Name- Anish Dubey Roll No.- 2006345 Branch- Information Technology

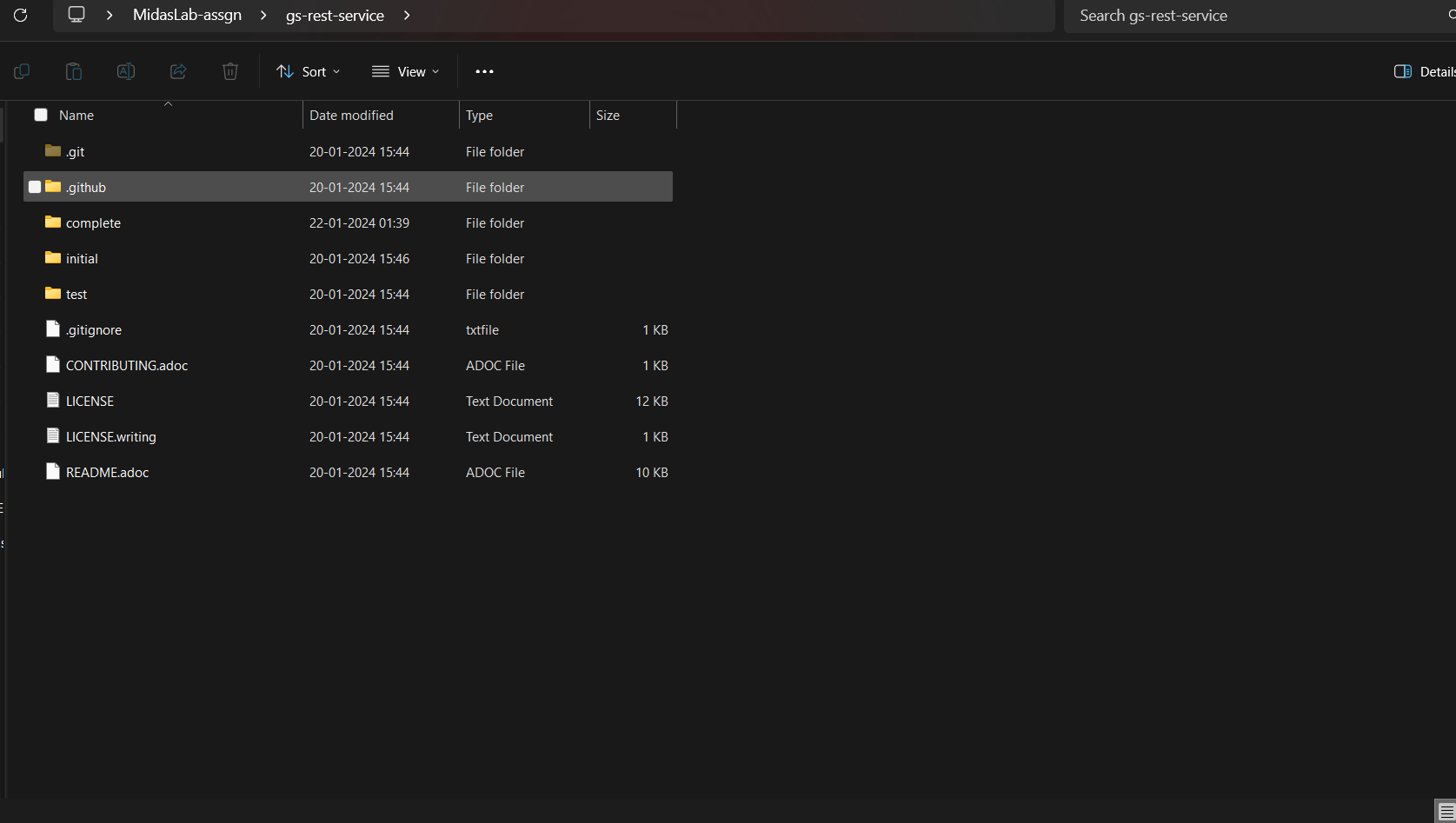
Email id- [anishd10lm@gmail.com](mailto:anishd10lm@gmail.com) / [2006345@kiit.ac.in](mailto:2006345@kiit.ac.in)

