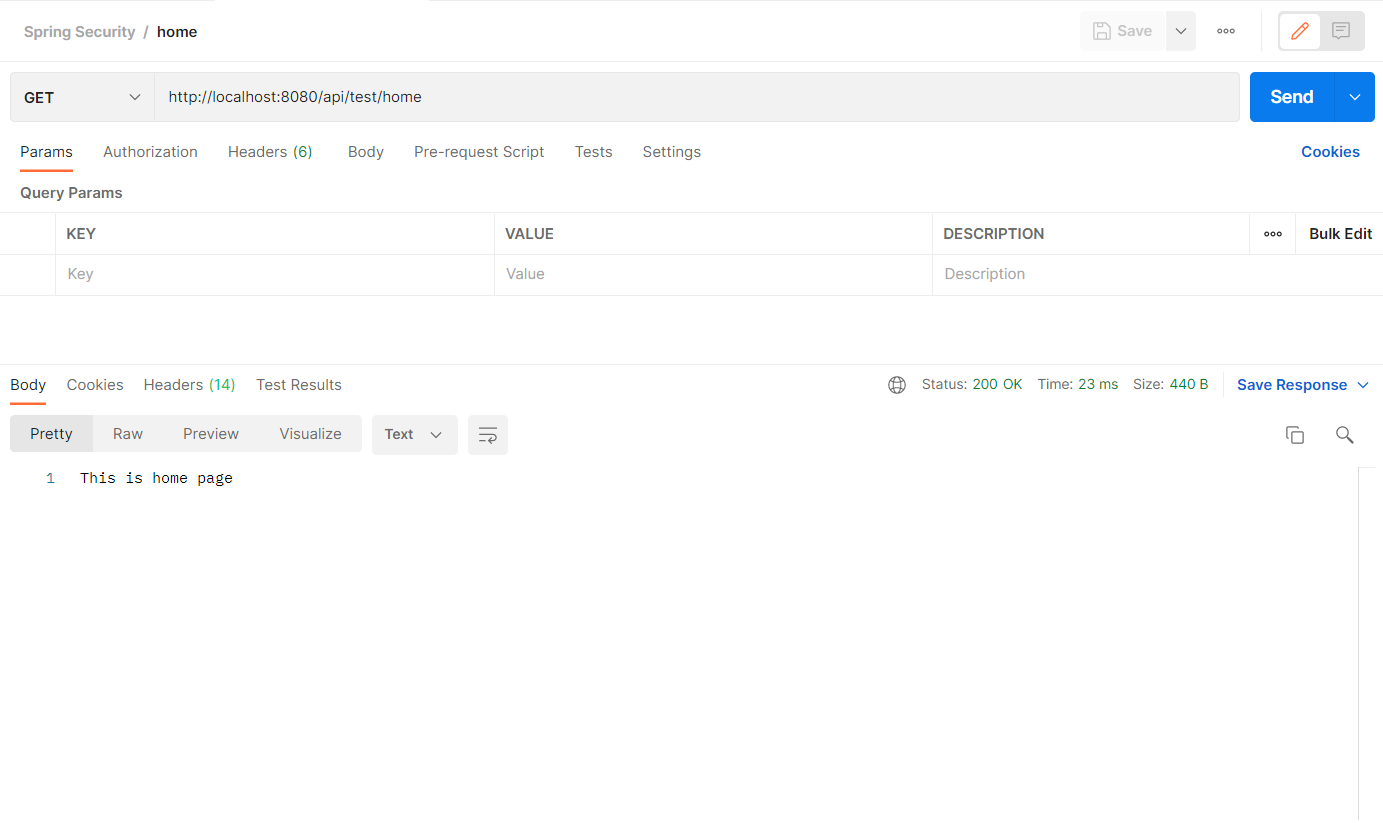
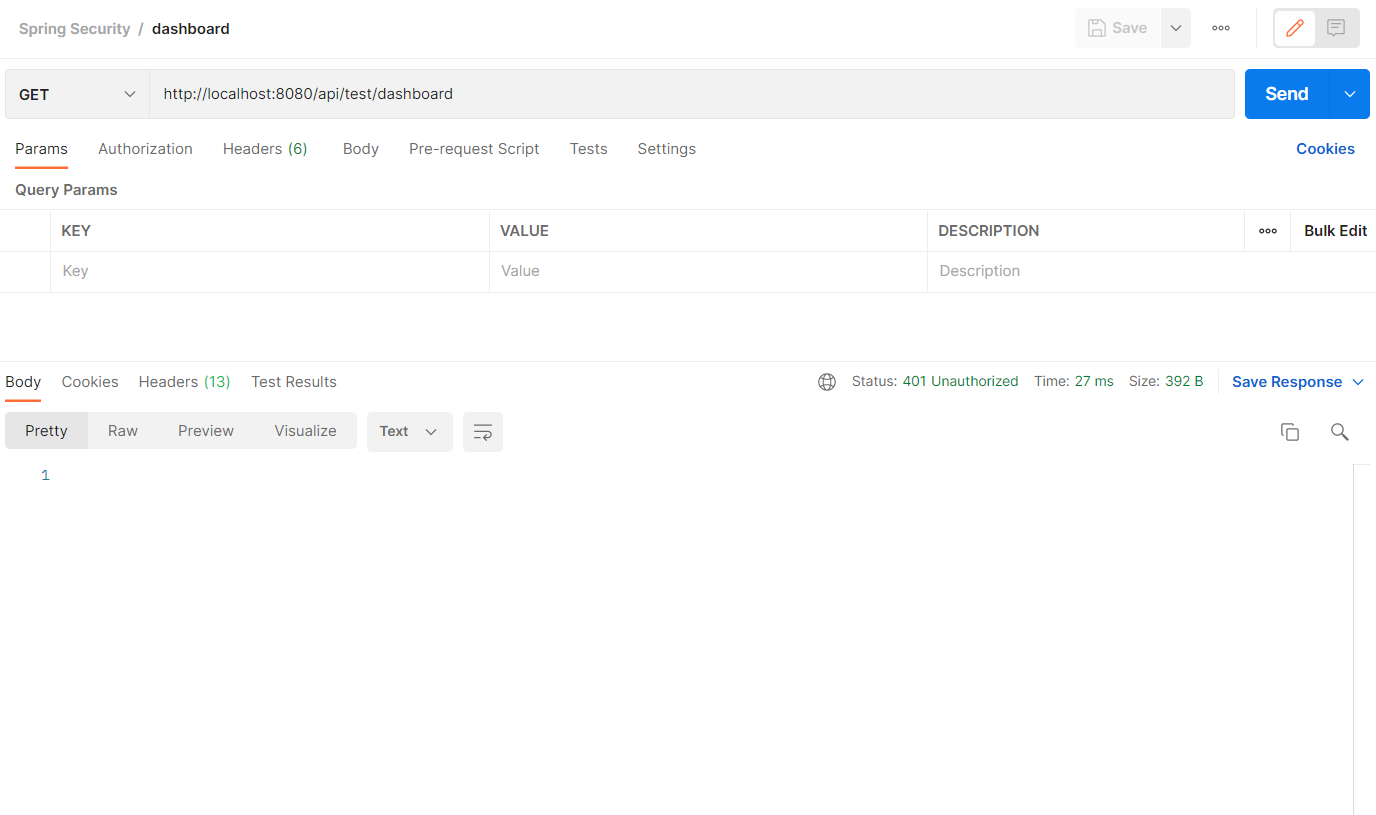
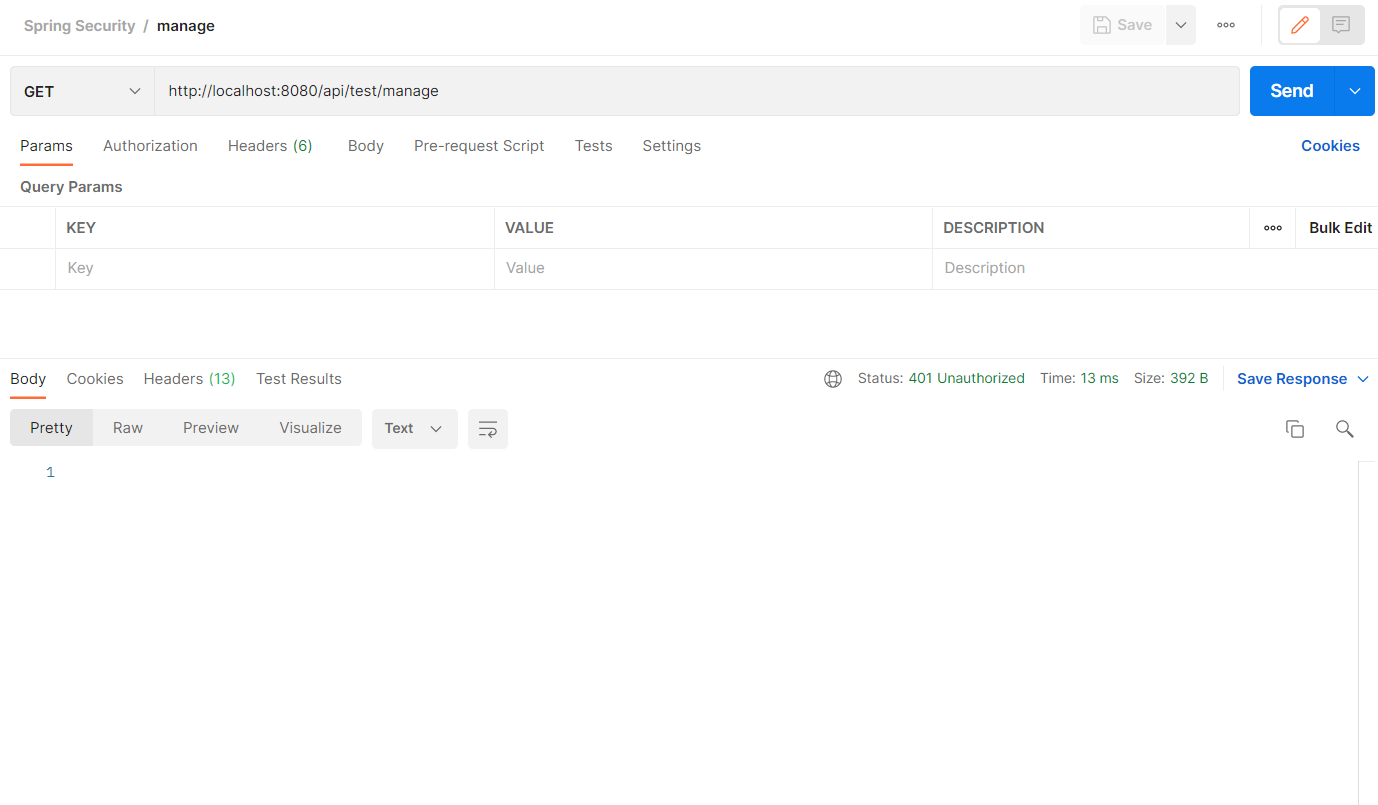
**7. Spring Security**

1. Create