Spring Boot JUnit with Mockito

**1. application-test.properties**

server.port=9999

##DATASOURCE##

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/sboot

spring.datasource.username=root

spring.datasource.password=root

##HIBERNATE##

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL55Dialect

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

2. Model class

**package** org.sathyatech.model;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** Customer {

@Id

@GeneratedValue

**private** Integer cid;

**private** String cname;

**private** String ctype;

**public** Customer() {

**super**();

}

**public** Integer getCid() {

**return** cid;

}

**public** **void** setCid(Integer cid) {

**this**.cid = cid;

}

**public** String getCname() {

**return** cname;

}

**public** **void** setCname(String cname) {

**this**.cname = cname;

}

**public** String getCtype() {

**return** ctype;

}

**public** **void** setCtype(String ctype) {

**this**.ctype = ctype;

}

@Override

**public** String toString() {

**return** "Customer [cid=" + cid + ", cname=" + cname + ", ctype=" + ctype + "]";

}

}

3. Repository

**package** org.sathyatech.repo;

**import** org.sathyatech.model.Customer;

**import** org.springframework.data.jpa.repository.JpaRepository;

**public** **interface** CustomerRepo

**extends** JpaRepository<Customer, Integer>

{

}

4. Service Interface and class

**package** org.sathyatech.service;

**import** org.sathyatech.model.Customer;

**public** **interface** ICustomerService {

**public** Integer saveCustomer(Customer c);

**public** Customer getOneCustomer(Integer id);

}

**package** org.sathyatech.service.impl;

**import** java.util.Optional;

**import** org.sathyatech.model.Customer;

**import** org.sathyatech.repo.CustomerRepo;

**import** org.sathyatech.service.ICustomerService;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

@Service

**public** **class** CustomerServiceImpl **implements** ICustomerService {

@Autowired

**private** CustomerRepo repo;

**public** Integer saveCustomer(Customer c) {

**return** repo.save(c).getCid();

}

**public** Customer getOneCustomer(Integer id) {

Optional<Customer> opt=repo.findById(id);

**if**(opt.isPresent()) {

**return** opt.get();

}

**return** **null**;

}

}

5. RestController

**package** org.sathyatech.rest;

**import** org.sathyatech.model.Customer;

**import** org.sathyatech.service.ICustomerService;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.PostMapping;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/customer")

**public** **class** CustomerRestController {

@Autowired

**private** ICustomerService service;

@PostMapping("/save")

**public** ResponseEntity<String> saveCustomer(

@RequestBody Customer cust)

{

ResponseEntity<String> resp=**null**;

**try** {

Integer id=service.saveCustomer(cust);

resp=**new** ResponseEntity<String>("saved with:"+id, HttpStatus.***OK***);

} **catch** (Exception e) {

resp=**new** ResponseEntity<String>(e.getMessage(), HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

**return** resp;

}

@GetMapping("/view/{id}")

**public** ResponseEntity<Customer> getOneCustomer(

@PathVariable Integer id)

{

ResponseEntity<Customer> resp=**null**;

Customer cust=service.getOneCustomer(id);

**if**(cust==**null**) {

resp=**new** ResponseEntity<Customer>(HttpStatus.***NO\_CONTENT***);

}**else** {

resp=**new** ResponseEntity<Customer>(cust, HttpStatus.***OK***);

}

**return** resp;

}

}

6. UnitTest code

**package** org.sathyatech;

**import** **static** org.junit.Assert.*assertEquals*;

**import** **static** org.junit.Assert.*assertNotNull*;

**import** **static** org.junit.Assert.*fail*;

**import** org.junit.Test;

**import** org.junit.runner.RunWith;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

**import** org.springframework.boot.test.context.SpringBootTest;

**import** org.springframework.boot.test.context.SpringBootTest.WebEnvironment;

**import** org.springframework.http.MediaType;

**import** org.springframework.mock.web.MockHttpServletResponse;

**import** org.springframework.test.context.TestPropertySource;

**import** org.springframework.test.context.junit4.SpringRunner;

**import** org.springframework.test.web.servlet.MockMvc;

**import** org.springframework.test.web.servlet.MvcResult;

**import** org.springframework.test.web.servlet.request.MockHttpServletRequestBuilder;

**import** org.springframework.test.web.servlet.request.MockMvcRequestBuilders;

@RunWith(SpringRunner.**class**)

@SpringBootTest(webEnvironment=WebEnvironment.***MOCK***)

//@WebMvcTest // RestController

@AutoConfigureMockMvc

@TestPropertySource("classpath:application-test.properties")

**public** **class** SpringBootCurdUnitTestApplicationTests {

@Autowired

**private** MockMvc mockMvc;

@Test

**public** **void** testCustSave() **throws** Exception {

//1. Create Req object

MockHttpServletRequestBuilder request=

MockMvcRequestBuilders

.*post*("/customer/save")

.contentType(MediaType.***APPLICATION\_JSON***)

.content("{\"cname\":\"AJAY\",\"ctype\":\"ABCD\"}");

//2. perform call

MvcResult result=mockMvc

.perform(request)

.andReturn();

//3. Read Response object

MockHttpServletResponse response=

result.getResponse();

//4. call assert method

*assertNotNull*(response.getContentAsString());

//assertEquals("saved with:2", response.getContentAsString());

**if**(!response.getContentAsString()

.contains("saved with")) {

*fail*("Customer Not saved");

}

*assertEquals*(200, response.getStatus());

*assertEquals*("text/plain;charset=UTF-8", response.getContentType());

}

@Test

**public** **void** testCustView() **throws** Exception {

//1. create req

MockHttpServletRequestBuilder

request=

MockMvcRequestBuilders

.*get*("/customer/view/101");

//2. perform call

MvcResult result=mockMvc

.perform(request)

.andReturn();

//3. read respo

MockHttpServletResponse response=

result.getResponse();

//4. assert calls

*assertEquals*(200, response.getStatus());

*assertNotNull*(response.getContentAsString());

*assertEquals*("application/json;charset=UTF-8", response.getContentType());

}

}

pom.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.1.8.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>org.sathyatech</groupId>

<artifactId>SpringBootCurdUnitTest</artifactId>

<version>1.0</version>

<name>SpringBootCurdUnitTest</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>