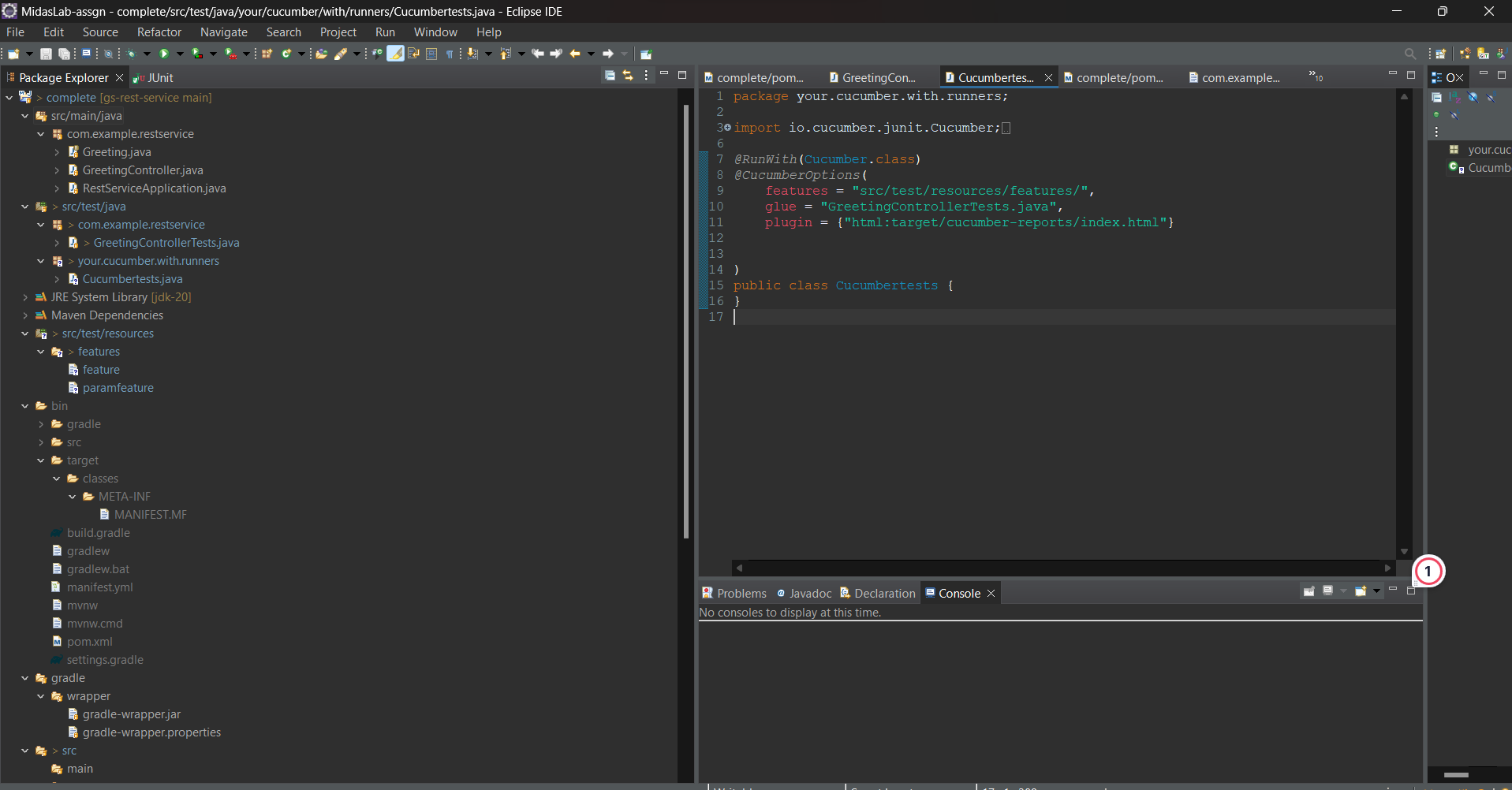
Github:- https://github.com/Anish10lm

Topic:

**Problem Statement 2:  
 In this task you will be testing a Java-based application built with Spring Boot, using the Cucumber framework. Cucumber is a tool for running automated acceptance tests written in a behaviour-driven development (BDD) style.**

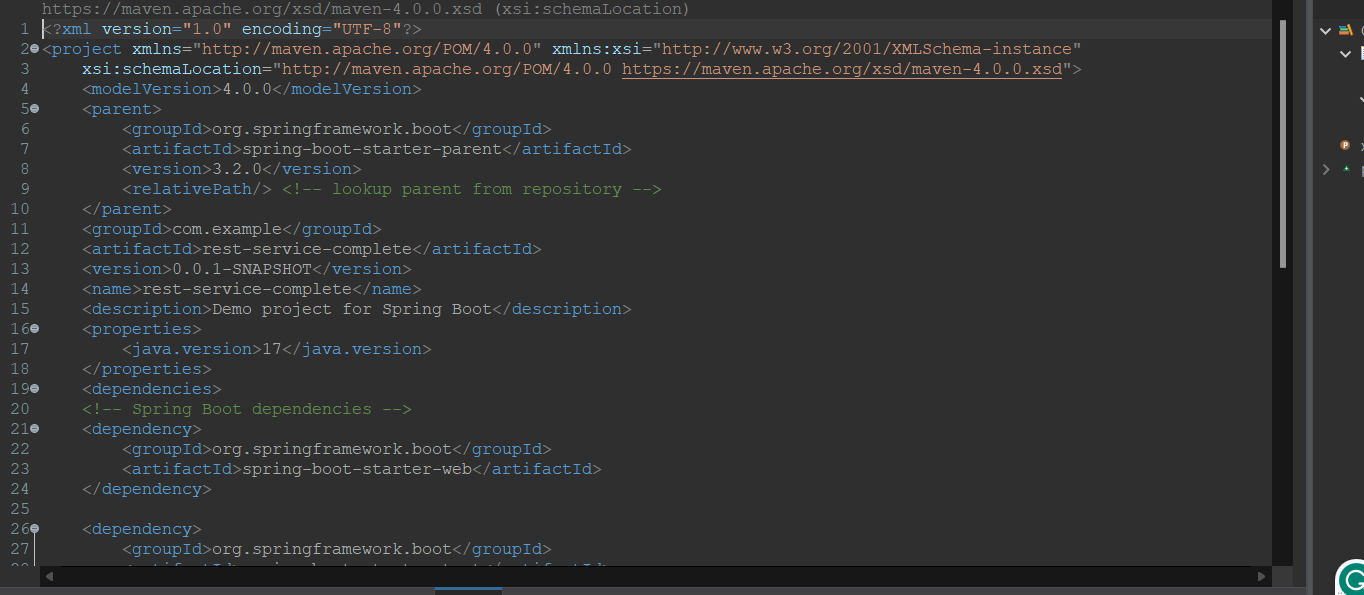
Step 1: Understand the Project 1. Clone the Spring Boot project from the provided GitHub link. 2. Explore the project to understand its structure and functionalities.

