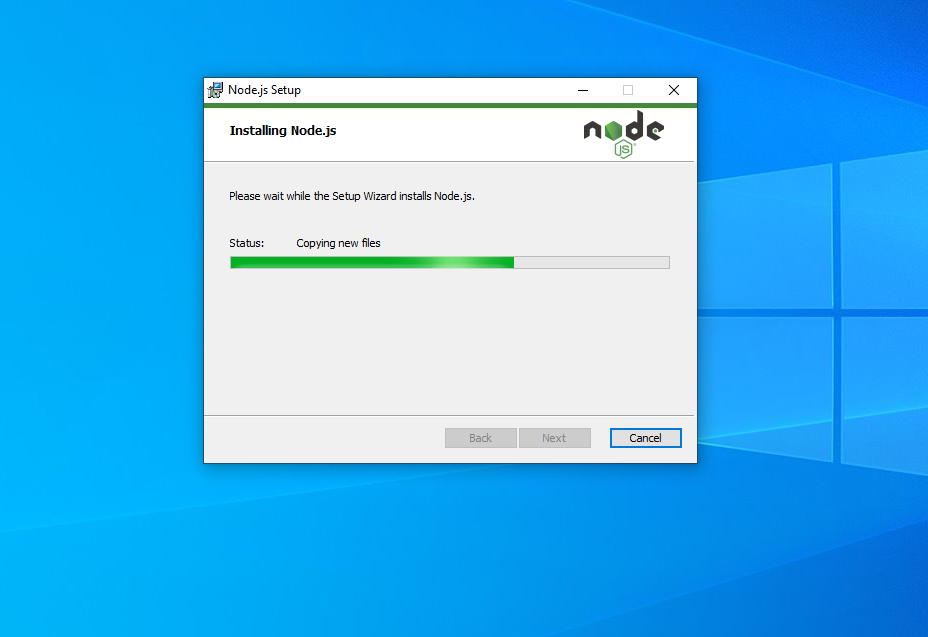
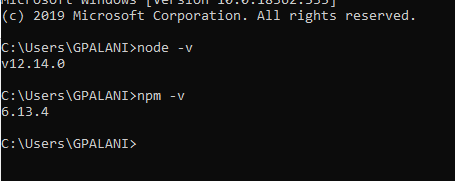
Installing and getting started with Json Server(Mock Server)

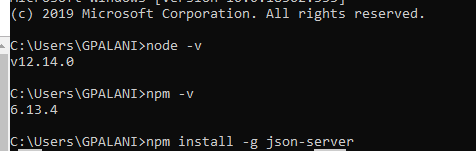
<https://github.com/typicode/json-server>

1. Install npm

<https://nodejs.org/en/download/>

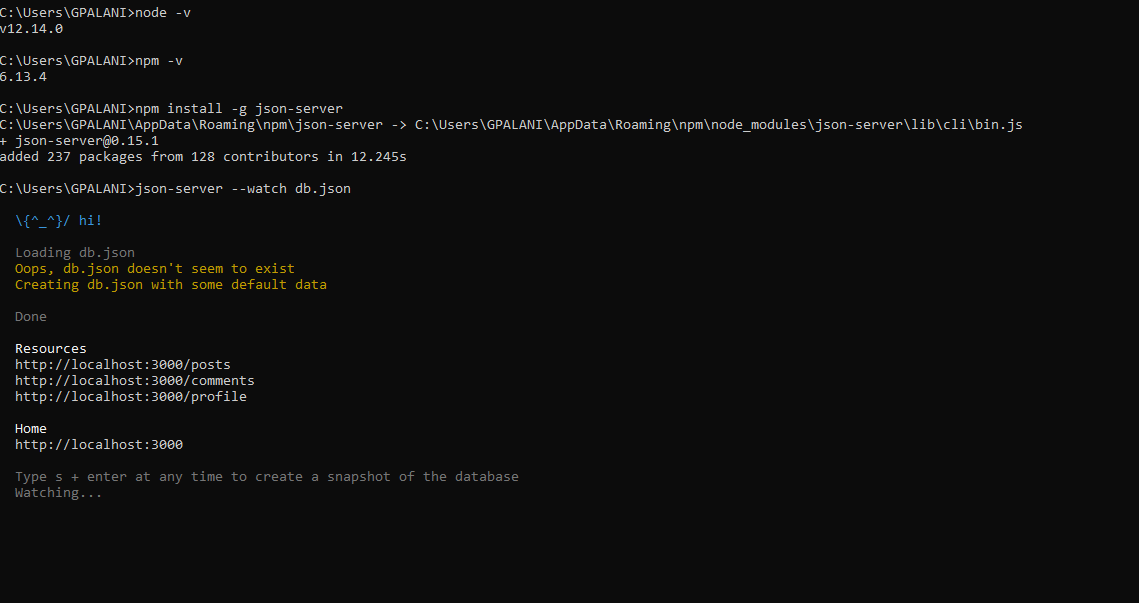


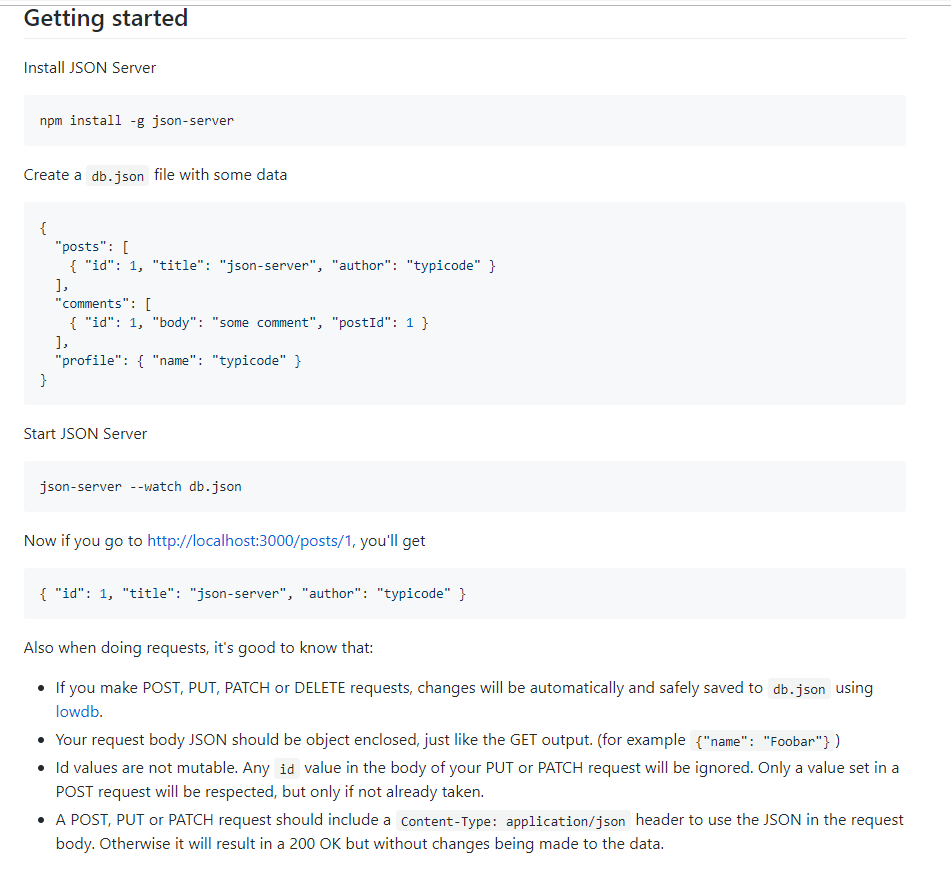




Start JSON Server

json-server --watch db.json

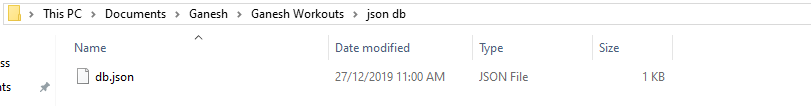






Keep the db.json inside any one of the folder

C:\Users\GPALANI\Documents\Ganesh\Ganesh Workouts\json db



Go to that db.json path

And hit the below command

json-server --watch db.json

Then the following resources will get displayed

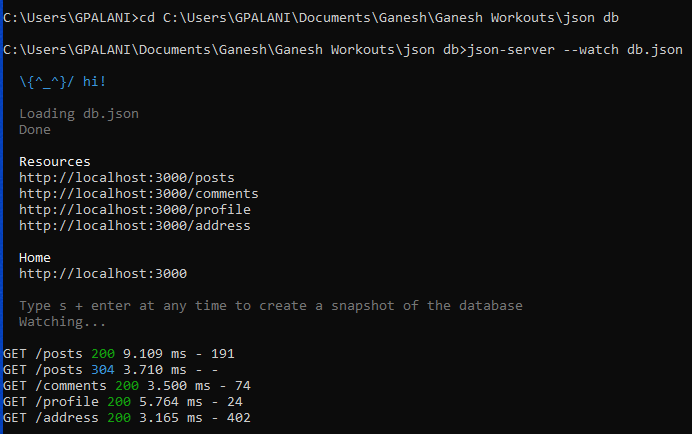
Resources

http://localhost:3000/posts

http://localhost:3000/comments

http://localhost:3000/profile

http://localhost:3000/address

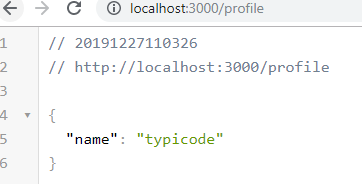


db.json file











Query pram:

Id=1



Id=2:



Id=3:



To limit the result

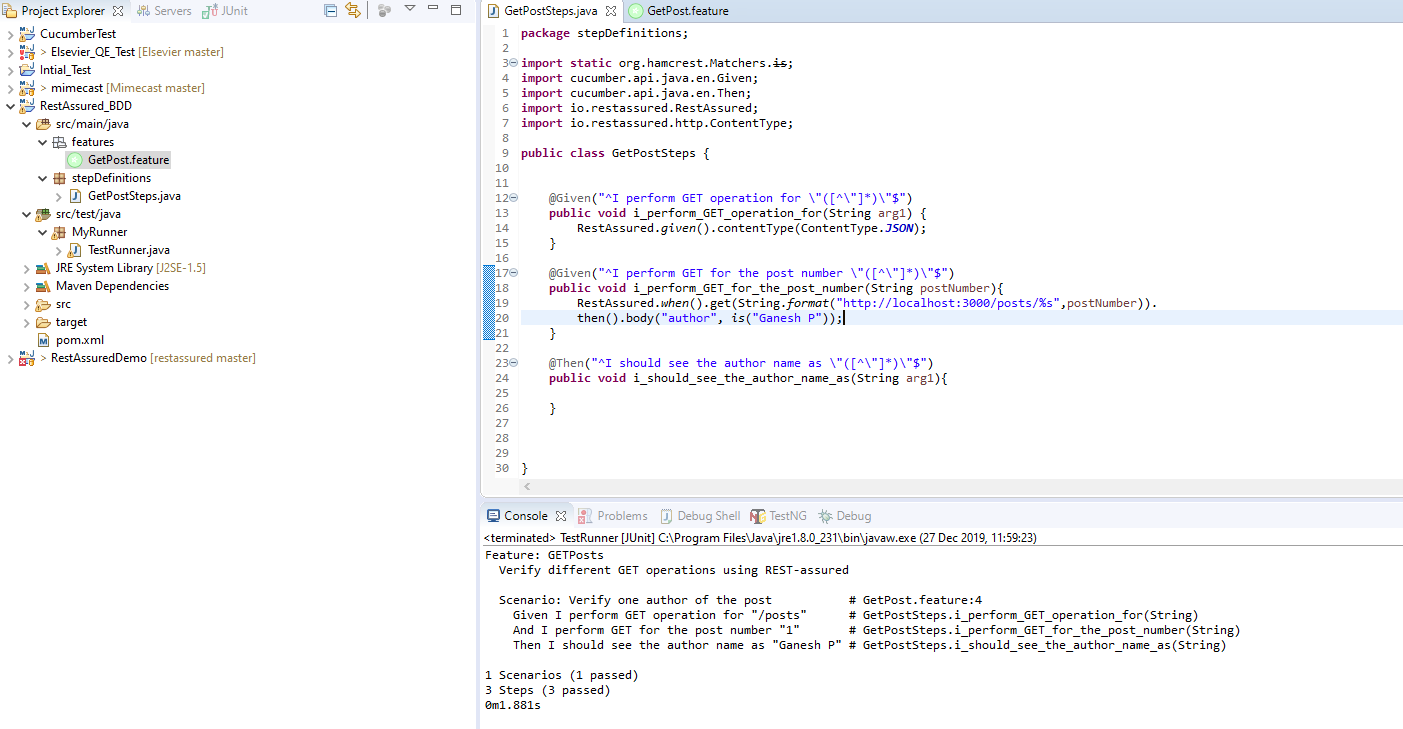
<http://localhost:3000/posts?_limit=2>

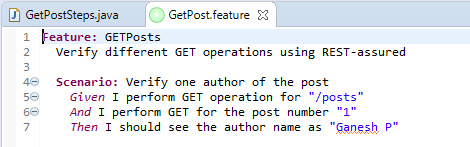


Limit =1

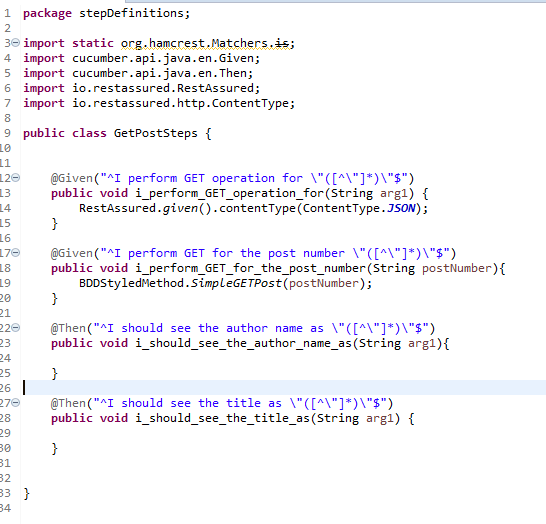




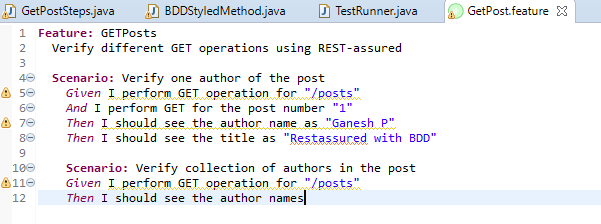


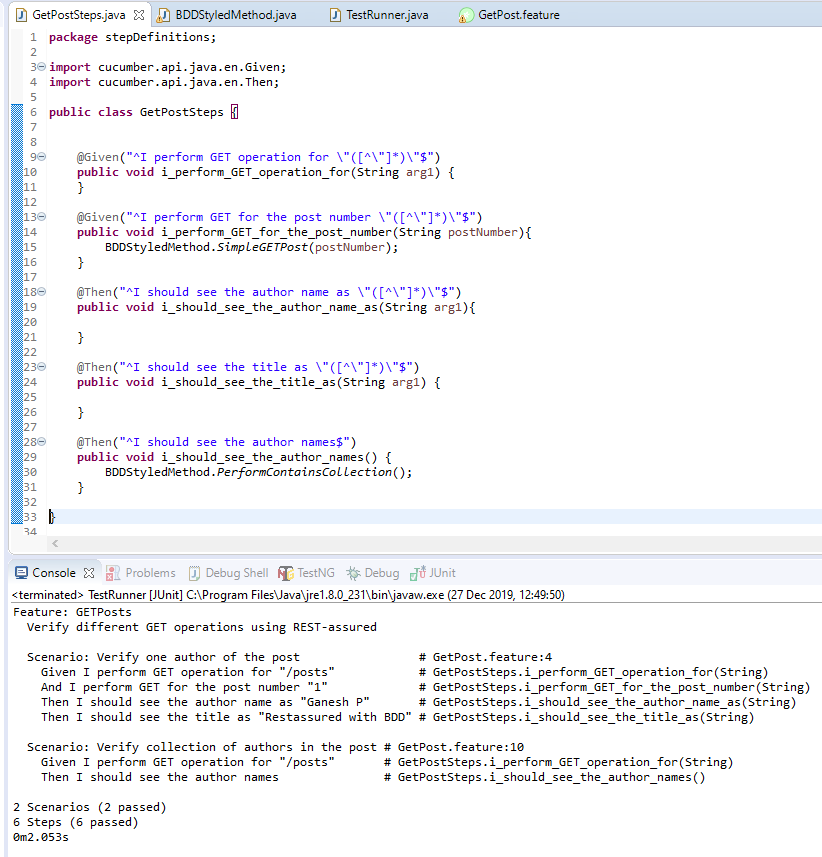


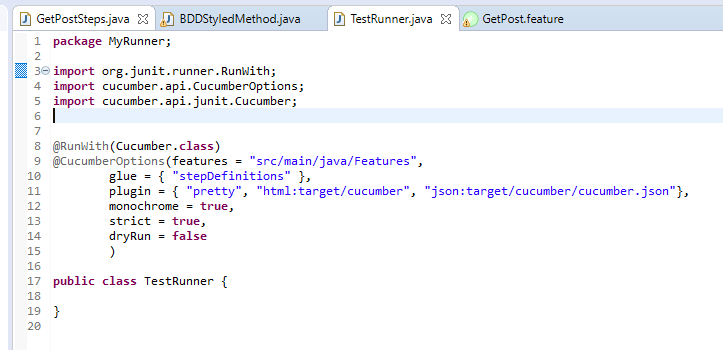