a spring boot application with mongodb connection
2. Create a controller class and add 3 endpoints

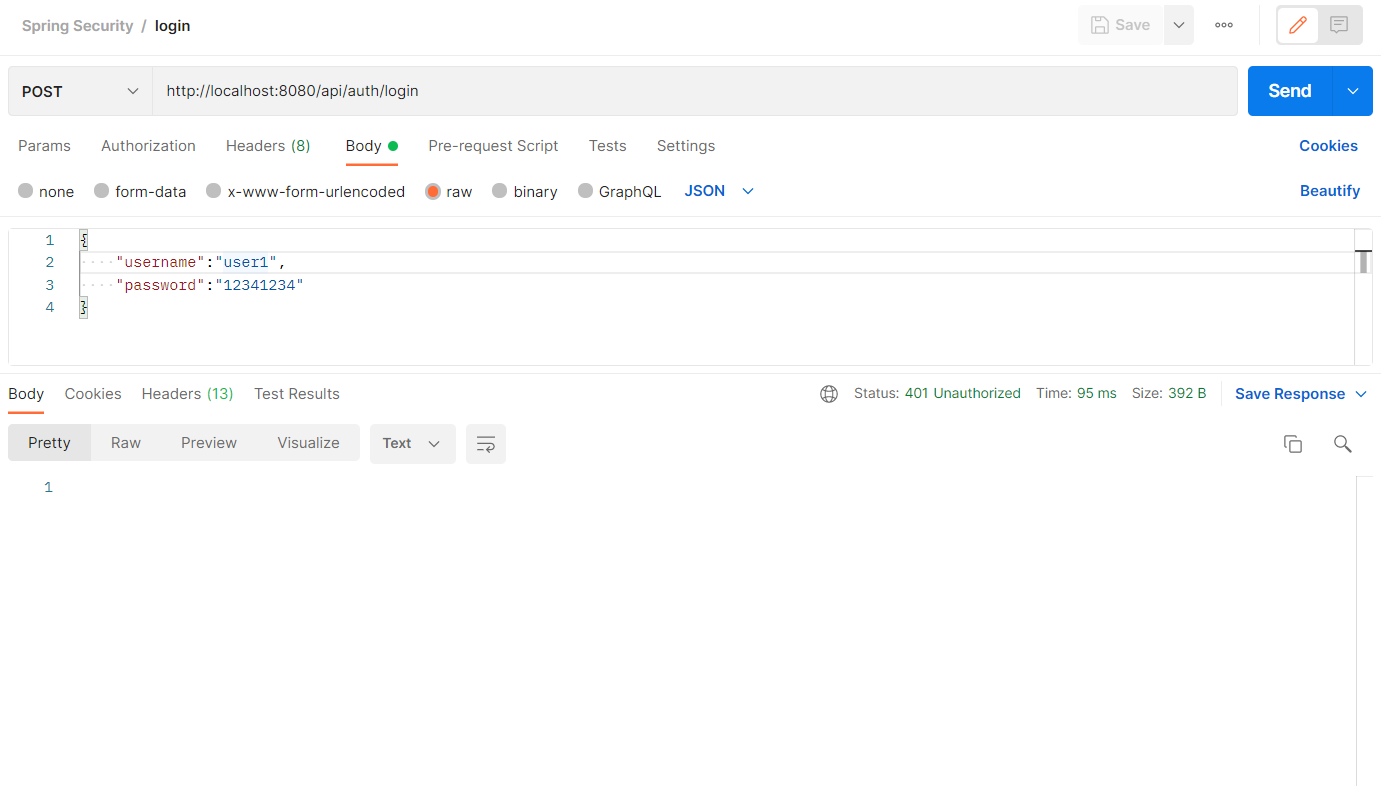