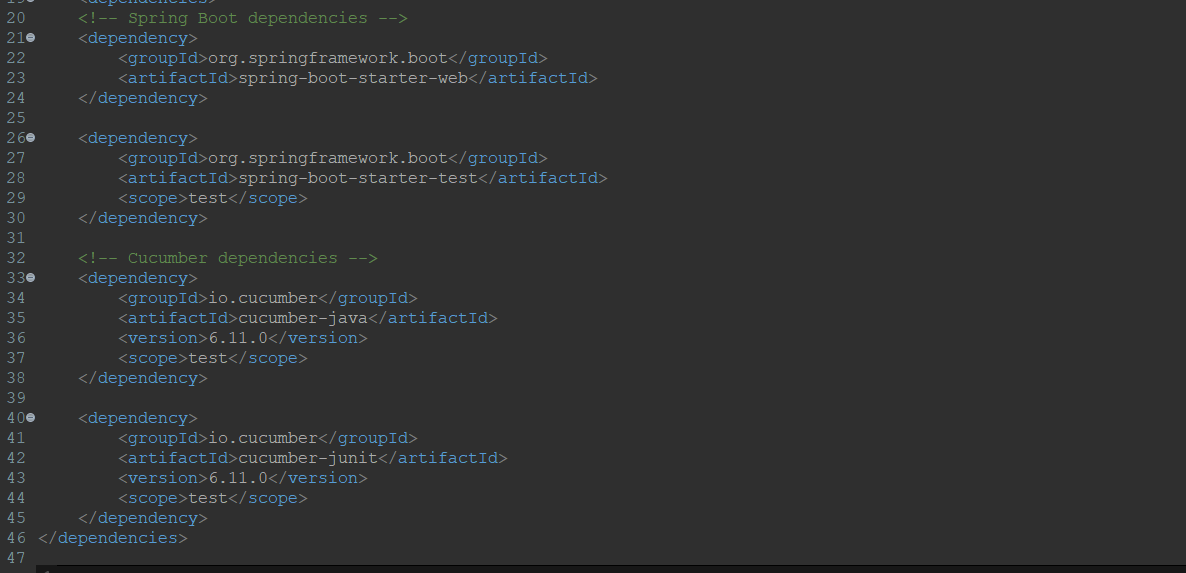


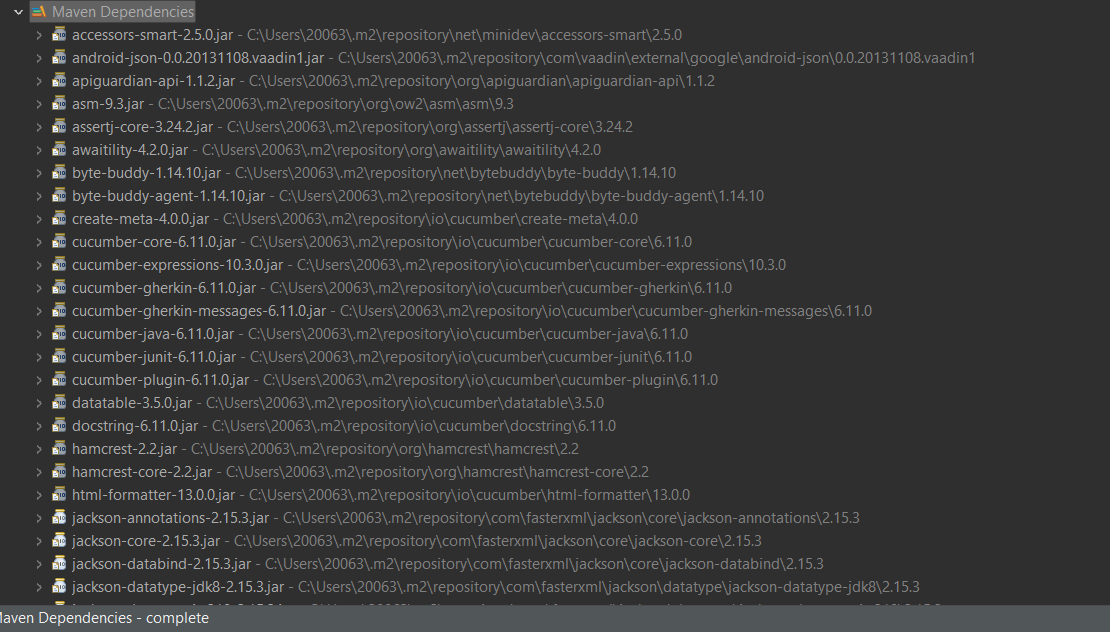


**Cloned the project from the given repository in the Eclipse IDE and went through with its structure and its contents.**

Step 2: Set Up Cucumber 1. Add Cucumber dependencies to the project's build configuration file (`pom.xml` for Maven or `build.gradle` for Gradle). 2. Set up the Cucumber environment in your IDE.

