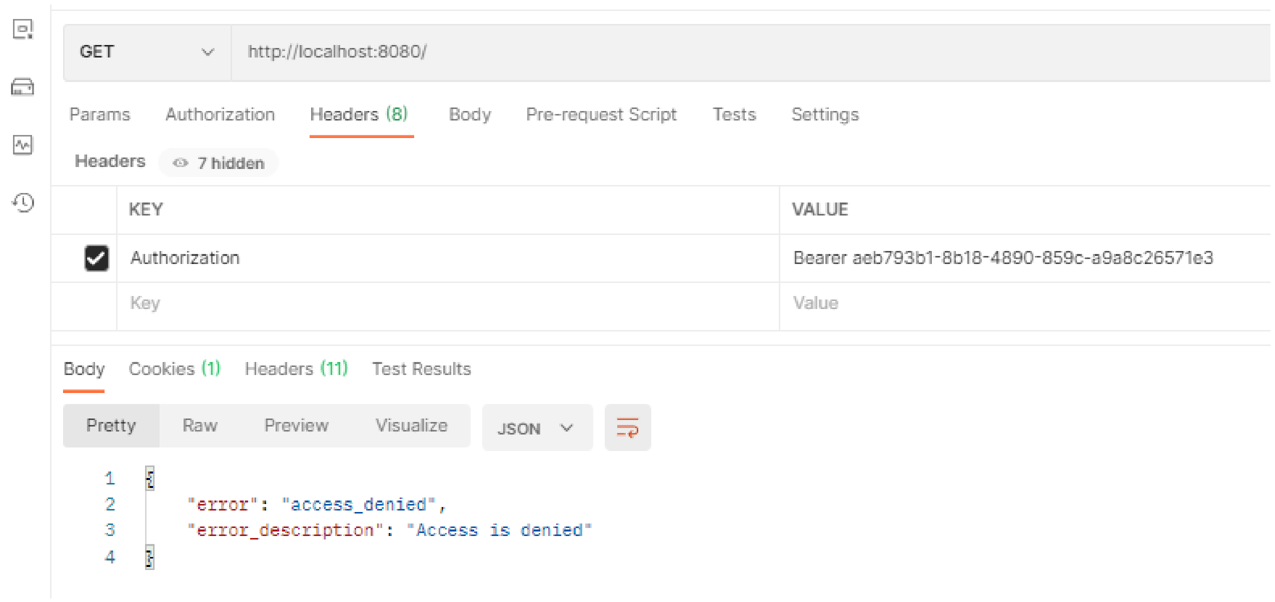


http://localhost:8080/oauth/token

| | |
|---|---|
| grant_type | password |
| client_id | live-test |
| username | admin |
| password | pass |
| client_secret | abcde |
| Key | Value |

```
{
    "access_token": "aeb793b1-8b18-4890-859c-a9a8c26571e3",
    "token_type": "bearer",
    "refresh_token": "c127c8e7-9d81-4cdb-9282-2359bcaa509f",
    "expires_in": 10079,
    "scope": "app"
}
```

- Request call for non-logged in user using access token of admin



GET http://localhost:8080/

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

Headers 7 hidden

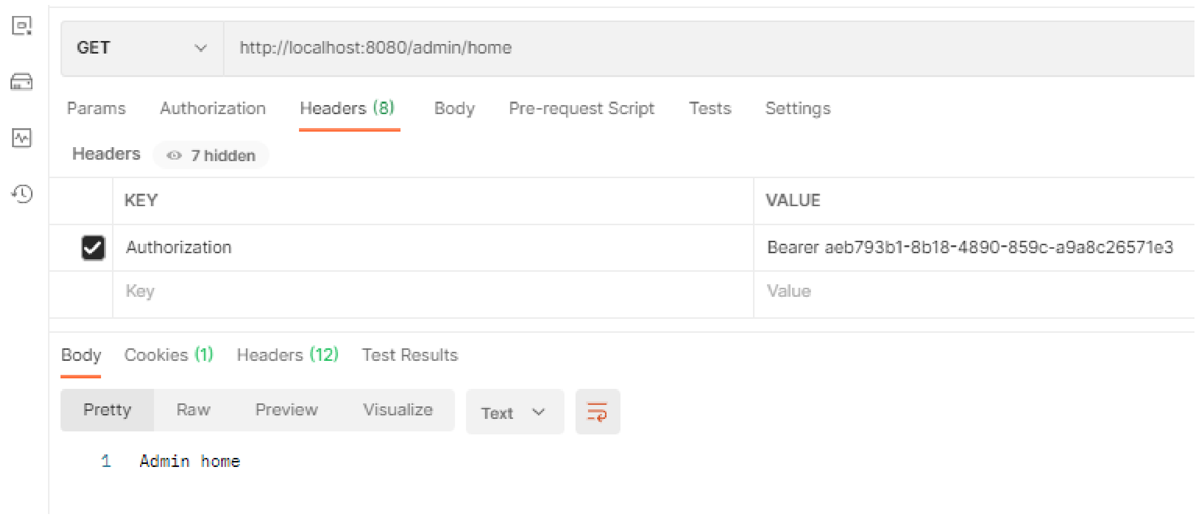
| | KEY | VALUE |
|---|---|---|
| ✓ | Authorization | Bearer aeb793b1-8b18-4890-859c-a9a8c26571e3 |
| | Key | Value |