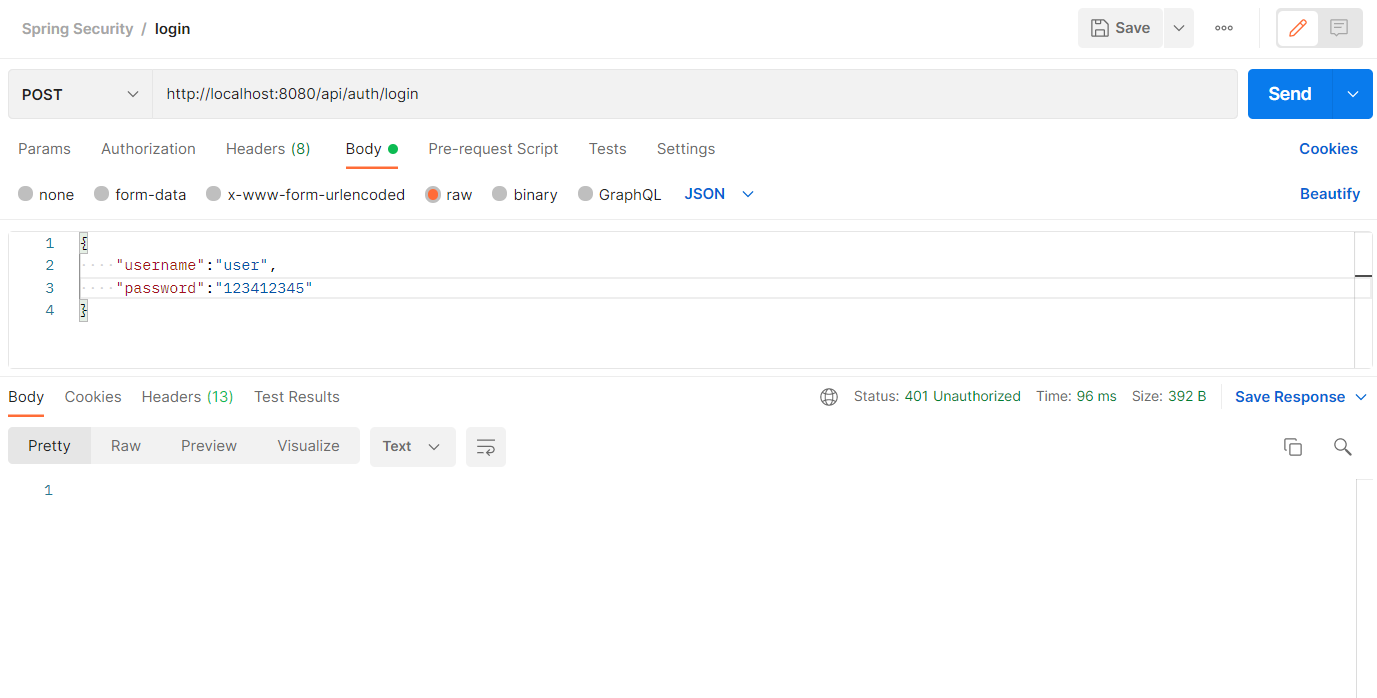
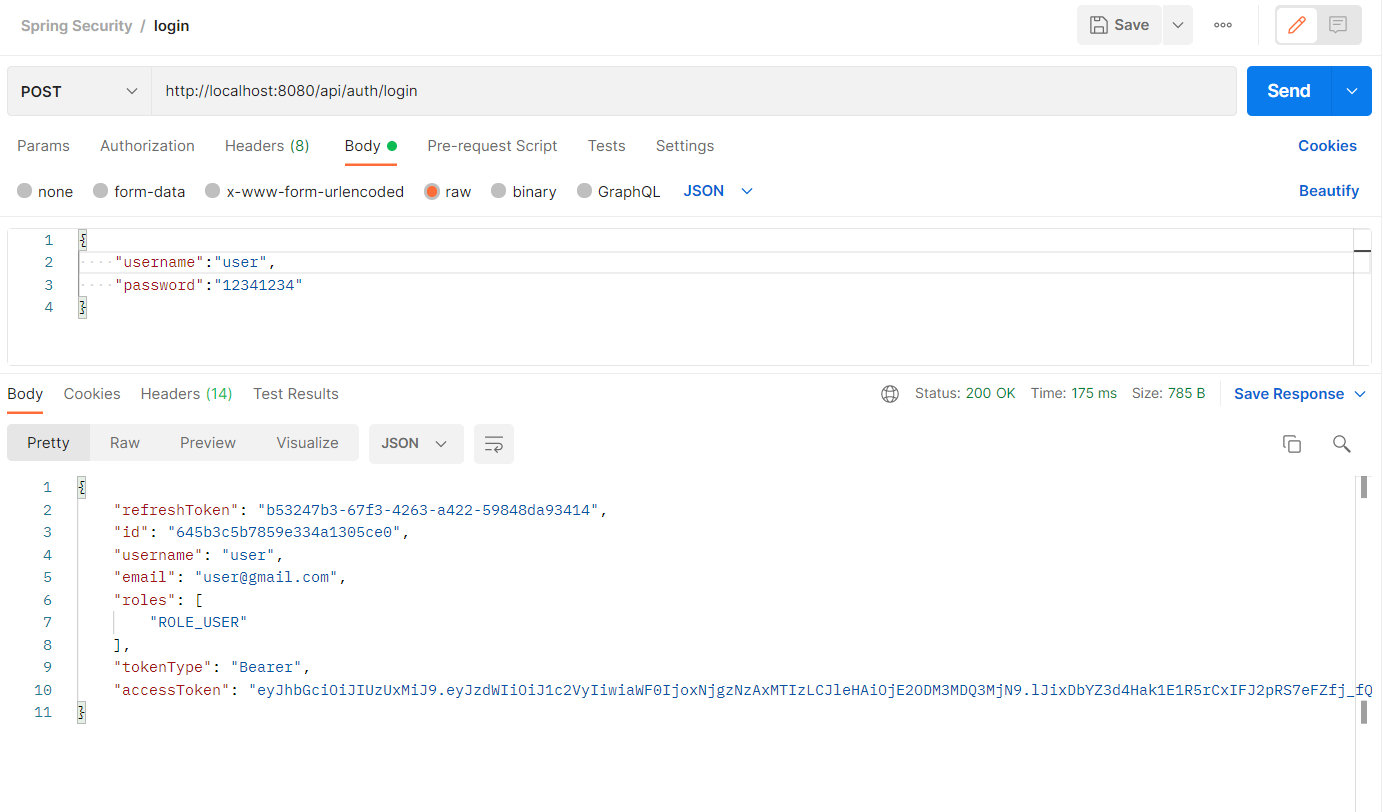
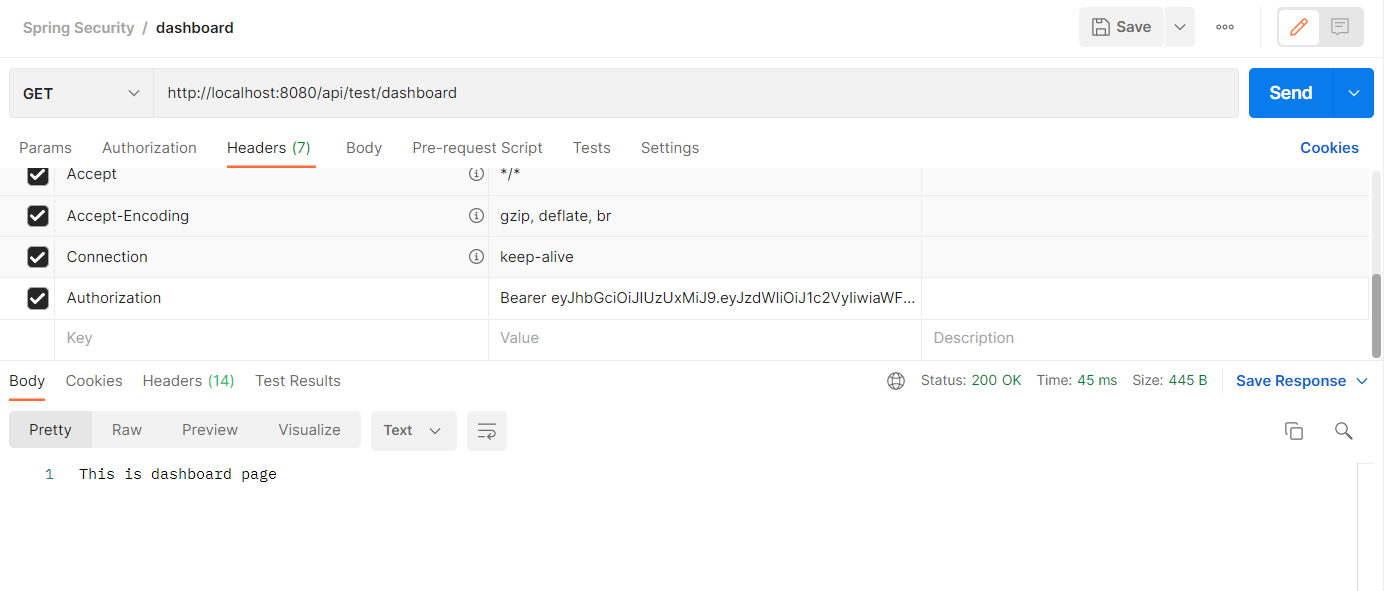
* *“/home” – return a string “this is home page”*
* *“/dashboard” - return a string “this is dashboard page”*
* *“/manage” - return a string “this is manage page”*
* *“/login – validate login and issue token*

1. Add spring security config file with jwt authentication
2. *“/home”* should be accessible for any user (no credentials needed)