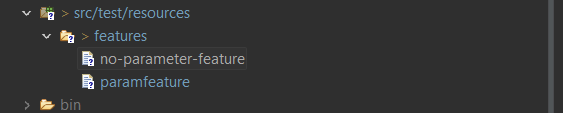




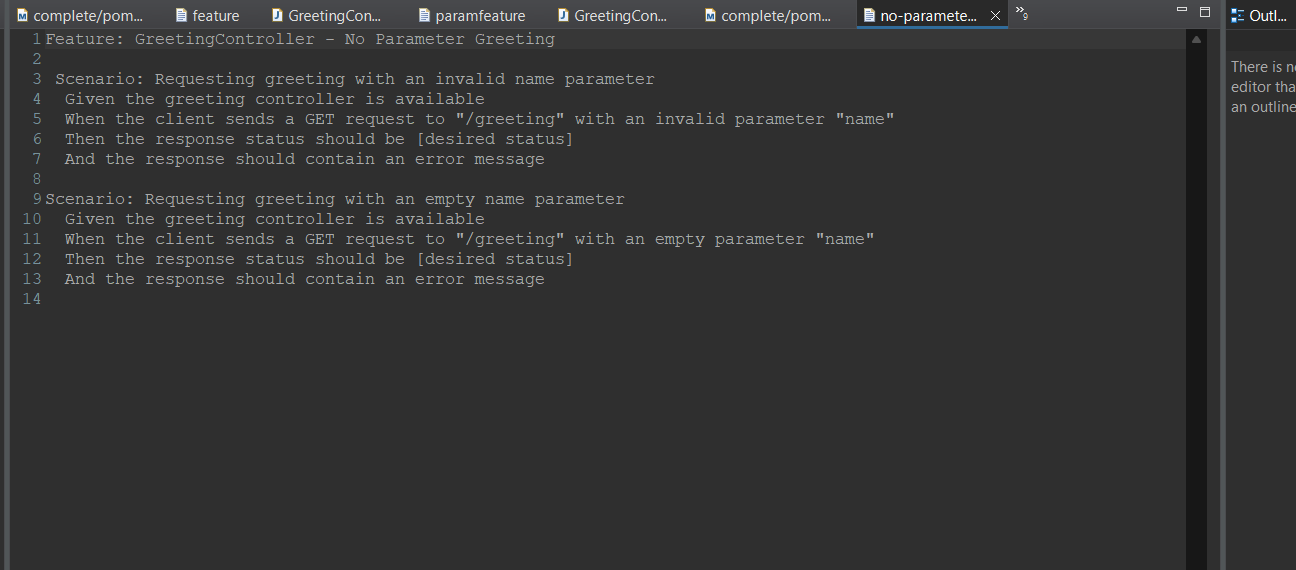


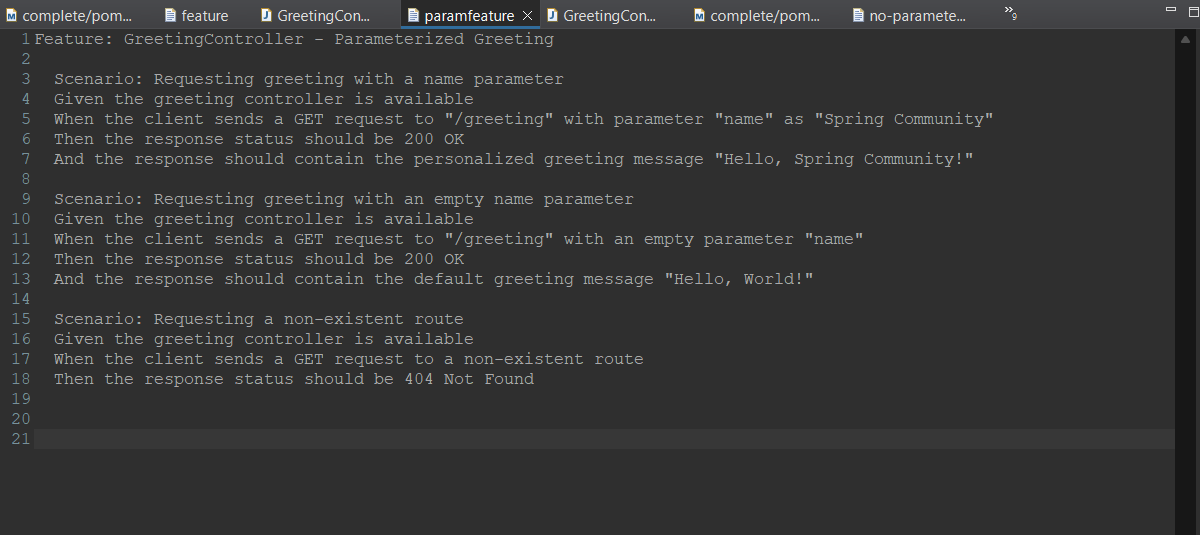
**Added the cucumber dependencies for the projects build configuration in pom.xml for Maven.**

Step 3: Write Cucumber Test Scenarios 1. Create feature files in the `src/test/resources` directory. 2. Write test scenarios in Gherkin language that describe the expected behavior of the application.



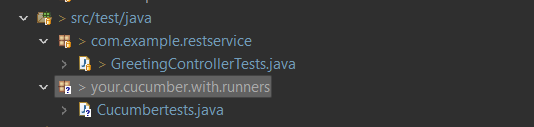
Created a features folder in src/test/resources and created two files with Gherkin language each for the parameter function and no parameter function given in the GreetingControllerTests.java



