# **Hands-On Exercise 02: How to do Spring Boot integration test with JUnit 5**

## **Introduction**

This exercise would help you with setting up Spring boot MVC web application, and how to perform unit tests with JUnit 5 and mocking with Mockito framework. We want to create a unit test for StudentController which is a Rest Controller. StudentController exposes two service methods - one Get and one Post. We will write unit tests for both these service methods.

In the unit test

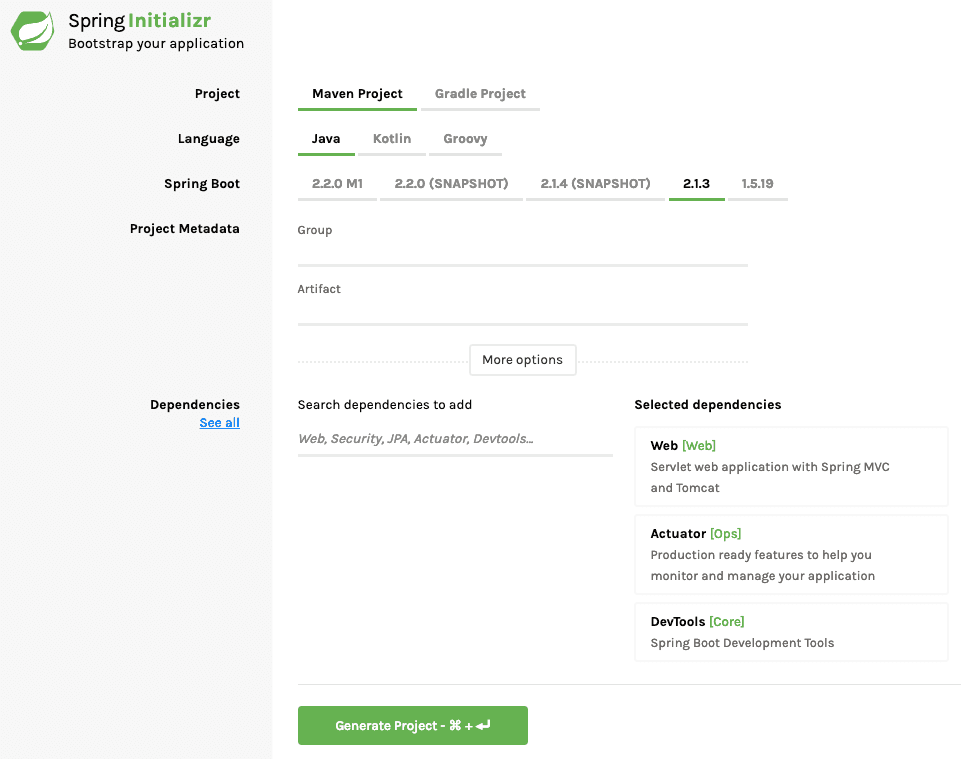
* We will mock out the StudentService using Mockito
* We will use Mock MVC framework to launch only StudentController.

A key part of unit testing is to restrict the scope to a minimum. In this unit test, we want to test only the methods in StudentController.

## **Let’s Get Started**

## **Bootstrap REST Services Application with Spring Initializr**

Let’s get started by creating a REST Service Application using Spring Initializr ([*http://start.spring.io/*](https://start.spring.io/)), it is a great tool to bootstrap your Spring Boot projects.



As shown in the image above, following steps have to be done

* Launch Spring Initializr and choose the following
  + Choose com.example as Group
  + Choose student-services as Artifact
  + Choose following dependencies
    - Web
    - Actuator
    - DevTools
* Click Generate Project.
* Import the project into Eclipse.
* If you want to understand all the files that are part of this project, you can go here.

In pom.xml set the configuration for junit5 just after the junit4 section.

|  |
| --- |
| <!-- junit 5 -->  <dependency>  <groupId>org.junit.jupiter</groupId>  <artifactId>junit-jupiter-engine</artifactId>  <version>${junit-jupiter.version}</version>  <scope>test</scope>  </dependency> |

Once all the dependencies are resolved then get started by create a controller named as ***MainConrollerTest.java,***

|  |
| --- |
| package com.example.controller;  import org.junit.jupiter.api.Test;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.boot.test.context.SpringBootTest;  import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;  import org.springframework.boot.test.web.client.TestRestTemplate;  import org.springframework.boot.web.server.LocalServerPort;  import org.springframework.http.ResponseEntity;  import java.net.URL;  import static org.junit.jupiter.api.Assertions.*assertEquals*;  @SpringBootTest(webEnvironment = WebEnvironment.*RANDOM\_PORT*)  public class MainControllerTest {  // bind the above RANDOM\_PORT  @LocalServerPort  private int port;  @Autowired  private TestRestTemplate restTemplate;  @Test  public void getHello() throws Exception {  ResponseEntity<String> response = restTemplate.getForEntity(new URL("http://localhost:" + port + "/").toString(), String.class);  *assertEquals*("Hello Controller", response.getBody());  }  } |