3. In order to access *“/dashboard”* or *“/manage”* users should provide valid user credentials (username and password).
4. *“/dashboard”* should be only accessible for users with role ADMIN or USER
5. *“/manage”* should be only accessible for users with role ADMIN
6. Test the API with postman and attach screenshots for the result of these scenarios
   1. call /home endpoint (without token)
   2. call /dashboard and /manage endpoints without token

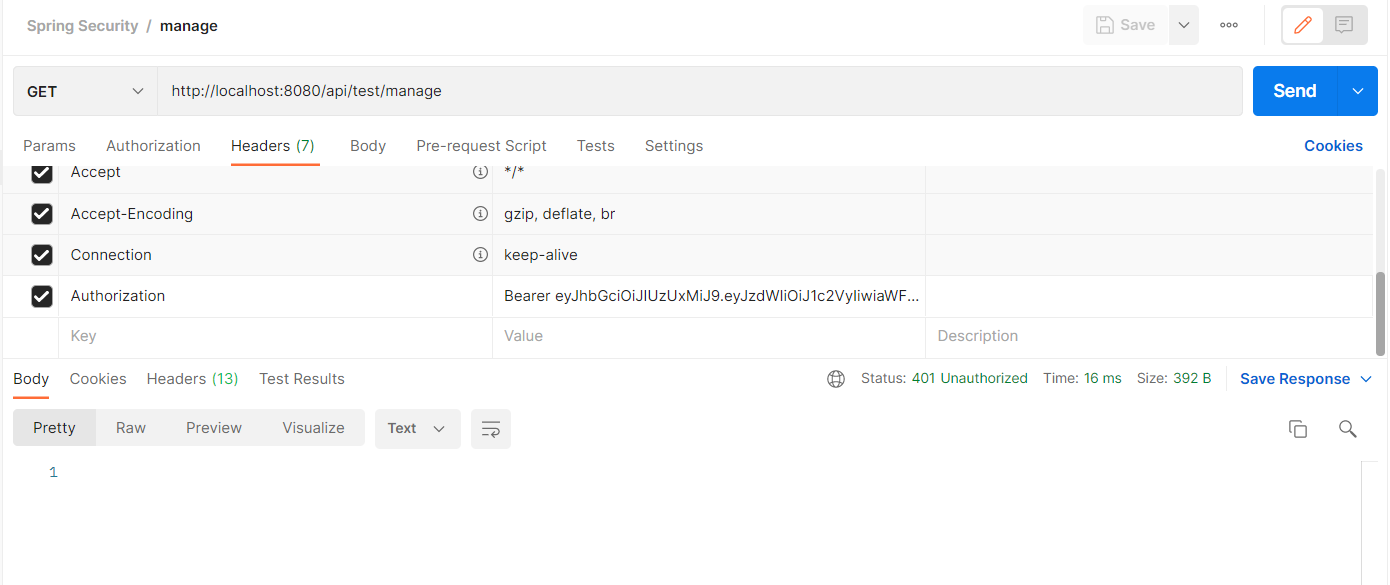
Navigate to dashboard without token



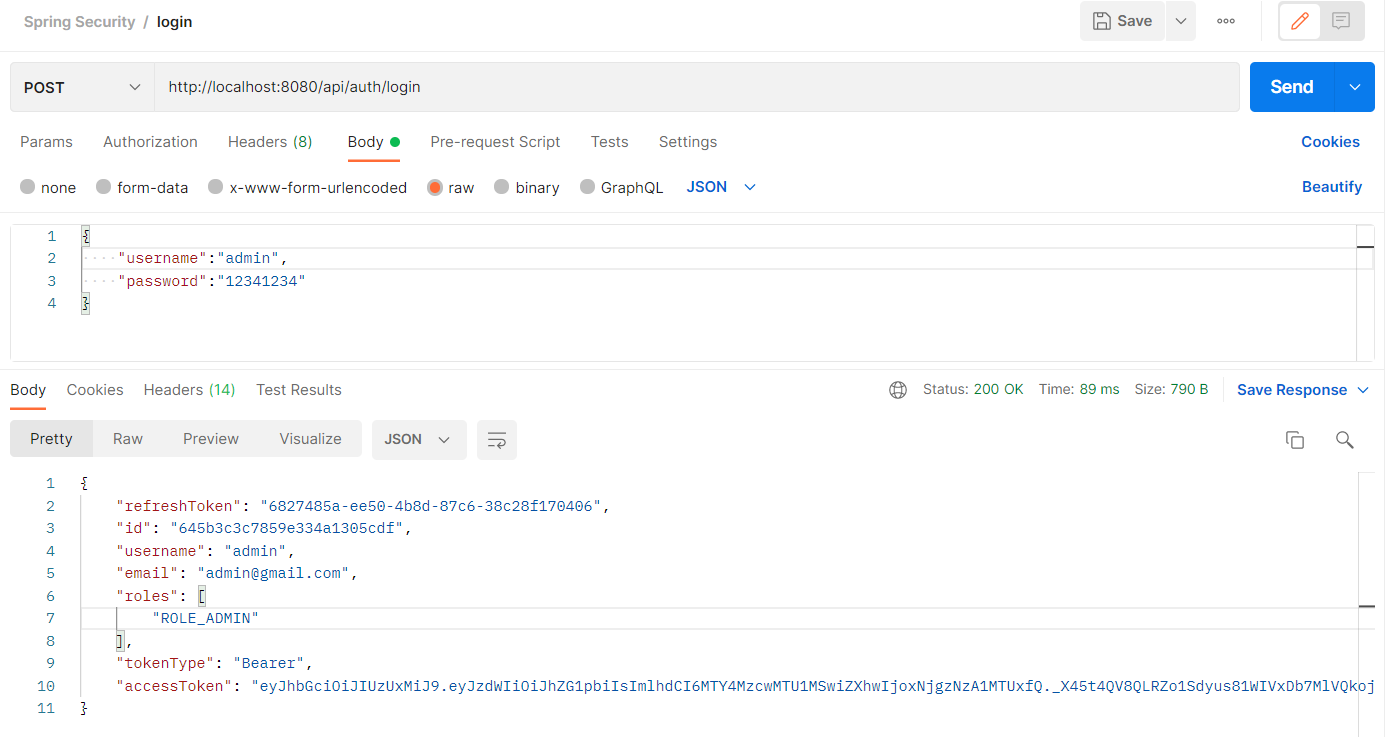
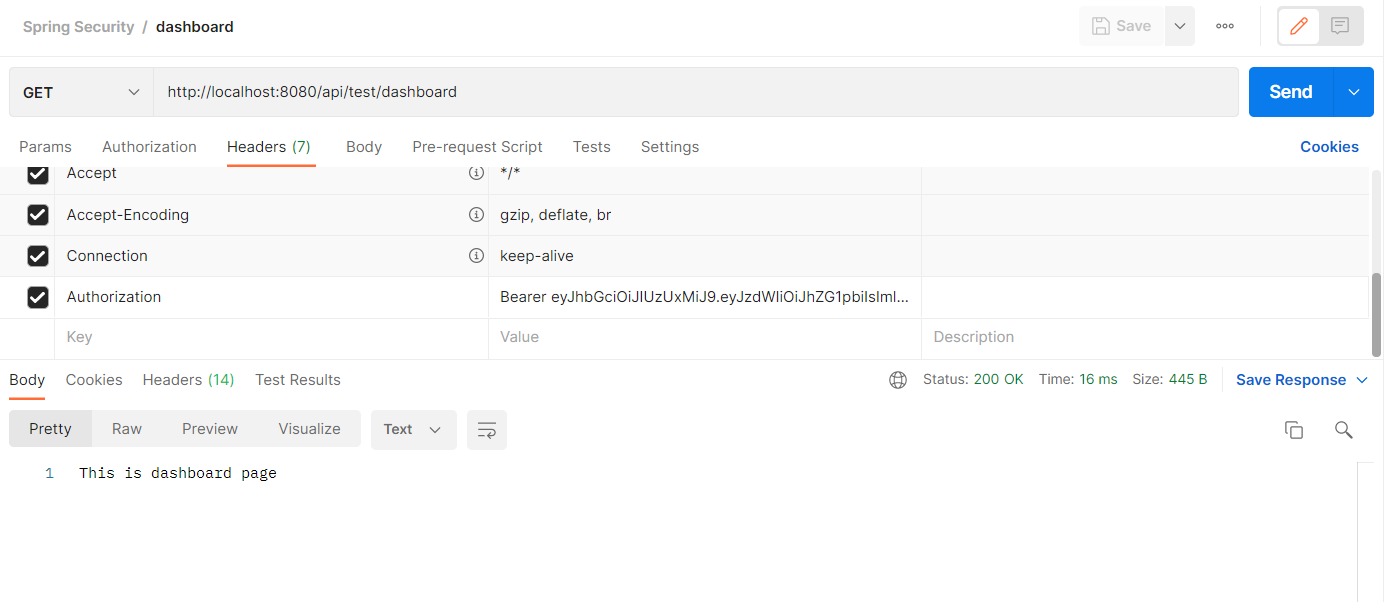
Navigate to manage without token

* 1. login with invalid username
  2. login with valid username but invalid password
  3. login with user with USER role
  4. call /dashboard and /manage endpoints with received token

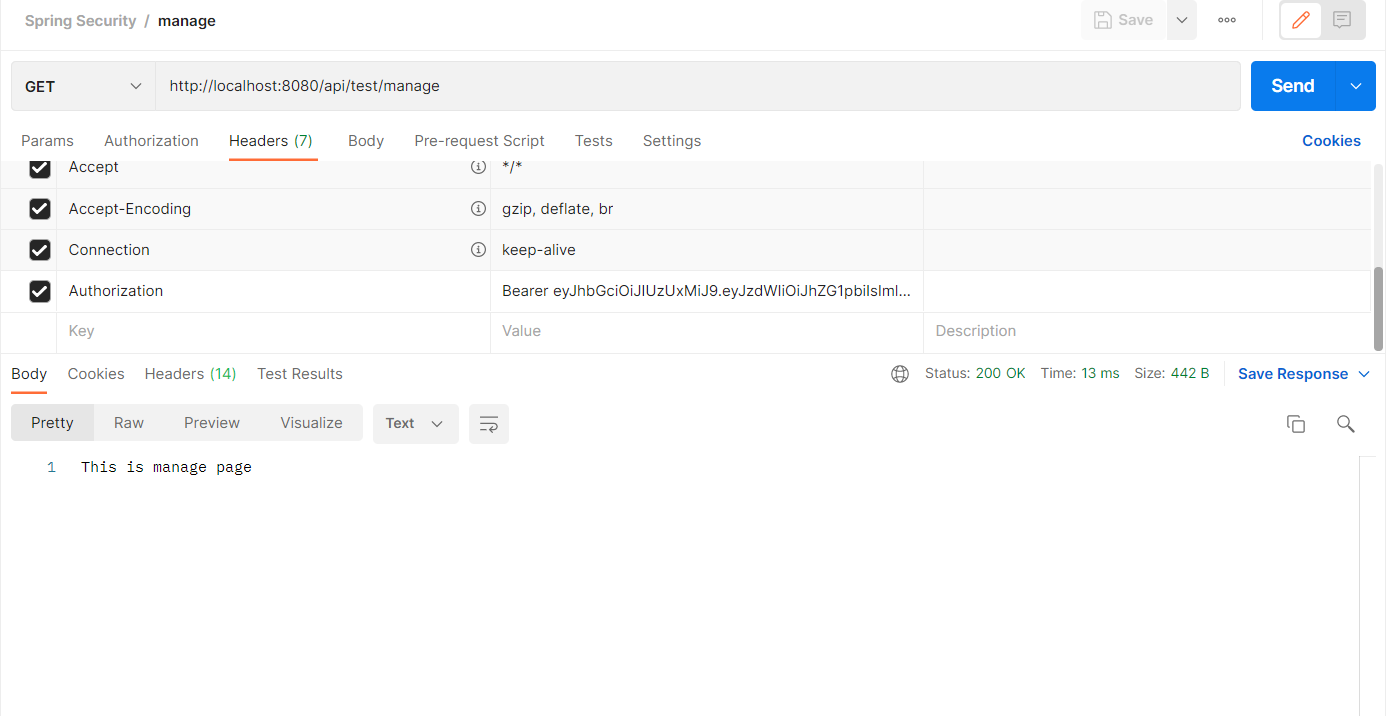
User navigate to dashboard



User navigate to manage

* 1. login with user with ADMIN role
  2. call /dashboard and /manage endpoints with received token

Admin navigate to dashboard



Admin navigate to manage

1. Create new branch “spring-security” and push the project you created
2. Add your codes and answer sheet to a directory named “spring-security-training” and push it to your training github repository
3. Create a pull request to main branch and assign it to your trainer