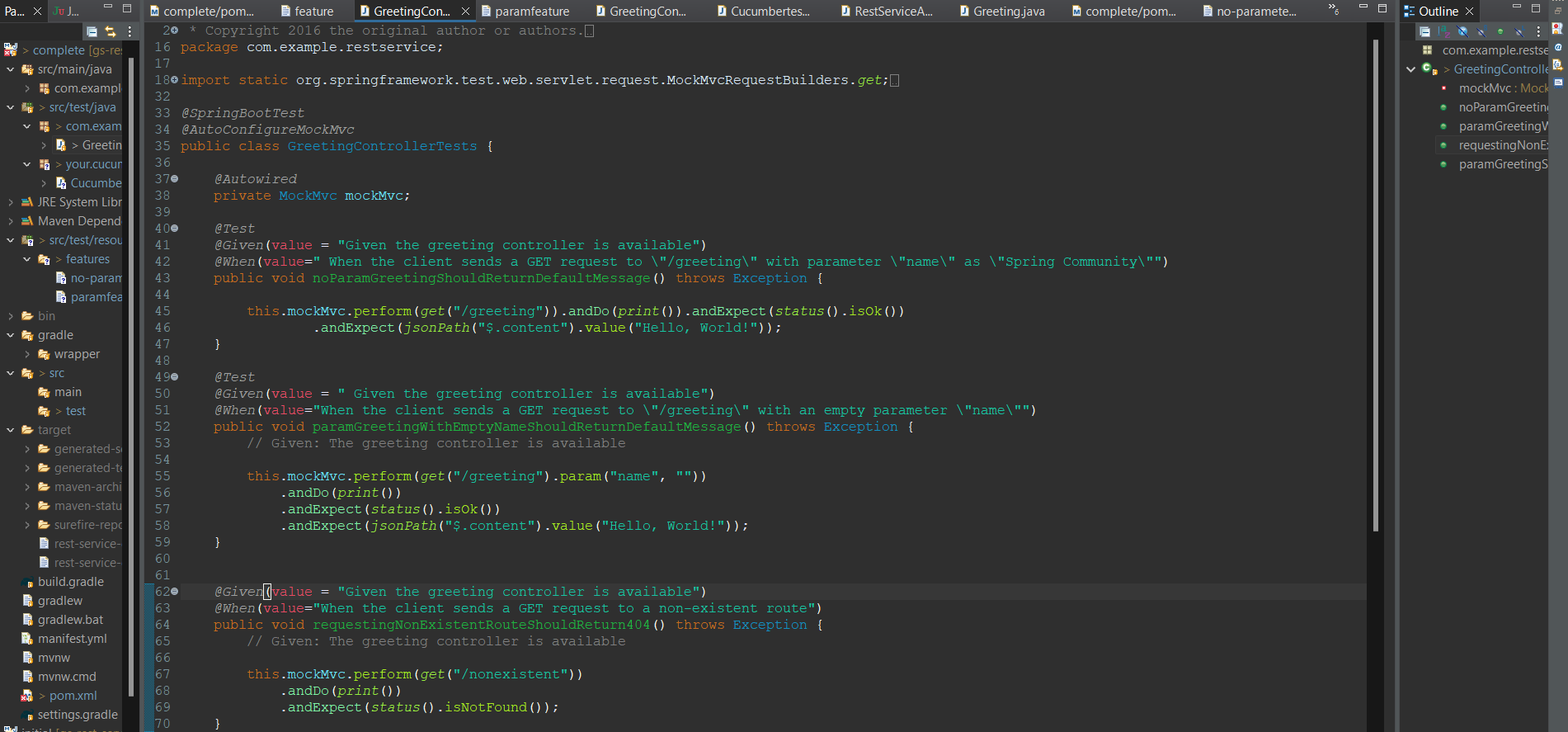


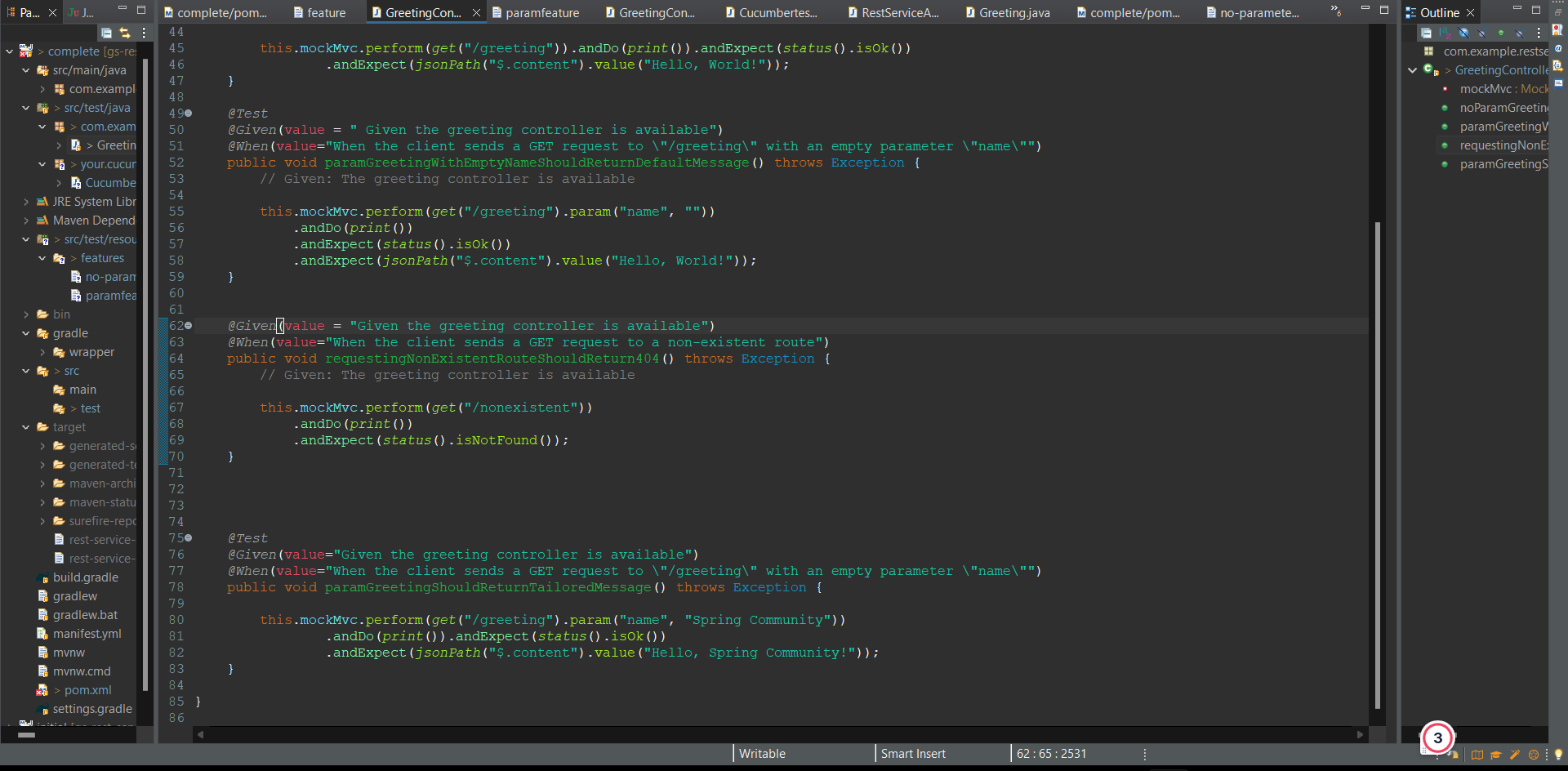
Wrote the test scenarios in Gherkin Language to test the expected behaviour of the application.   
Created two files, each for parameter and no- parameter features.

Step 4: Implement Step Definitions 1. Create step definition classes in the `src/test/java` directory. 2. Implement the steps for each scenario written in the feature files.

