### **Spring Boot Fundamentals Assignment Solutions**

## **Assignment: 1 Spring Boot Project**

```
package com.bharath.core.dao;
     import org.springframework.stereotype.Repository;
     @Repository
     public class PaymentDAOImpl implements PaymentDAO {
}
     CoreApplicationTests.java
     package com.bharath.core;
     import com.bharath.core.service.PaymentServiceImpl;
     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.*;
     @SpringBootTest
```

```
class CoreApplicationTests {
     @Autowired
     PaymentServiceImpl paymentService;
     @Test
     void testDependencyInjection() {
     assertNotNull(paymentService);
}
     @Test
     void testSumCalculation() {
     assertEquals(36, paymentService.calculateSum(9, 27));
}
}
     package com.bharath.core.dao;
     import org.springframework.stereotype.Repository;
     @Repository
```

```
public class PaymentDAOImpl implements PaymentDAO {
}
     package com.bharath.core;
     import com.bharath.core.service.PaymentServiceImpl;
     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.*;
     @SpringBootTest
     class CoreApplicationTests {
     @Autowired
     PaymentServiceImpl paymentService;
     @Test
     void testDependencyInjection() {
```

```
assertNotNull(paymentService);

}

@Test

void testSumCalculation() {

assertEquals(36, paymentService.calculateSum(9, 27));
}
```

## **Assignment: 2 Spring Data JPA**

```
package com.example.demo.entity;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.SequenceGenerator;
```

```
import javax.persistence.Table;
     @Entity
     @Table(name = "student")
     public class Student {
     @ld
     @SequenceGenerator(name = "student_sequence", initialValue
= 1)
     @GeneratedValue(strategy =
GenerationType.SEQUENCE,generator = "student_sequence")
     private Long id;
     private String name;
     private Integer testscore;
     public Student(){
}
     public Long getId() {
     return id;
```

```
}
     public void setId(Long id) {
     this.id = id;
}
     public String getName() {
     return name;
}
     public void setName(String name) {
     this.name = name;
}
     public Integer getTestscore() {
     return testscore;
}
     public void setTestscore(Integer testscore) {
     this.testscore = testscore;
```

```
}
}
     package com.example.demo.repo;
     import org.springframework.data.repository.CrudRepository;
     import org.springframework.stereotype.Repository;
     import com.example.demo.entity.Student;
     @Repository
     public interface StudentRepo extends CrudRepository<Student,
Long> {
}
     package com.example.demo.service;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.stereotype.Service;
     import com.example.demo.entity.Student;
```

```
import com.example.demo.repo.StudentRepo;
     @Service
     public class StudentService {
     @Autowired
     StudentRepo repo;
     public Student save(Student student) {
     return repo.save(student);
}
     public Student update(Student stu) {
     return repo.save(stu);
}
     public String delete(Long stuld) {
     repo.deleteByld(stuld);
     return "User " + stuld + " deleted..";
}
```

```
public Optional<Student> get(Long id) {
     return repo.findById(id);
}
}
     package com.example.demo;
     import org.junit.jupiter.api.Test;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.boot.test.context.SpringBootTest;
     import com.example.demo.entity.Student;
     import com.example.demo.service.StudentService;
     @SpringBootTest
     class Demo1ApplicationTests {
     @Autowired
     StudentService service;
     @Test
```

```
void curdOperations() {