Body | Cookies (1) | Headers (11) | Test Results

Pretty | Raw | Preview | Visualize | JSON

```
{
    "error": "access_denied",
    "error_description": "Access is denied"
}
```

- Request call for admin login



GET http://localhost:8080/admin/home

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

Headers 7 hidden

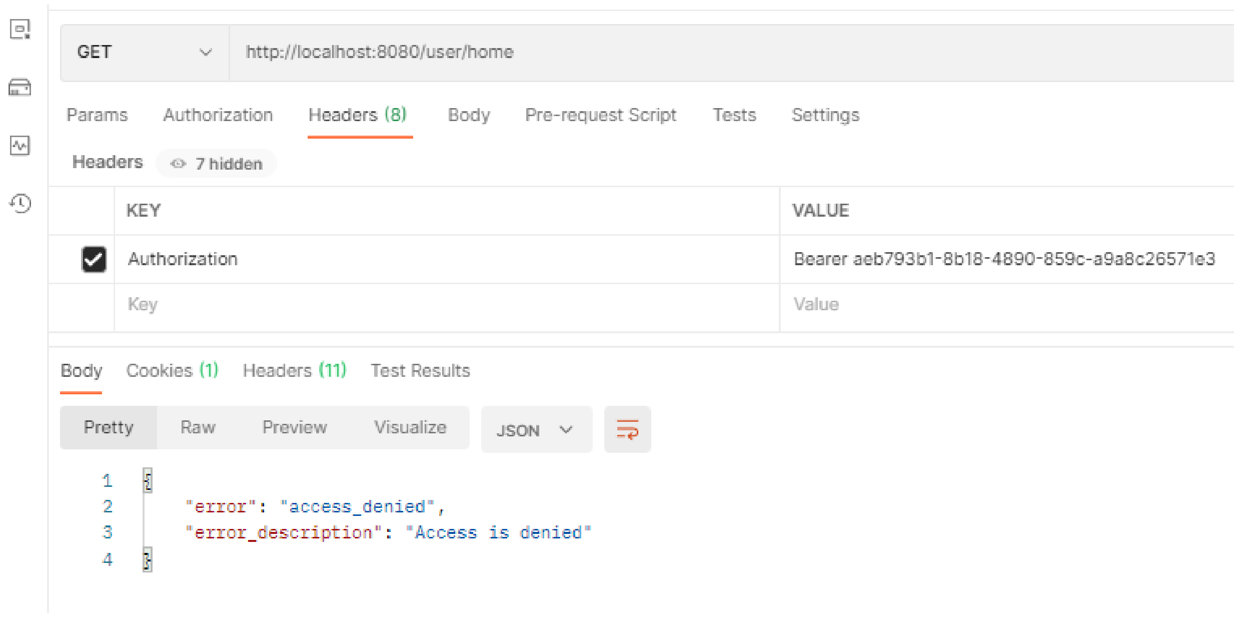
| | KEY | VALUE |
|---|---|---|
| ✓ | Authorization | Bearer aeb793b1-8b18-4890-859c-a9a8c26571e3 |
| | Key | Value |