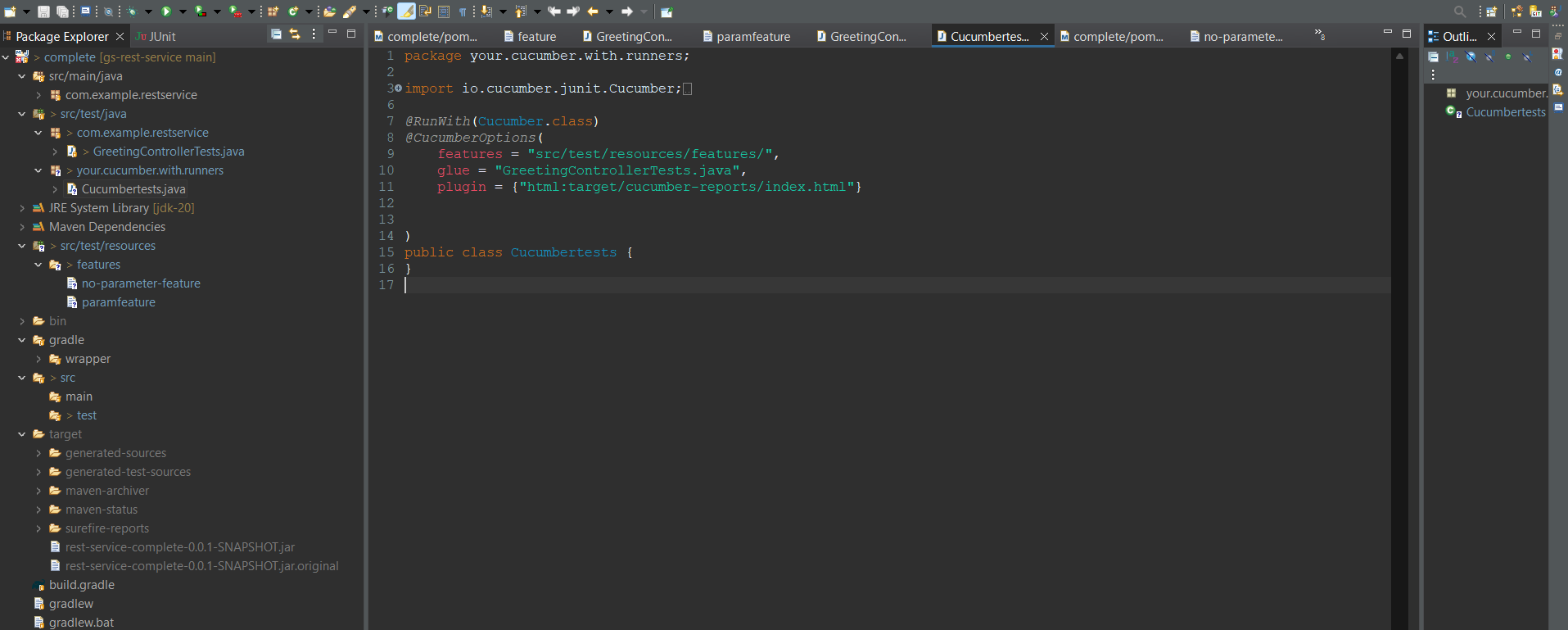


Implemented and added the functions for the test cases in GreetingControllerTests.java to the scenarios written in the Gherkin Language :-





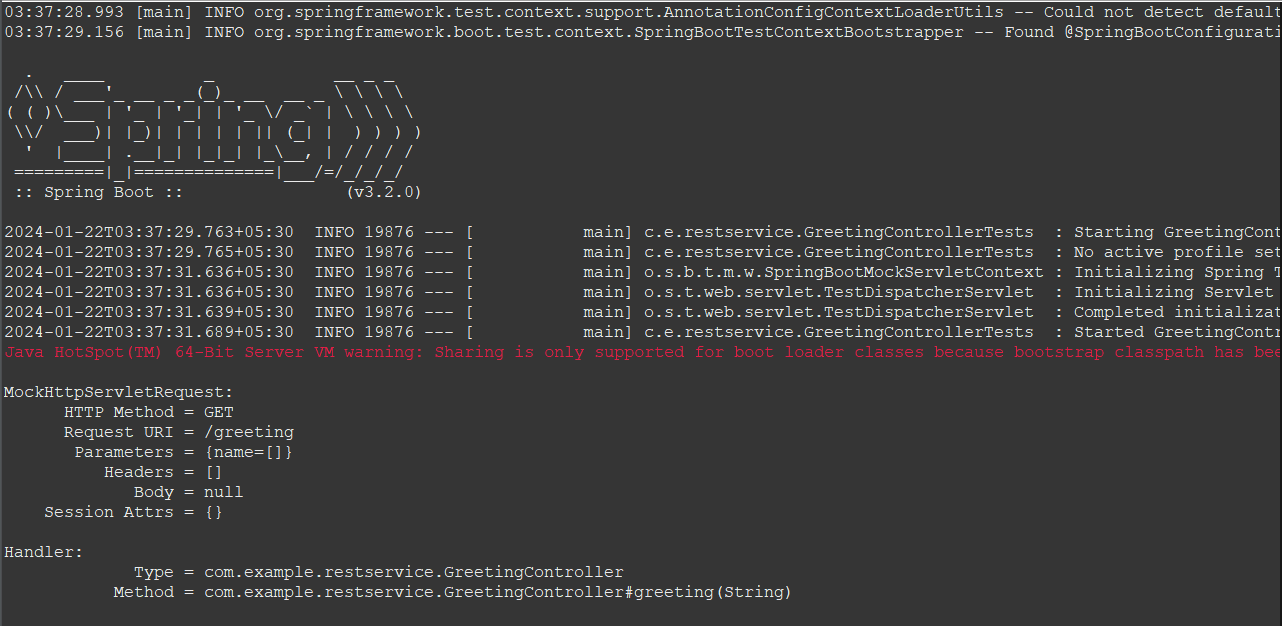
Step 5: Configure Cucumber Options 1. Configure Cucumber options in a runner class or using a `cucumber.properties` file to specify features, glue, plugins, etc.