     Student stu = new Student();
     stu.setName("nitin");
     stu.setTestscore(20);
     System.out.println(service.save(stu));
     Student savedStu = service.save(stu);
     System.out.println(savedStu.getId() + ": "+ savedStu.getName()
+ ": "+ savedStu.getTestscore());
     Student stuFromDB = service.get(11).get();
     System.out.println(stuFromDB.getId() + ": "+
stuFromDB.getName() + " : "+ stuFromDB.getTestscore());
     Student updateStu = stuFromDB;
     updateStu.setName("Nitin-1");
     Student save = service.save(updateStu);
     System.out.println(save.getId() + ": "+ save.getName() + ": "+
save.getTestscore());
}
```

```
}
     package com.example.demo.entity;
     import javax.persistence.Entity;
     import javax.persistence.GeneratedValue;
     import javax.persistence.GenerationType;
     import javax.persistence.ld;
     import javax.persistence.SequenceGenerator;
     import javax.persistence.Table;
     @Entity
     @Table(name = "student")
     public class Student {
     @Id
     @SequenceGenerator(name = "student_sequence", initialValue
= 1)
     @GeneratedValue(strategy =
GenerationType.SEQUENCE,generator = "student_sequence")
```

```
private Long id;
     private String name;
     private Integer testscore;
     public Student(){
}
     public Long getId() {
     return id;
}
     public void setId(Long id) {
     this.id = id;
}
     public String getName() {
     return name;
}
     public void setName(String name) {
```

```
this.name = name;
}
     public Integer getTestscore() {
     return testscore;
}
     public void setTestscore(Integer testscore) {
     this.testscore = testscore;
}
}
     package com.example.demo.repo;
     import org.springframework.data.repository.CrudRepository;
     import org.springframework.stereotype.Repository;
     import com.example.demo.entity.Student;
     @Repository
     public interface StudentRepo extends CrudRepository<Student,
```

```
Long> {
}
     package com.example.demo.service;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.stereotype.Service;
     import com.example.demo.entity.Student;
     import com.example.demo.repo.StudentRepo;
     @Service
     public class StudentService {
     @Autowired
     StudentRepo repo;
     public Student save(Student student) {
     return repo.save(student);
}
```

```
public Student update(Student stu) {
     return repo.save(stu);
}
     public String delete(Long stuld) {
     repo.deleteByld(stuld);
     return "User " + stuld + " deleted..";
}
     public Optional<Student> get(Long id) {
     return repo.findById(id);
}
}
     package com.example.demo;
     import org.junit.jupiter.api.Test;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.boot.test.context.SpringBootTest;
```

```
import com.example.demo.entity.Student;
     import com.example.demo.service.StudentService;
     @SpringBootTest
     class Demo1ApplicationTests {
     @Autowired
     StudentService service;
     @Test
     void curdOperations() {
     Student stu = new Student();
     stu.setName("nitin");
     stu.setTestscore(20);
     System.out.println(service.save(stu));
     Student savedStu = service.save(stu);
     System.out.println(savedStu.getId() + ": "+ savedStu.getName()
+ ": "+ savedStu.getTestscore());
     Student stuFromDB = service.get(1I).get();
```

```
System.out.println(stuFromDB.getId() + " : "+
stuFromDB.getName() + " : "+ stuFromDB.getTestscore());

Student updateStu = stuFromDB;

updateStu.setName("Nitin-1");

Student save = service.save(updateStu);

System.out.println(save.getId() + " : "+ save.getName() + " : "+
save.getTestscore());

}
```