With 2 Scenarios

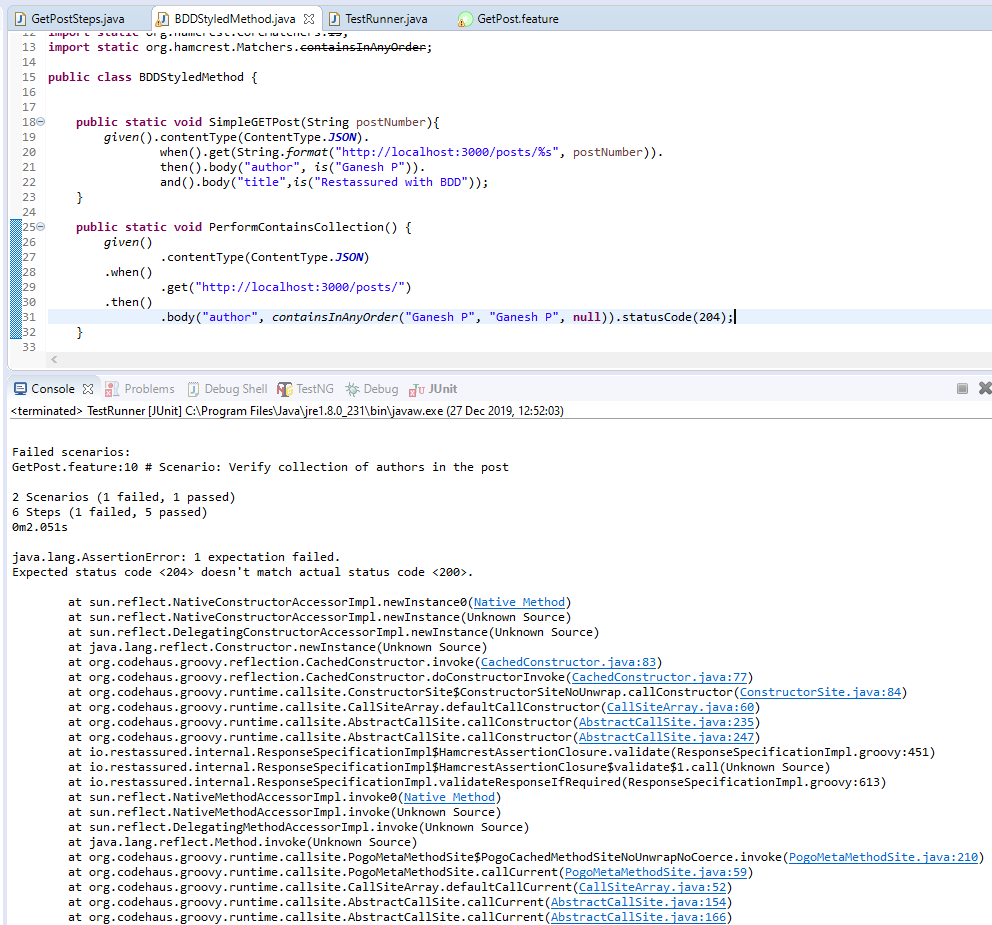




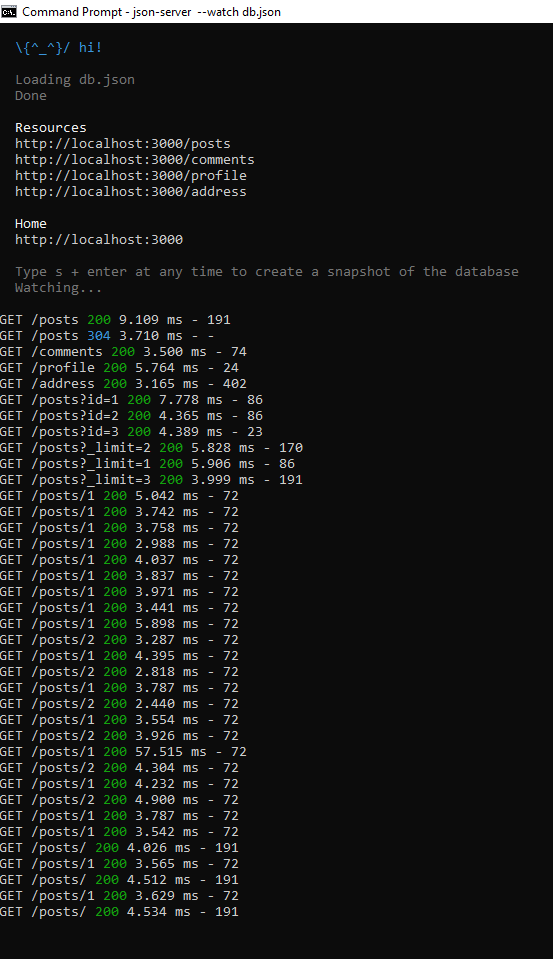




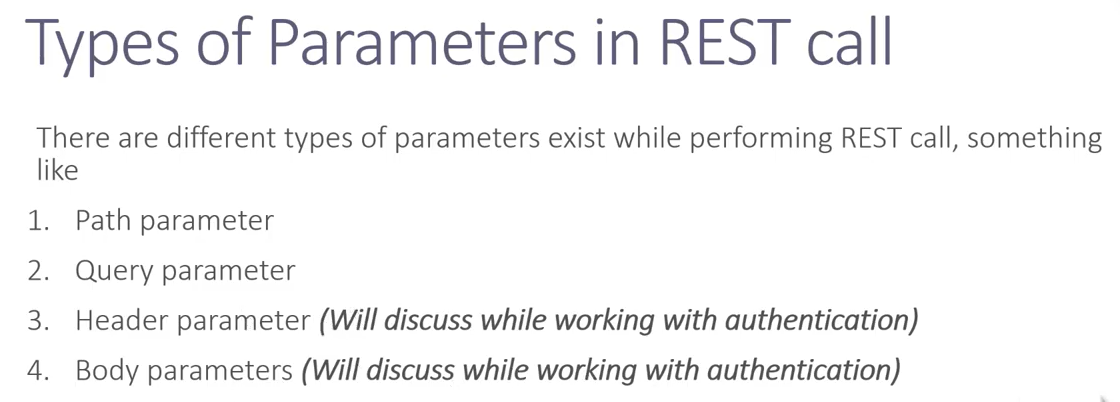
With status code 204 to make the test case fail

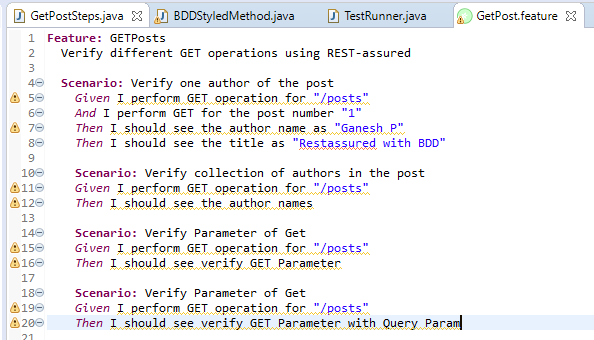


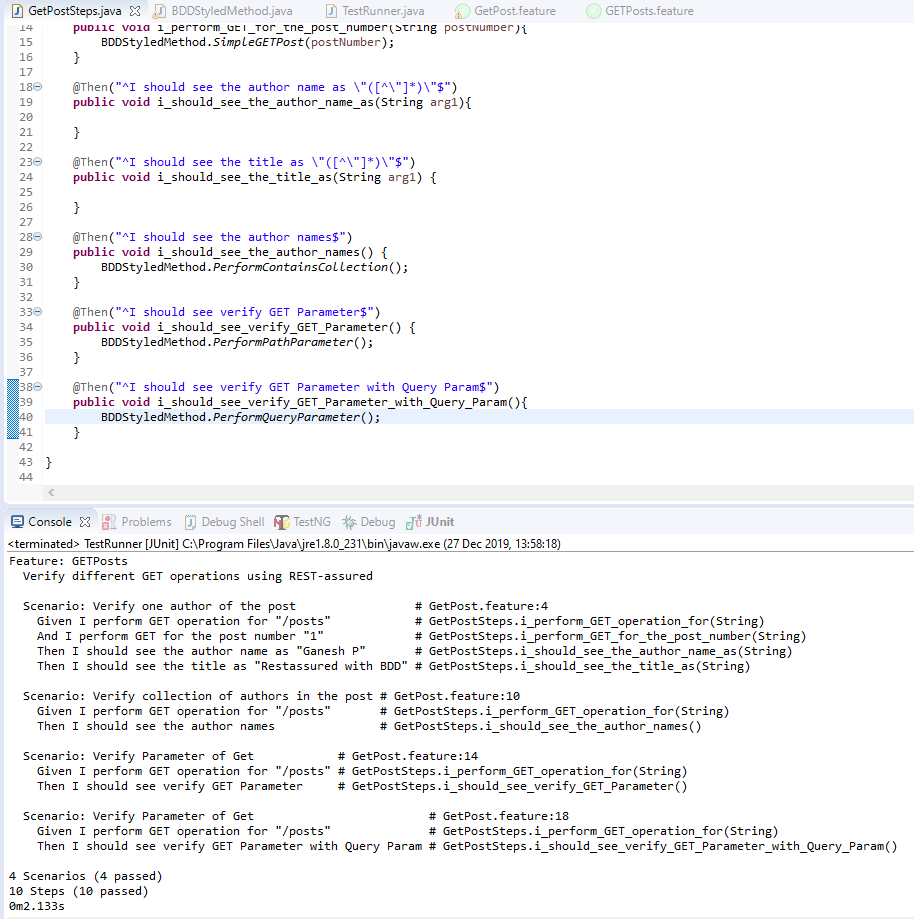
Json Server



Get operations with params:

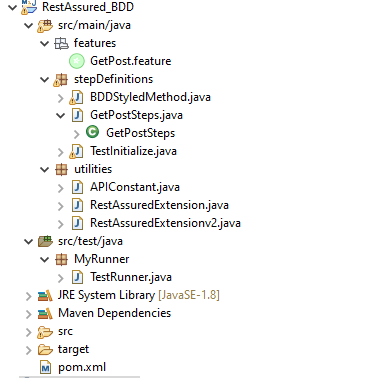




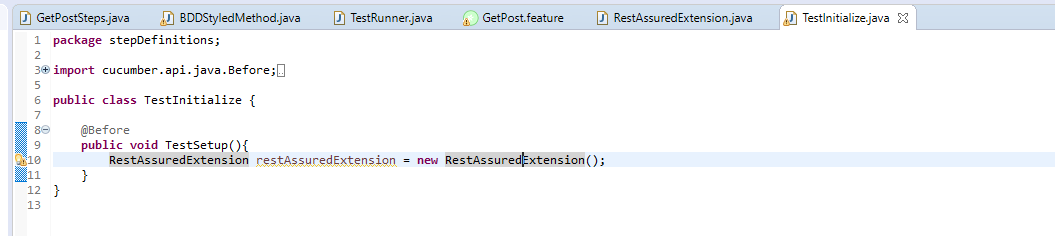




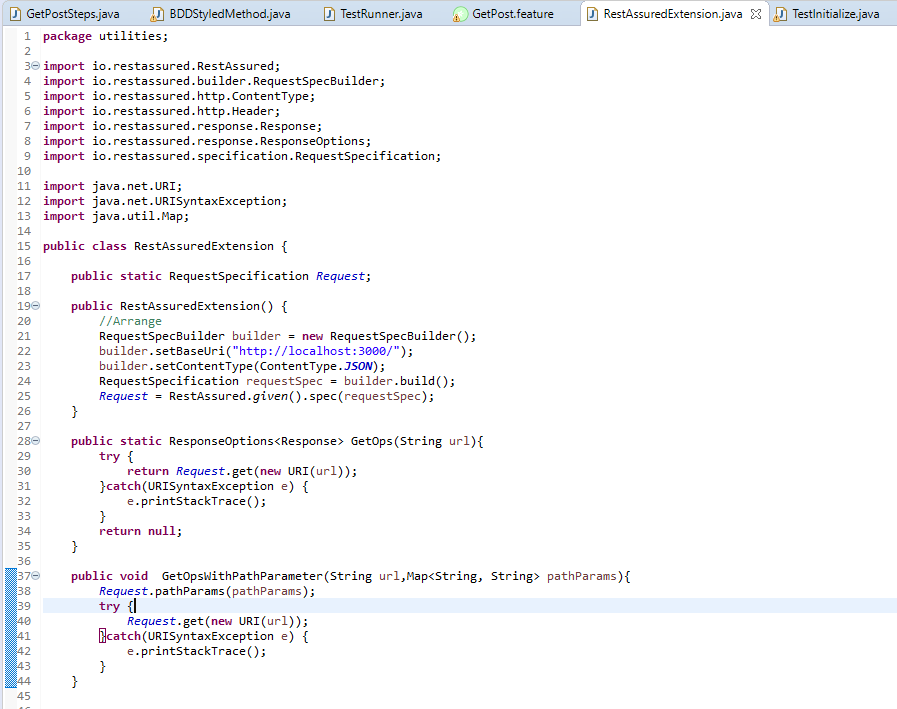
Understanding and building Non-BDD styled reusable REST-assured libraries