Body | Cookies (1) | Headers (12) | Test Results

Pretty | Raw | Preview | Visualize | Text

```
Admin home
```

- Request call for user login using access token of admin

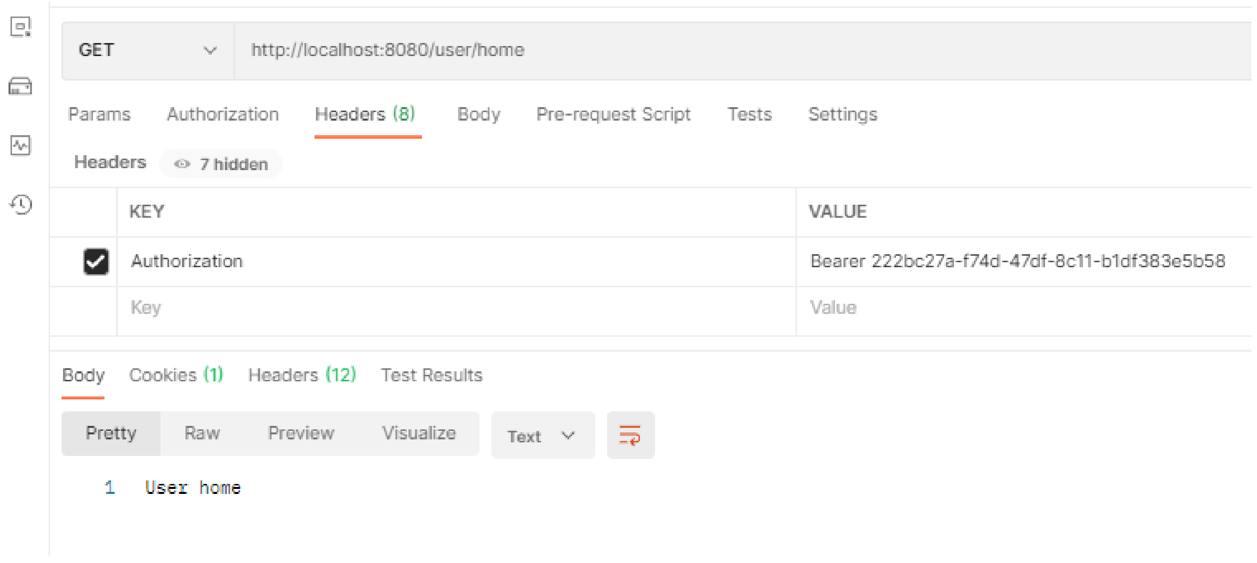


GET http://localhost:8080/user/home

| KEY | VALUE |
|---|---|
| ✓ Authorization | Bearer aeb793b1-8b18-4890-859c-a9a8c26571e3 |
| Key | Value |

```
{
    "error": "access_denied",
    "error_description": "Access is denied"
}
```