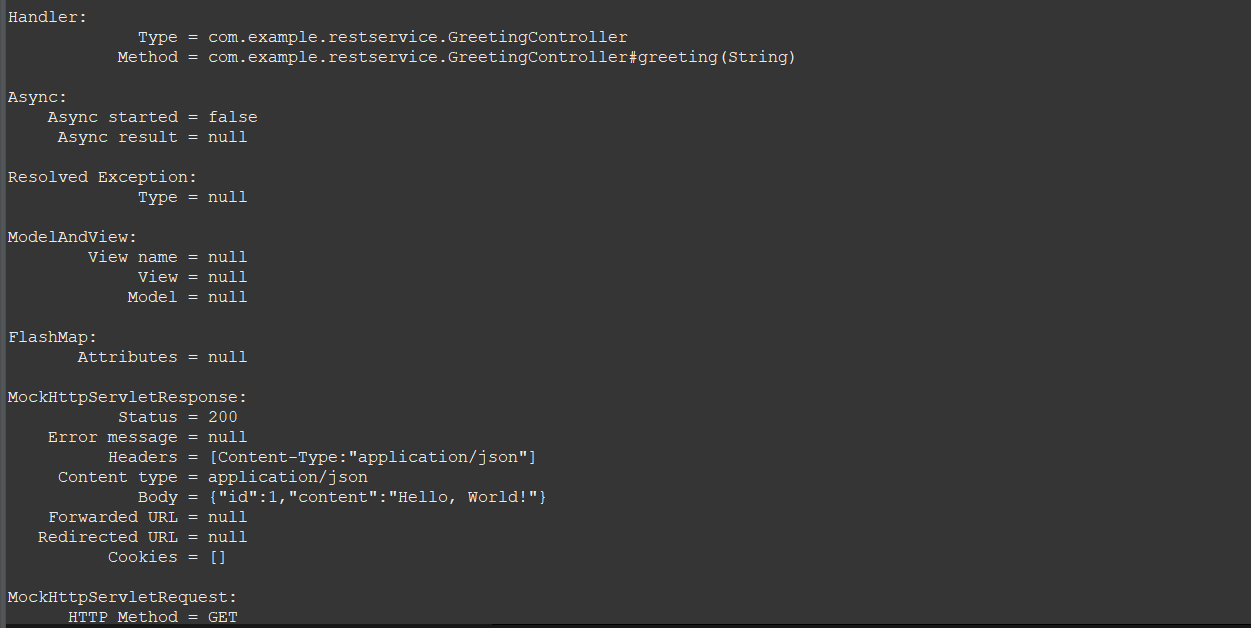


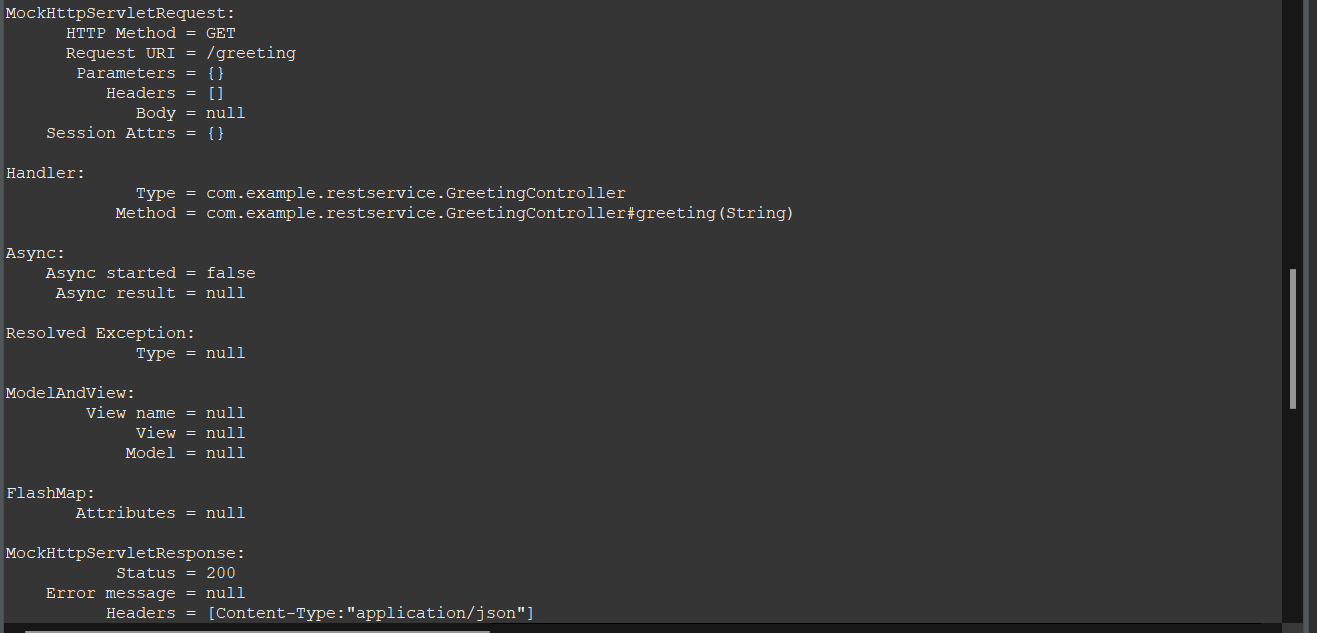
**Created a runner class named Cucumbertests.java and declared the features, glue and plugin.**