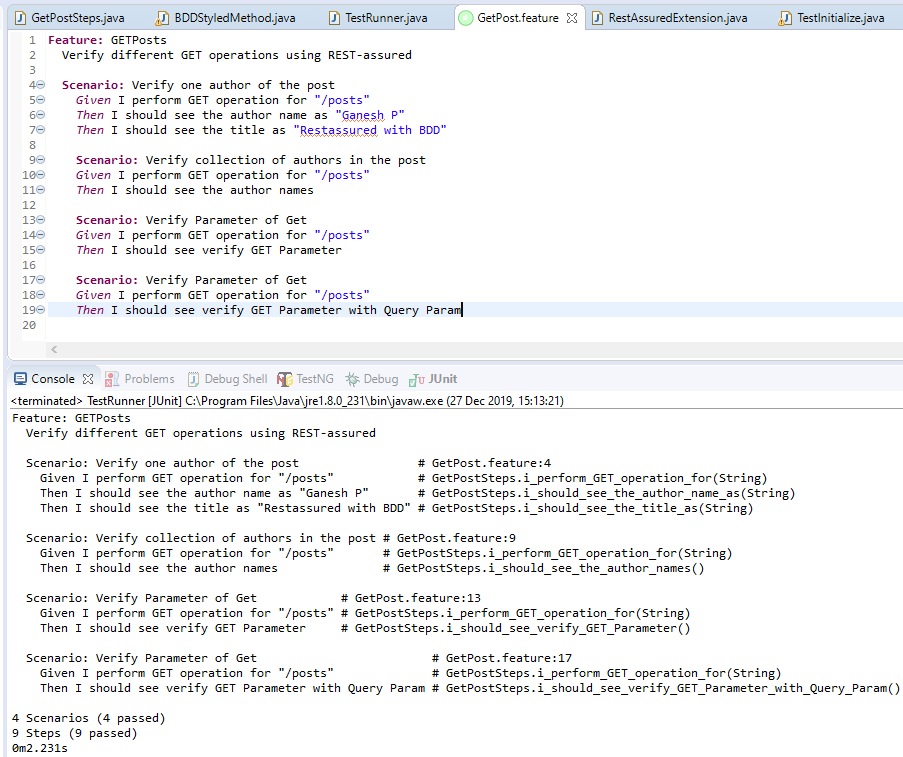


1. Test Initializer – This is like a hooks.

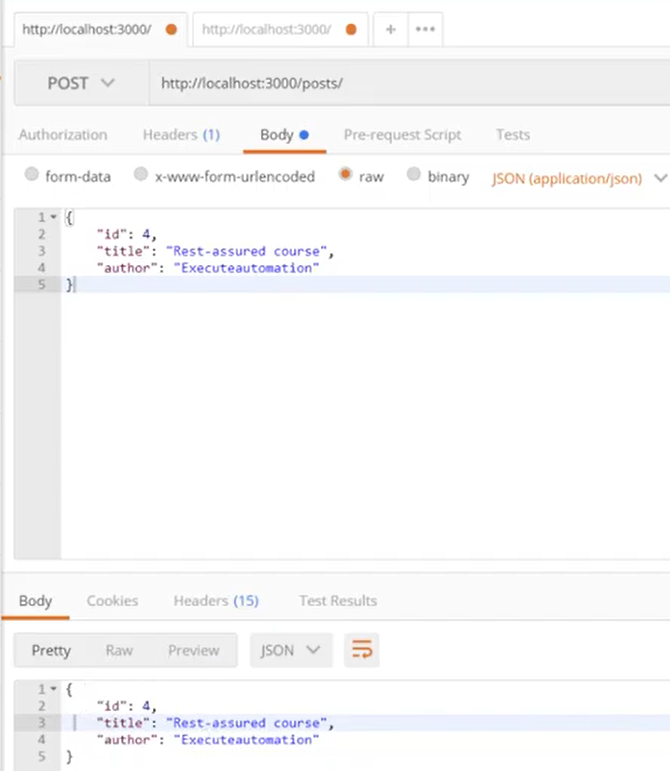


RestAssuredExtension

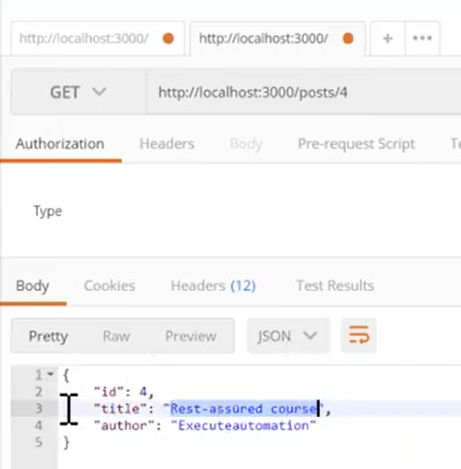


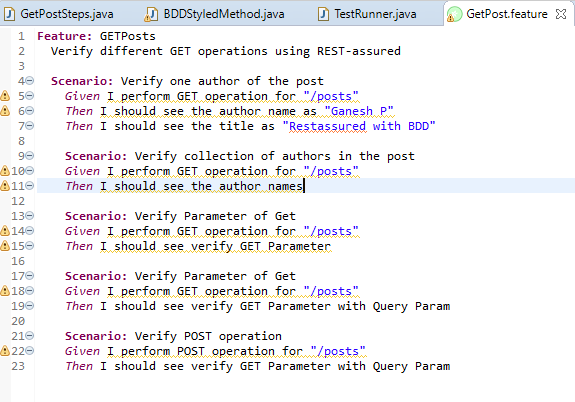


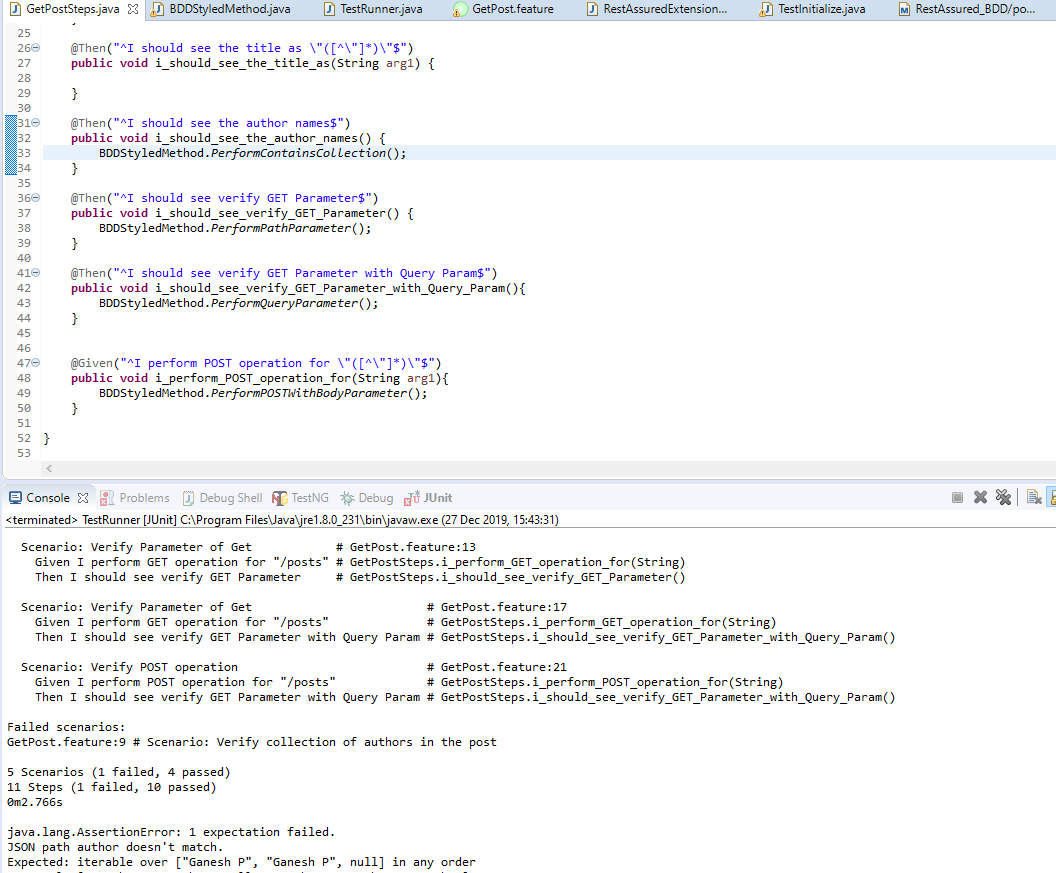
Creating a new Id.



Getting the Id value.

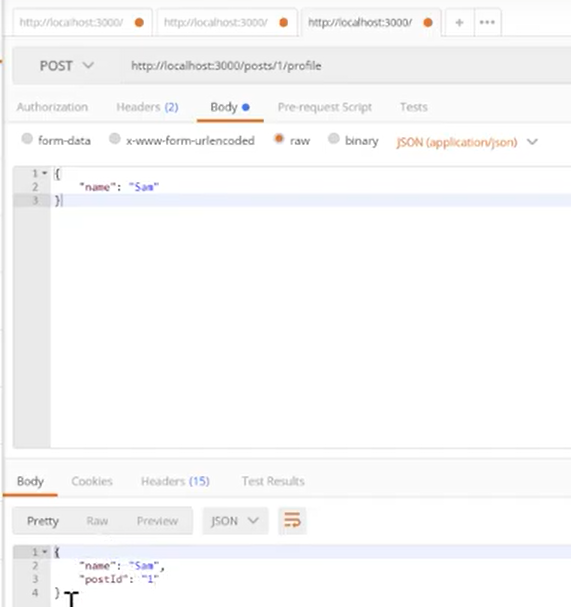


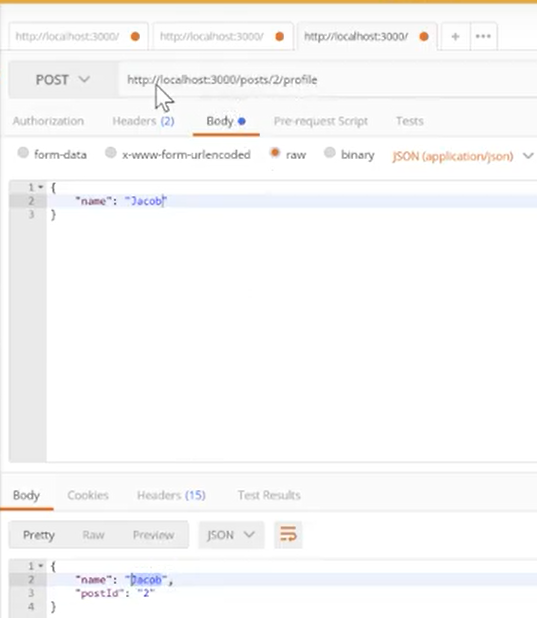




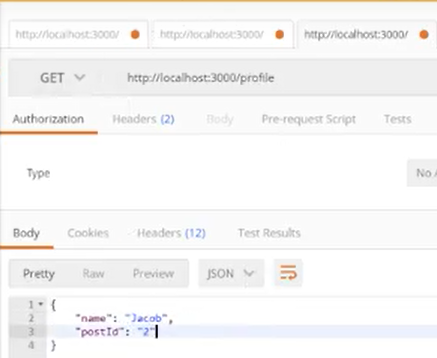


Updating the name:

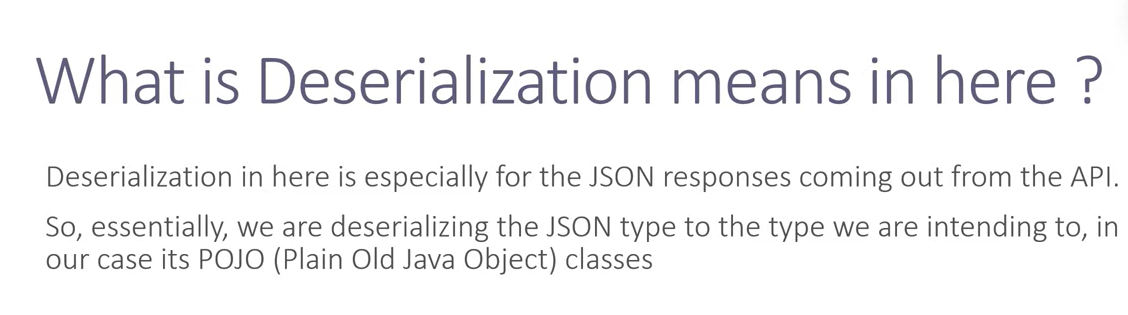


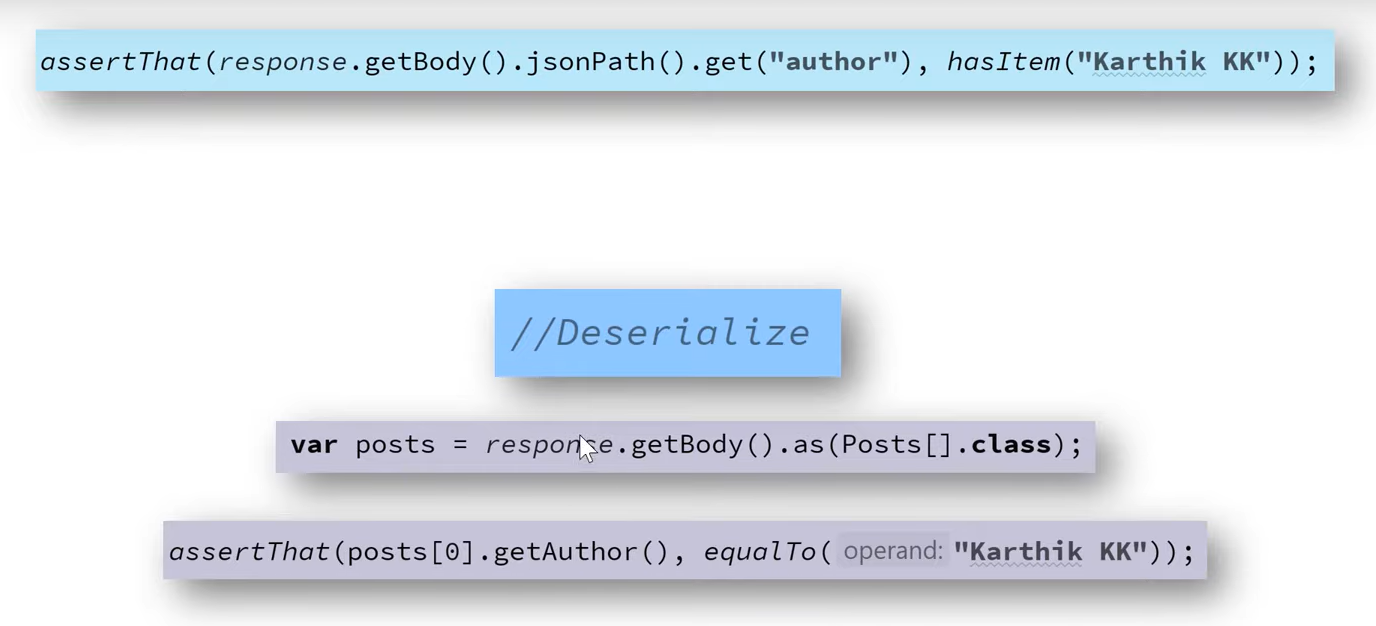


To Get the details:



Deserializing JSON response to POJO class in RestAssured





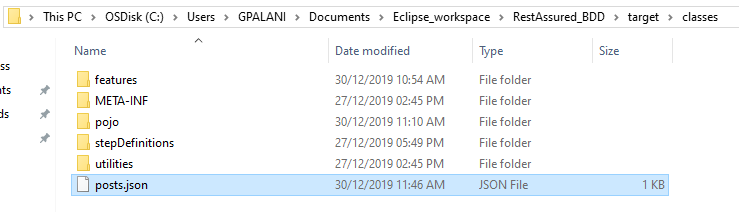




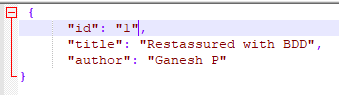


Json Schema validation

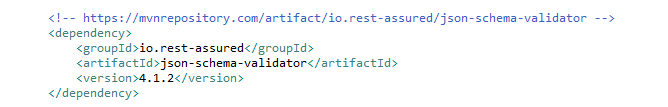
Keep the file inside the target /classes

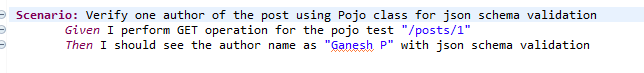


posts.json

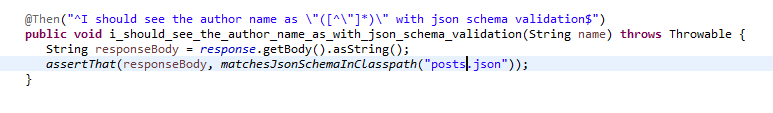


Pom.xml

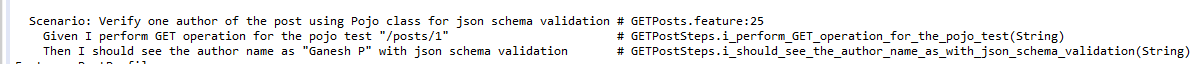




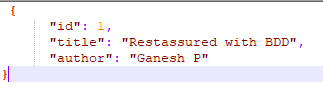
**import** **static** io.restassured.module.jsv.JsonSchemaValidator.*matchesJsonSchemaInClasspath*;

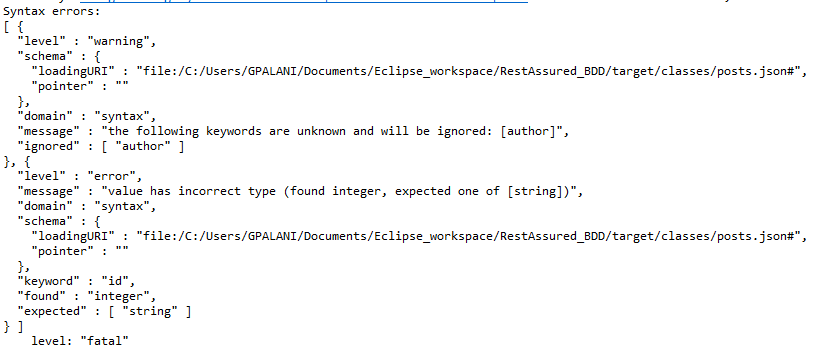


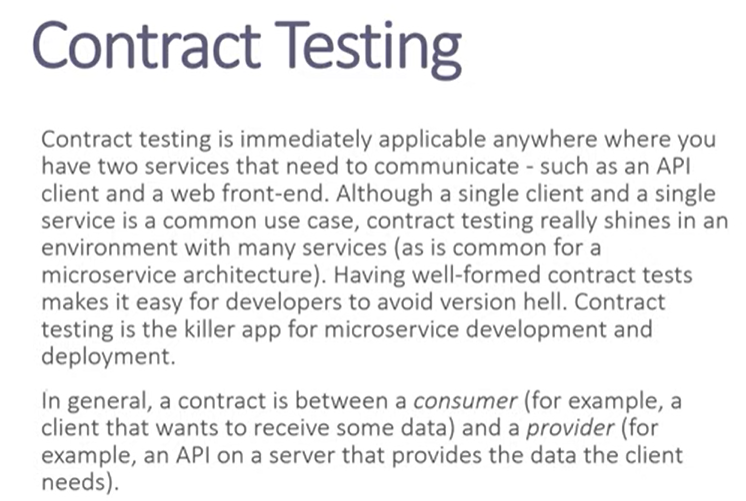
Pass test

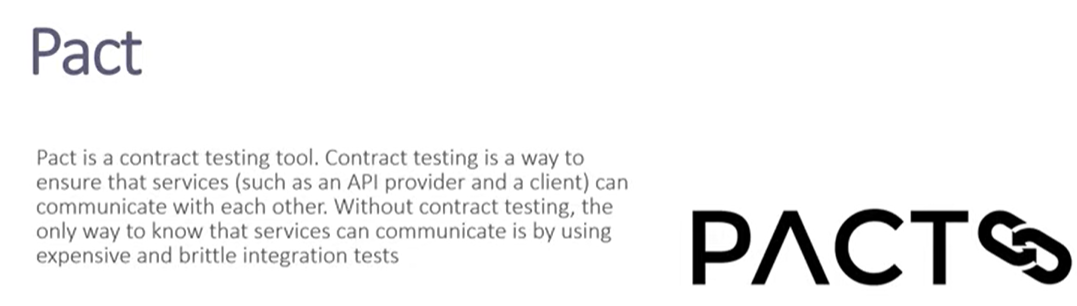


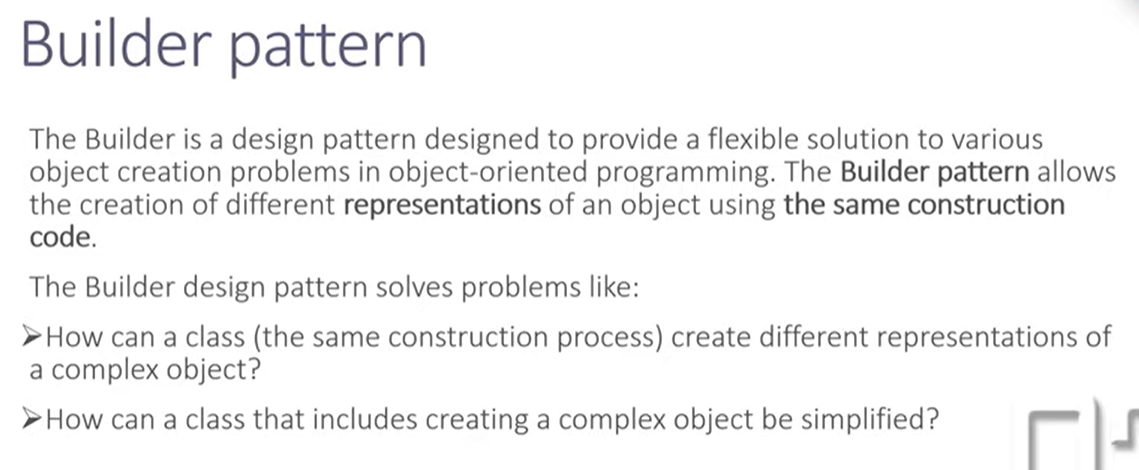
To make the schema validation fail:

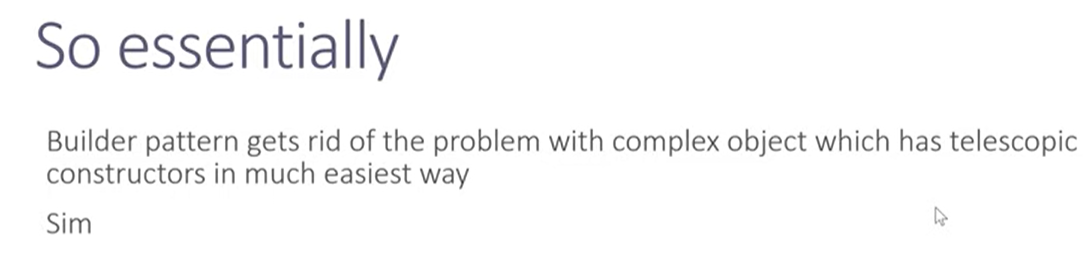


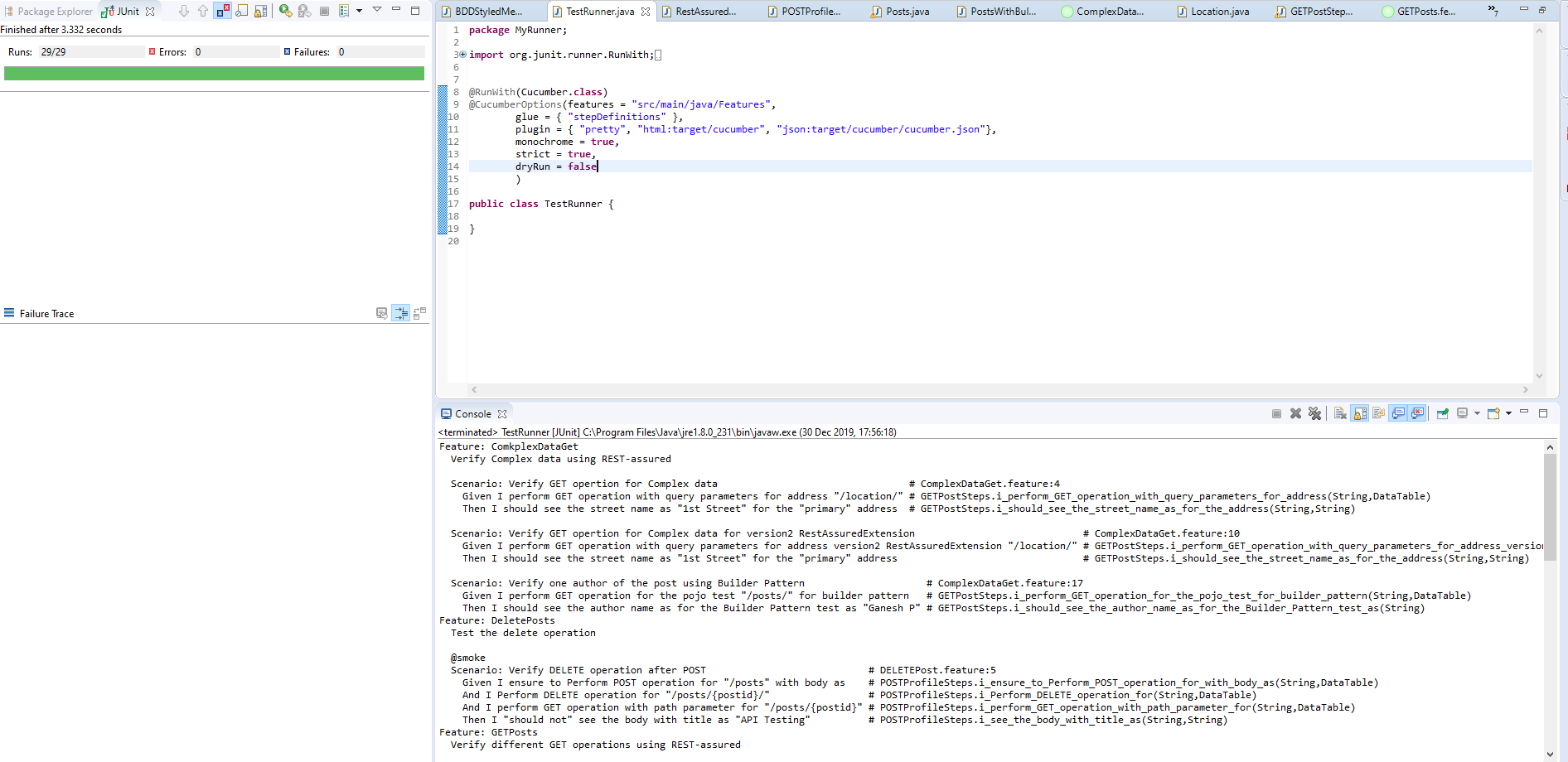




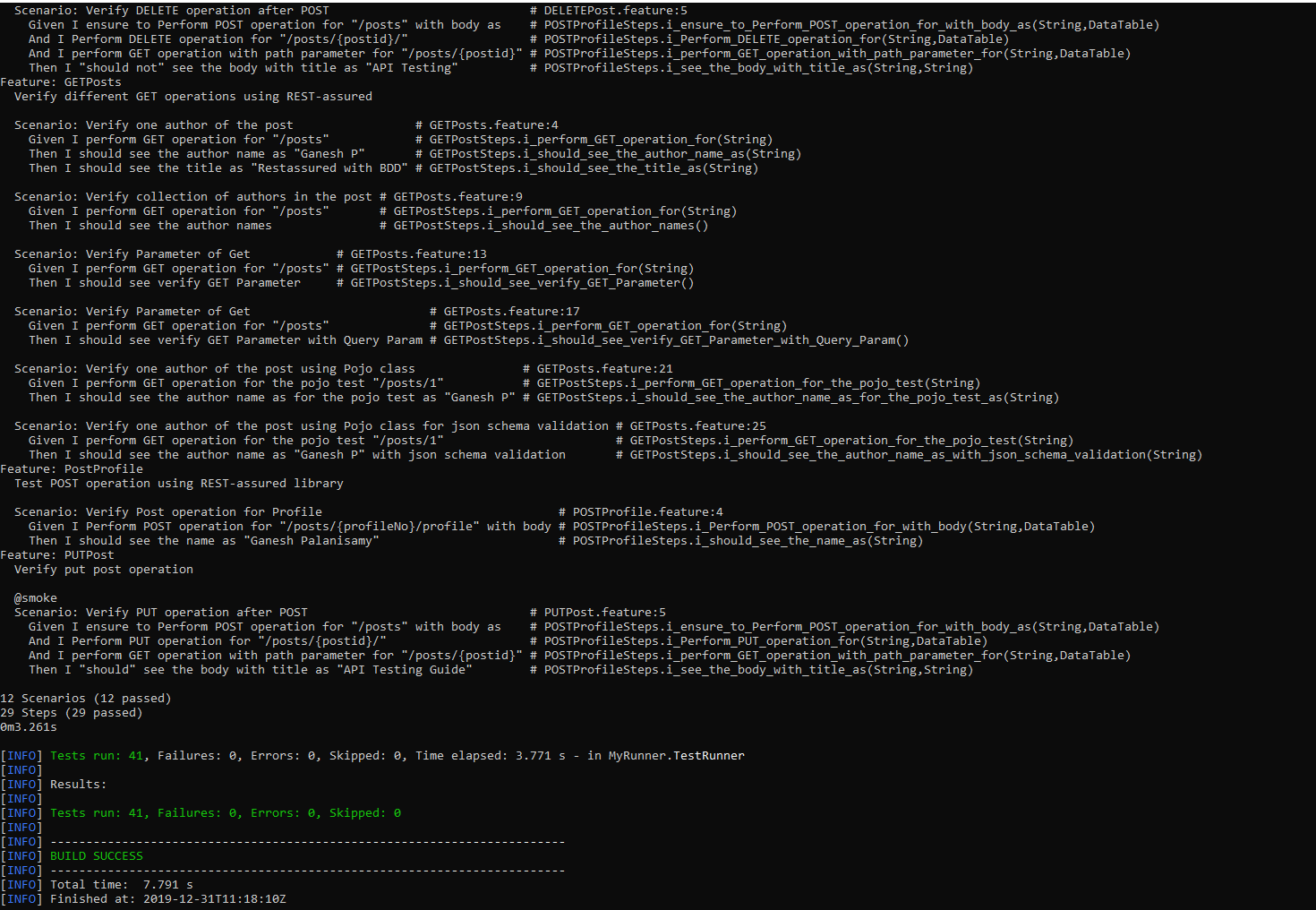




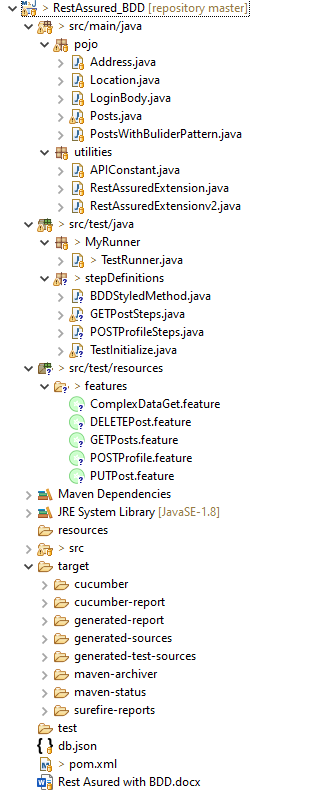




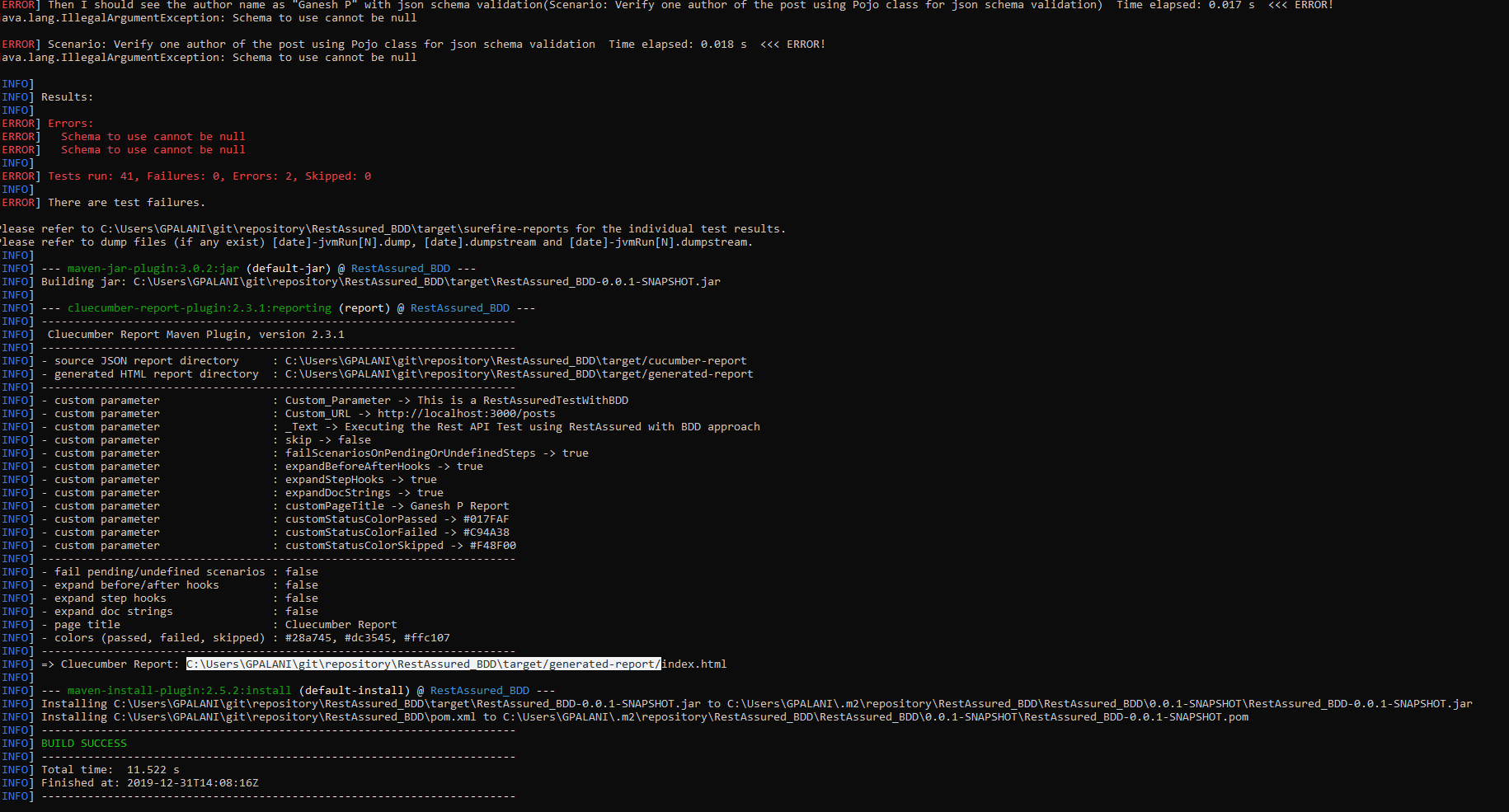
Running from maven

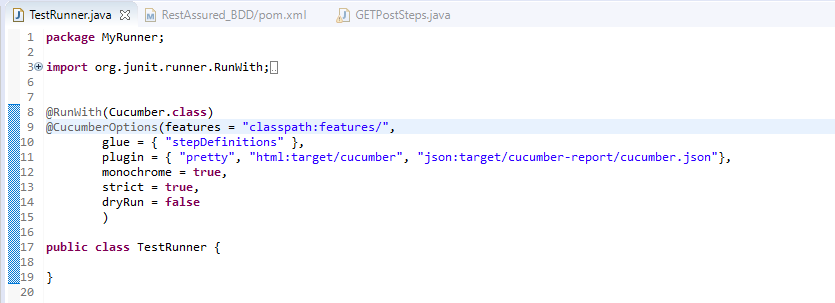


Project Structure



Maven console





Report

<https://github.com/trivago/cluecumber-report-plugin/blob/master/README.md>

<!-- https://mvnrepository.com/artifact/com.trivago.rta/cluecumber-report-plugin -->

<dependency>

<groupId>com.trivago.rta</groupId>

<artifactId>cluecumber-report-plugin</artifactId>

<version>2.3.1</version>

</dependency>

<build>

<plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.0</version>

<configuration>

<testFailureIgnore>true</testFailureIgnore>

<includes>

<include>\*\*/Test\*.java</include>

</includes>

</configuration>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

<plugin>

<artifactId>maven-site-plugin</artifactId>

<version>3.7.1</version>

</plugin>

<plugin>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.0.0</version>

</plugin>

<plugin>

<groupId>com.trivago.rta</groupId>

<artifactId>cluecumber-report-plugin</artifactId>

<version>2.3.1</version>

<executions>

<execution>