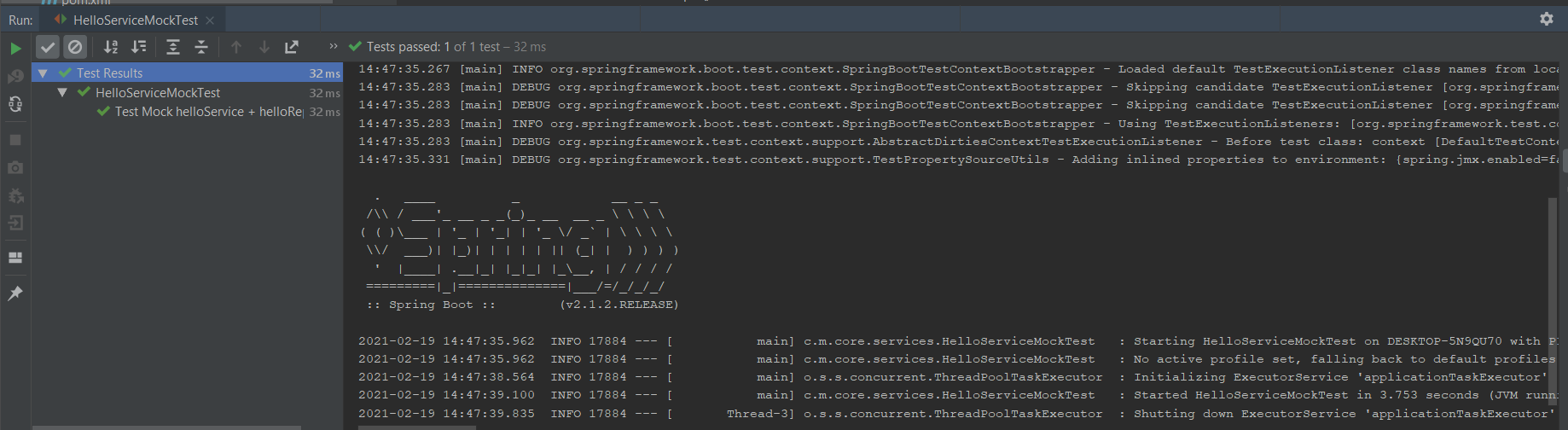
After this, create ***HelloServiceMockTest.java,*** This would create a Mocito service and then we would test it.

|  |
| --- |
| package com.example.services;  import com.training.example.repository.HelloRepository;  import org.junit.jupiter.api.BeforeEach;  import org.junit.jupiter.api.DisplayName;  import org.junit.jupiter.api.Test;  import org.mockito.InjectMocks;  import org.mockito.Mock;  import org.springframework.boot.test.context.SpringBootTest;  import static org.junit.jupiter.api.Assertions.*assertEquals*;  import static org.mockito.Mockito.*when*;  //@ExtendWith(MockitoExtension.class) , need this? still able to run.  @SpringBootTest  public class HelloServiceMockTest {  @Mock  private HelloRepository helloRepository;  //@Spy  @InjectMocks // auto inject helloRepository  private HelloService helloService = new HelloServiceImpl();  @BeforeEach  void setMockOutput() {  //when(helloService.get()).thenReturn("Hello Mockito");  *when*(helloRepository.get()).thenReturn("Hello Mockito From Responsitory");  }  @DisplayName("Test Mock helloService + helloRepository")  @Test  void testGet() {  *assertEquals*("Hello Mockito From Responsitory", helloService.get());  }  } |

Finally create ***HelloServiceTest.java,***

|  |
| --- |
| package com.example.services;  import org.junit.jupiter.api.DisplayName;  import org.junit.jupiter.api.Test;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.boot.test.context.SpringBootTest;  import static org.junit.jupiter.api.Assertions.*assertEquals*;  //https://docs.spring.io/spring-boot/docs/2.1.2.RELEASE/reference/htmlsingle/#boot-features-testing-spring-boot-applications  @SpringBootTest  public class HelloServiceTest {  @Autowired  HelloService helloService;  @DisplayName("Test Spring @Autowired Integration")  @Test  void testGet() {  *assertEquals*("Hello JUnit 5", helloService.get());  }  } |

Once all this is done. Click on the **Run** button, present on the top of the page. If it shows output like this,



Then we can say.

**Voila!!** We have successfully completed this Exercise.