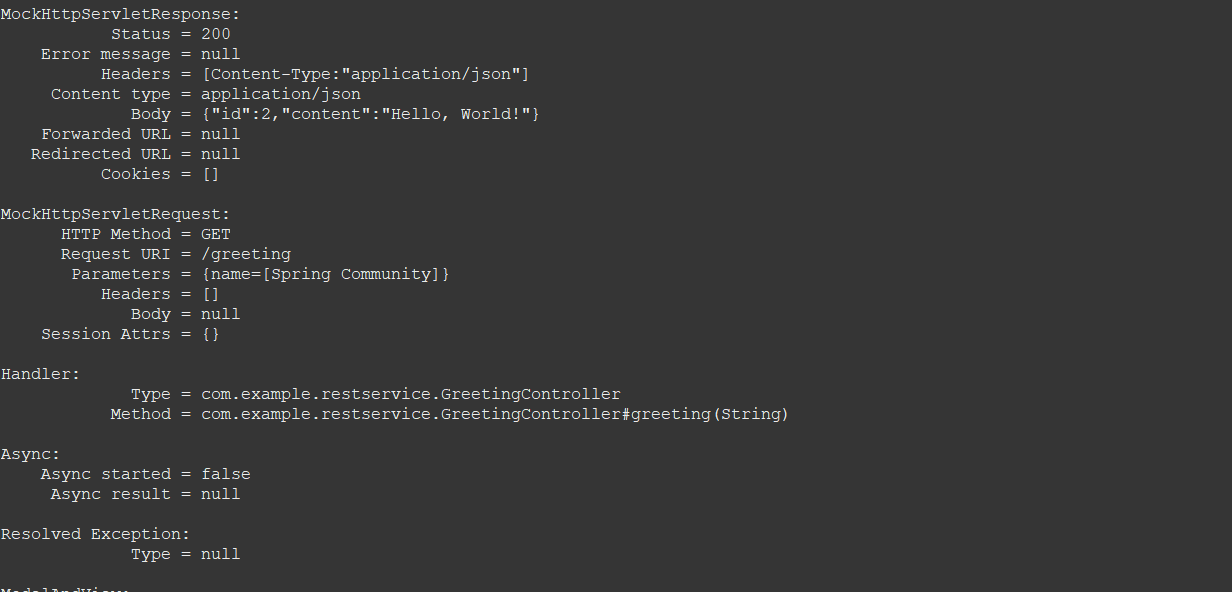
Step 6: Execute Tests and Generating Reports 1. Run the test runner class to execute the Cucumber tests. 2. Configure Cucumber to generate reports. This can be done using Cucumber's built-in report generation feature or by integrating with a reporting tool. Step 7: Review Test Reports 1. Analyze the generated test reports to check the outcomes of the tests. 2. Identify any failures or areas for improvement in the application.

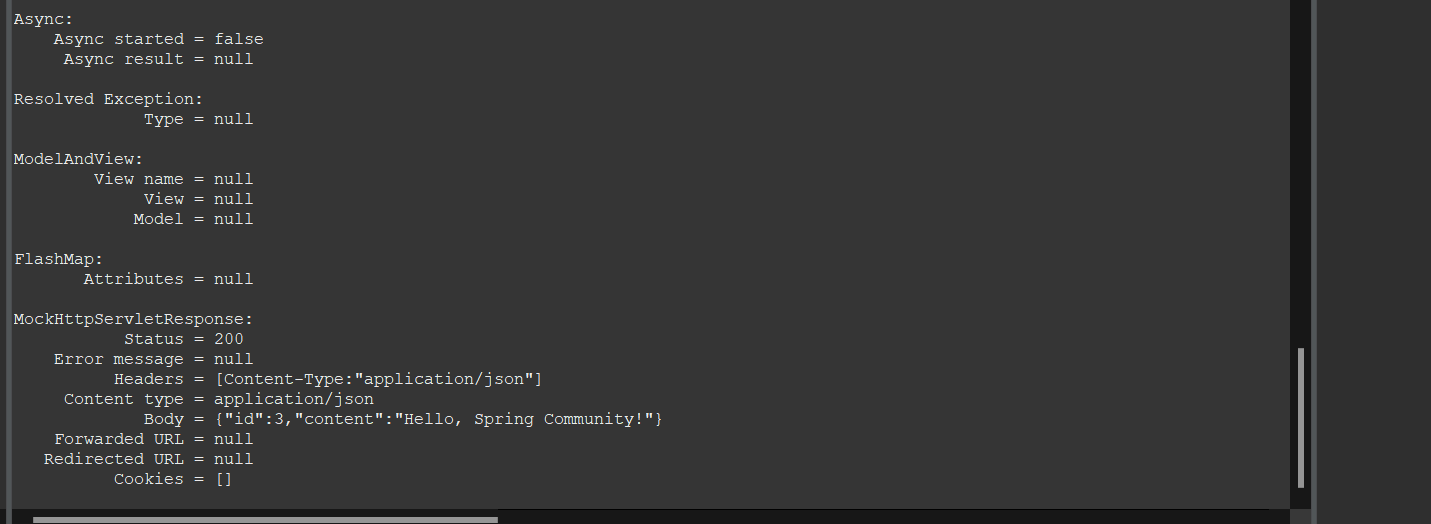
**Ran the java project and the unitjs tests and got these outputs:**

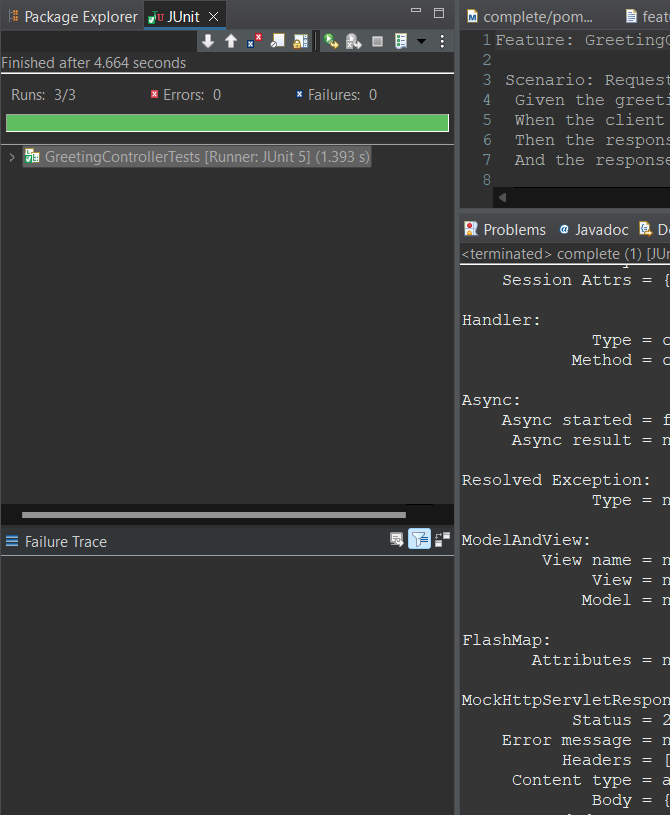












Output on the Local Host.:-

