

Task 1 : Create a Spring boot project configure with Github and database write junit test case for a student details name , age ,salary and create a jar and deploy it.

Student application:

```
package com.upendra.student;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import io.swagger.v3.oas.annotations.OpenAPIDefinition;
import io.swagger.v3.oas.annotations.info.Info;

@OpenAPIDefinition(info = @Info(title = "student"))
@SpringBootApplication
public class StudentApplication {

    public static void main(String[] args) {
        SpringApplication.run(StudentApplication.class, args);
    }
}
```

Controller:

```
package com.upendra.student.controller;

import java.security.PublicKey;
import java.util.List;

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

import com.upendra.student.model.Student;
import com.upendra.student.service.StudentService;

@RestController
@RequestMapping("/student")
public class StudentCont {

    @Autowired
    StudentService studentService;

    @GetMapping("/getByName/{name}")
    public List<Student> getByName(@PathVariable("name") String name) {
        return studentService.findByName(name);
    }

    @GetMapping("/getByAge/{age}")
    public List<Student> getByAge(@PathVariable("age") int age) {
        return studentService.findByAge(age);
    }

    @GetMapping("/getBySalary/{salary}")
    public List<Student> getBySalary(@PathVariable("salary") long salary) {
        return studentService.findBySalary(salary);
    }
}
```

```
}
```

```
@PostMapping("/post")
```

```
public Student post(@RequestBody Student student)
```

```
{
```

```
    return studentService.post(student);
```

```
}
```

```
}
```

Service:

```
package com.upendra.student.service;
```

```
import java.util.List;
```

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

```
import org.springframework.stereotype.Service;
```

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

```
import com.upendra.student.model.Student;
```

```
import com.upendra.student.repo.StudentRepo;
```

```
@Service
```

```
public class StudentService {
```

```
    @Autowired
```

```
    StudentRepo repo;
```

```
    public List<Student> findByName(String name) {
```

```

        return repo.findByName(name);
    }

    public List<Student> findByAge(int age) {
        return repo.findByAge(age);
    }

    public List<Student> findBySalary(long salary) {
        return repo.findBySalary(salary);
    }

    public Student post(Student student)
    {
        return repo.save(student);
    }
}

```

Repository

```
package com.upendra.student.repo;
```

```
import java.util.List;
```

```
import org.springframework.data.mongodb.repository.MongoRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.upendra.student.model.Student;
```

@Repository

```
public interface StudentRepo extends MongoRepository<Student, String>{
```

```

    List<Student> findByName(String name);
}

```

```
        List<Student> findBySalary(long salary);

        List<Student> findByAge(int age);
    }
}
```

Test Classes:

FOR CONTROLLER

```
package com.upendra.student.controller;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
import static org.mockito.Mockito.when;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.junit.jupiter.api.Test;
```

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

```
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
```

```
import org.springframework.boot.test.mock.mockito.MockBean;
```

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

```
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
```

```
import com.upendra.student.model.Student;
```

```
import com.upendra.student.service.StudentService;
```

```
@WebMvcTest(value = StudentCont.class)
```

```
class StudentContTest {
```

```
    @Autowired
```

```
MockMvc mockMvc;
```

```
@MockBean
```

```
StudentService studentService;
```

```
@Test
```

```
void testGetByName() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findByName("upendra")).thenReturn(list);
```

```
}
```

```
@Test
```

```
void testGetBySalary() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findBySalary(246465)).thenReturn(list);
```

```
}
```

```
@Test
```

```
void testGetByAge() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findByAge(0)).thenReturn(list);
```

```
}
```

```
}
```

FOR SERVICE:

```
package com.upendra.student.service;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
import static org.mockito.Mockito.when;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.junit.jupiter.api.Test;
```

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

```
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
```

```
import org.springframework.boot.test.mock.mockito.MockBean;
```