#### **Assignment: 3 Spring Web and REST API**

```
package com.example.demo.controller;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
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.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo.entity.Student;
import com.example.demo.service.StudentService;
@RestController
public class StudentController {
@Autowired
public StudentService service;
@GetMapping("/{id}")
public Student getStudent(@PathVariable("id") Long id) {
Optional < Student > optional = service.get(id);
if(optional.isPresent())
```

```
return service.get(id).get();
     return new Student();
}
     @PostMapping
     public Student saveStudent(@RequestBody Student student) {
     return service.save(student);
}
     @PutMapping
     public Student updateStudent(@RequestBody Student student) {
     return service.save(student);
}
     @DeleteMapping("/{id}")
     public String deleteStudent(@PathVariable("id") Long id) {
     service.delete(id);
     return "deleted Successfully "+ id;
```

```
}
}
     package com.example.demo.entity;
     import javax.persistence.Entity;
     import javax.persistence.GeneratedValue;
     import javax.persistence.GenerationType;
     import javax.persistence.ld;
     import javax.persistence.SequenceGenerator;
     import javax.persistence.Table;
     @Entity
     @Table(name = "student")
     public class Student {
     @Id
     @SequenceGenerator(name = "student_sequence", initialValue
= 1)
```

```
@GeneratedValue(strategy =
GenerationType.SEQUENCE,generator = "student_sequence")
     private Long id;
     private String name;
     private Integer testscore;
     public Student(){
     }
     public Long getId() {
     return id;
}
     public void setId(Long id) {
     this.id = id;
}
     public String getName() {
     return name;
}
```

```
public void setName(String name) {
     this.name = name;
}
     public Integer getTestscore() {
     return testscore;
}
     public void setTestscore(Integer testscore) {
     this.testscore = testscore;
}
}
     package com.example.demo.repo;
     import org.springframework.data.repository.CrudRepository;
     import org.springframework.stereotype.Repository;
     import com.example.demo.entity.Student;
     @Repository
```

```
public interface StudentRepo extends CrudRepository<Student,
Long> {
}
     package com.example.demo.service;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.stereotype.Service;
     import com.example.demo.entity.Student;
     import com.example.demo.repo.StudentRepo;
     @Service
     public class StudentService {
     @Autowired
     StudentRepo repo;
     public Student save(Student student) {
     return repo.save(student);
```

```
}
     public Student update(Student stu) {
     return repo.save(stu);
}
     public String delete(Long stuld) {
     repo.deleteByld(stuld);
     return "User " + stuld + " deleted..";
}
     public Optional<Student> get(Long id) {
     return repo.findById(id);
}
}
     package com.example.demo.controller;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
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.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo.entity.Student;
import com.example.demo.service.StudentService;
@RestController
public class StudentController {
@Autowired
public StudentService service;
@GetMapping("/{id}")
public Student getStudent(@PathVariable("id") Long id) {
```

```
Optional < Student > optional = service.get(id);
     if(optional.isPresent())
     return service.get(id).get();
     return new Student();
}
     @PostMapping
     public Student saveStudent(@RequestBody Student student) {
     return service.save(student);
}
     @PutMapping
     public Student updateStudent(@RequestBody Student student) {
     return service.save(student);
}
     @DeleteMapping("/{id}")
     public String deleteStudent(@PathVariable("id") Long id) {
```

```
service.delete(id);
     return "deleted Successfully "+ id;
}
}
     package com.example.demo.entity;
     import javax.persistence.Entity;
     import javax.persistence.GeneratedValue;
     import javax.persistence.GenerationType;
     import javax.persistence.ld;
     import javax.persistence.SequenceGenerator;
     import javax.persistence.Table;
     @Entity
     @Table(name = "student")
     public class Student {
     @ld
```

```
@SequenceGenerator(name = "student sequence", initialValue
= 1)
     @GeneratedValue(strategy =
GenerationType.SEQUENCE,generator = "student_sequence")
     private Long id;
     private String name;
     private Integer testscore;
     public Student(){
}
     public Long getId() {
     return id;
}
     public void setId(Long id) {
     this.id = id;
}
     public String getName() {
```

```
return name;
}
     public void setName(String name) {
     this.name = name;
}
     public Integer getTestscore() {
     return testscore;
}
     public void setTestscore(Integer testscore) {
     this.testscore = testscore;
}
}
     package com.example.demo.repo;
     import org.springframework.data.repository.CrudRepository;
     import org.springframework.stereotype.Repository;
```

```
import com.example.demo.entity.Student;
     @Repository
     public interface StudentRepo extends CrudRepository<Student,
Long> {
}
     package com.example.demo.service;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.stereotype.Service;
     import com.example.demo.entity.Student;
     import com.example.demo.repo.StudentRepo;
     @Service
     public class StudentService {
     @Autowired
     StudentRepo repo;
     public Student save(Student student) {
```

```
return repo.save(student);
}
     public Student update(Student stu) {
     return repo.save(stu);
}
     public String delete(Long stuld) {
     repo.deleteByld(stuld);
     return "User " + stuld + " deleted..";
}
     public Optional<Student> get(Long id) {
     return repo.findById(id);
}
}
```