- Request call for user login

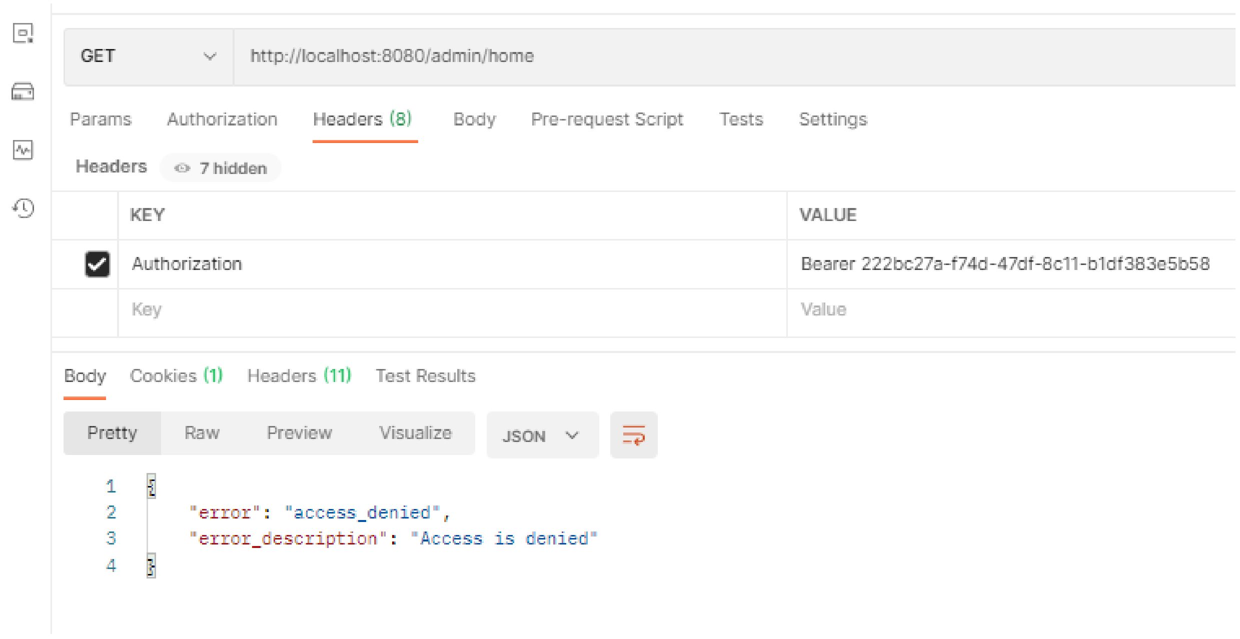


GET http://localhost:8080/user/home

| KEY | VALUE |
|---|---|
| ✓ Authorization | Bearer 222bc27a-f74d-47df-8c11-b1df383e5b58 |
| Key | Value |

```
User home
```

- Request call for admin login using access token of user

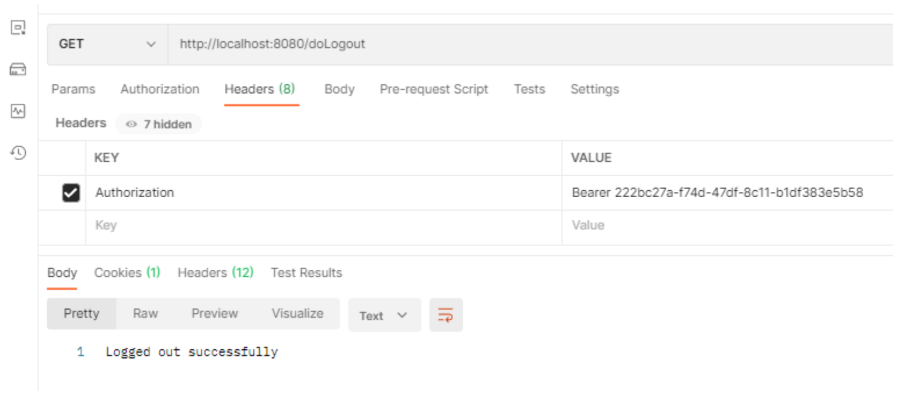


GET http://localhost:8080/admin/home

| | KEY | VALUE |
|---|---|---|
| ✓ | Authorization | Bearer 222bc27a-f74d-47df-8c11-b1df383e5b58 |
| | Key | Value |

```
{
    "error": "access_denied",
    "error_description": "Access is denied"
}
```

- Request call for user logout using access token of user

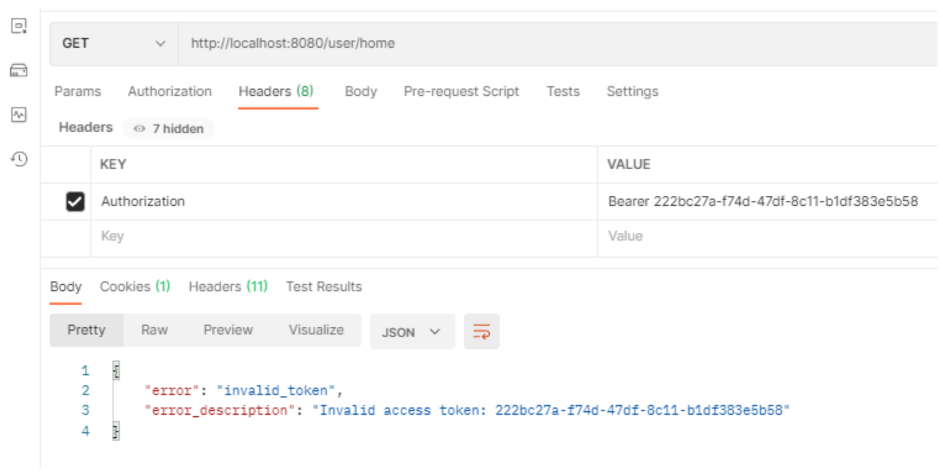


GET http://localhost:8080/doLogout

| | KEY | VALUE |
|---|---|---|
| ✓ | Authorization | Bearer 222bc27a-f74d-47df-8c11-b1df383e5b58 |
| | Key | Value |

```
Logged out successfully
```

- Request call for user login after logout



GET http://localhost:8080/user/home

| | KEY | VALUE |
|---|---|---|
| ✓ | Authorization | Bearer 222bc27a-f74d-47df-8c11-b1df383e5b58 |
| | Key | Value |

```
{
    "error": "invalid_token",
    "error_description": "Invalid access token: 222bc27a-f74d-47df-8c11-b1df383e5b58"
}
```