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

```
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
```

```
import com.upendra.student.model.Student;
```

```
@WebMvcTest(value = StudentService.class)
```

```
class StudentServiceTest {
```

```
    @Autowired
```

```
    MockMvc mockMvc;
```

```
    @MockBean
```

```
StudentService studentService;
```

```
@Test
```

```
void testGetByName() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findByName("upendra")).thenReturn(list);
```

```
}
```

```
@Test
```

```
void testGetBySalary() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findBySalary(246465)).thenReturn(list);
```

```
}
```

```
@Test
```

```
void testGetByAge() {
```

```
    List<Student> list = new ArrayList<>();
```

```
    Student student = new Student("Upendra", 44, 40000000);
```

```
    list.add(student);
```

```
    when(studentService.findByAge(0)).thenReturn(list);
```

```
}
```


}

Test coverage Screen shot :

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with folders for `com.upendra.student` and `com.upendra.student.service`.
- JUnit Console:** Shows test results: "Finished after 11.952 seconds", "Runs: 7/7", "Errors: 0", "Failures: 0".
- Code Editor:** Displays `StudentServiceTest.java` with the following code:

```
1 package com.upendra.student.service;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 @WebMvcTest(value = StudentService.class)
6 class StudentServiceTest {
7     @Autowired
8     MockMvc mockMvc;
9     @MockBean
10    StudentService studentService;
11
12    @Test
13    void testGetByName() {
14        List<Student> list = new ArrayList<>();
15
16        Student student = new Student("Upendra", 44, 40000000);
17        list.add(student);
18        when(studentService.findByName("upendra")).thenReturn(list);
19    }
20
21    @Test
22    void testGetBySalary() {
23        List<Student> list = new ArrayList<>();
24
25        Student student = new Student("Upendra", 44, 40000000);
26        list.add(student);
27    }
28 }
```
- Outline:** Shows the class hierarchy: `com.upendra.student.service` > `StudentServiceTest` > `mockMvc: MockMvc` > `studentService: StudentService` > `testGetByName(): void` > `testGetBySalary(): void` > `testGetByAge(): void`.
- Coverage Table:**

Element	Covera...	Covered Ins...	Missed Instr...	Total Instru...
src/test/java	100.0 %	154	0	154
com.upendra.student	100.0 %	4	0	4
com.upendra.student.con	100.0 %	75	0	75
com.upendra.student.serv	100.0 %	75	0	75

Github Screen shot:

The screenshot shows a GitHub repository page with the following details:

- Repository:** `upendraRambarki/NOV20-Java-practice/tree/master`
- Branches:** 2 branches, 0 tags
- Commit History:**
 - `uxr070301` updated 2565cea 11 minutes ago 11 commits
 - `.metadata` updated 11 minutes ago
 - `Customer` day3 5 hours ago
 - `Day1` update yesterday
 - `Day2` day 2 update 20 hours ago
 - `Person` day3 2 hours ago
 - `Student` updated 11 minutes ago
 - `.gitignore` day2 yesterday
- Releases:** No releases published. [Create a new release](#)
- Packages:** No packages published. [Publish your first package](#)
- Help:** Help people interested in this repository understand your project by adding a README. [Add a README](#)

Swagger UI interface showing a REST API endpoint: `http://localhost:8081/student/getByAge/22`.

Curl

```
curl -X 'GET' \
  'http://localhost:8081/student/getByAge/22' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8081/student/getByAge/22
```

Server response

Code **Details**

200

Response body

```
{
  "name": "upendra",
  "age": 22,
  "salary": 28000
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Wed, 22 Nov 2023 10:02:02 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code	Description
------	-------------

Links

Eclipse IDE interface showing a Java project with a Git push notification and commit details.

Push Results: Java NOV 20 refs/heads/master - origin

Pushed to Java NOV 20 refs/heads/master - origin

master → master [ae5abc9.2565cea] (1)

ae5abc9: update (uxr070301 on 2023-11-22 16:06:04)

Message Details

Repository <https://github.com/upendraRambarki/NOV20-Java-practice.git>

Commit Message

Author: uxr070301 <uxr070301@darden.com>

Committer: uxr070301 <uxr070301@darden.com>

Unstaged Changes (1)

- > 0.log - .metadata/.plugins/org.eclipse.m2e.logback

Staged Changes (0)

Push HEAD...