package com.example.demo.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.DeleteMapping; 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.PutMapping; import org.springframework.web.bind.annotation.RequestBody; import org.springframework.web.bind.annotation.RestController; import com.example.demo.entity.Student;

import com.example.demo.service.StudentService;

@RestController

public class StudentController {

@Autowired

```
public StudentService service;
     @GetMapping("/{id}")
     public Student getStudent(@PathVariable("id") Long id) {
     Optional < Student > optional = service.get(id);
     if(optional.isPresent())
     return service.get(id).get();
     return new Student();
}
     @PostMapping
     public Student saveStudent(@RequestBody Student student) {
     return service.save(student);
}
     @PutMapping
     public Student updateStudent(@RequestBody Student student) {
     return service.save(student);
```

```
}
     @DeleteMapping("/{id}")
     public String deleteStudent(@PathVariable("id") Long id) {
     service.delete(id);
     return "deleted Successfully "+ id;
}
}
     package com.example.demo.entity;
     import javax.persistence.Entity;
     import javax.persistence.GeneratedValue;
     import javax.persistence.GenerationType;
     import javax.persistence.ld;
     import javax.persistence.SequenceGenerator;
     import javax.persistence.Table;
     @Entity
```

```
@Table(name = "student")
     public class Student {
     @ld
     @SequenceGenerator(name = "student_sequence", initialValue
= 1)
     @GeneratedValue(strategy =
GenerationType.SEQUENCE,generator = "student_sequence")
     private Long id;
     private String name;
     private Integer testscore;
     public Student(){
}
     public Long getId() {
     return id;
}
     public void setId(Long id) {
```

```
this.id = id;
}
     public String getName() {
     return name;
}
     public void setName(String name) {
     this.name = name;
}
     public Integer getTestscore() {
     return testscore;
}
     public void setTestscore(Integer testscore) {
     this.testscore = testscore;
}
}
```

```
package com.example.demo.repo;
     import org.springframework.data.repository.CrudRepository;
     import org.springframework.stereotype.Repository;
     import com.example.demo.entity.Student;
     @Repository
     public interface StudentRepo extends CrudRepository<Student,
Long> {
}
     package com.example.demo.service;
     import java.util.Optional;
     import org.springframework.beans.factory.annotation.Autowired;
     import org.springframework.stereotype.Service;
     import com.example.demo.entity.Student;
     import com.example.demo.repo.StudentRepo;
     @Service
     public class StudentService {
```

```
@Autowired
     StudentRepo repo;
     public Student save(Student student) {
     return repo.save(student);
}
     public Student update(Student stu) {
     return repo.save(stu);
}
     public String delete(Long stuld) {
     repo.deleteByld(stuld);
     return "User " + stuld + " deleted..";
}
     public Optional<Student> get(Long id) {
     return repo.findById(id);
}
```

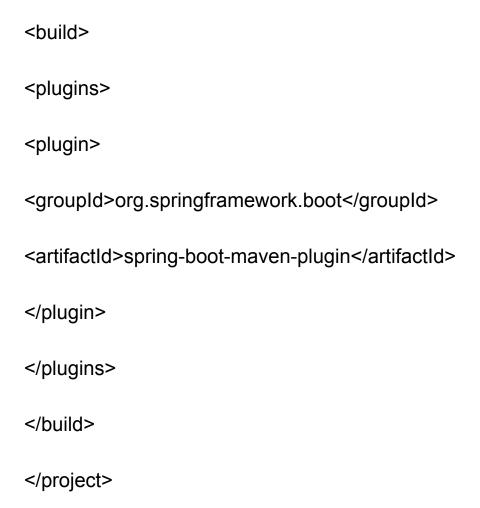
#### **Assignment: 4 Add context-path and change the server**

```
Application.properties
     spring.datasource.url=jdbc:mysql://localhost:3306/mydb
     spring.datasource.username=root
     spring.datasource.password=root
     spring.jpa.hibernate.ddl-auto=update
     server.servlet.context-path=/studentapi
     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>2.5.5</version>
<relativePath /> <!-- lookup parent from repository -->
</parent>
<groupId>com.example.springweb</groupId>
<artifactId>productrestapi</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>productrestapi</name>
<description>Demo project for Spring Boot Web</description>
cproperties>
<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>
<exclusions>
<exclusion>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
</exclusion>
</exclusions>
</dependency>
```

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jetty</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>
```



# Assignment: 5 Enable health checks and customize info endpoint

package com.example.student.health;
import org.springframework.boot.actuate.info.Info.Builder;
import org.springframework.boot.actuate.info.InfoContributor;
import org.springframework.stereotype.Component;

```
@Component
     public class CustomInfoContributor implements InfoContributor {
     @Override
     public void contribute(Builder builder) {
     boolean error = false;
     if (error) {
     builder.withDetail("Error Key", 123).build();
}
     builder.build();
}
}
     package com.example.student.health;
     import org.springframework.boot.actuate.info.Info.Builder;
     import org.springframework.boot.actuate.info.InfoContributor;
     import org.springframework.stereotype.Component;
```

```
@Component
     public class CustomInfoContributor implements InfoContributor {
     @Override
     public void contribute(Builder builder) {
     boolean error = false;
     if (error) {
     builder.withDetail("Error Key", 123).build();
}
     builder.build();
}
}
```