<id>report</id>

<phase>post-integration-test</phase>

<goals>

<goal>reporting</goal>

</goals>

</execution>

</executions>

<configuration>

<sourceJsonReportDirectory>${project.build.directory}/cucumber-report</sourceJsonReportDirectory>

<generatedHtmlReportDirectory>${project.build.directory}/generated-report</generatedHtmlReportDirectory>

<logLevel>default</logLevel>

<customParameters>

<Custom\_Parameter>This is a RestAssuredTestWithBDD</Custom\_Parameter>

<Custom\_URL>http://localhost:3000/posts</Custom\_URL>

<\_Text>Executing the Rest API Test using RestAssured with BDD approach</\_Text>

<skip>false</skip>

<failScenariosOnPendingOrUndefinedSteps>true</failScenariosOnPendingOrUndefinedSteps>

<expandBeforeAfterHooks>true</expandBeforeAfterHooks>

<expandStepHooks>true</expandStepHooks>

<expandDocStrings>true</expandDocStrings>

<customPageTitle>Ganesh P Report</customPageTitle>

<customStatusColorPassed>#017FAF</customStatusColorPassed>

<customStatusColorFailed>#C94A38</customStatusColorFailed>

<customStatusColorSkipped>#F48F00</customStatusColorSkipped>

</customParameters>

</configuration>

</plugin>

</plugins>

</build>

Run always with mvn clean install – then only it will generate a report. Don’t run mvn test