

**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> getBySytudent(@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 CONTROLER

```
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:

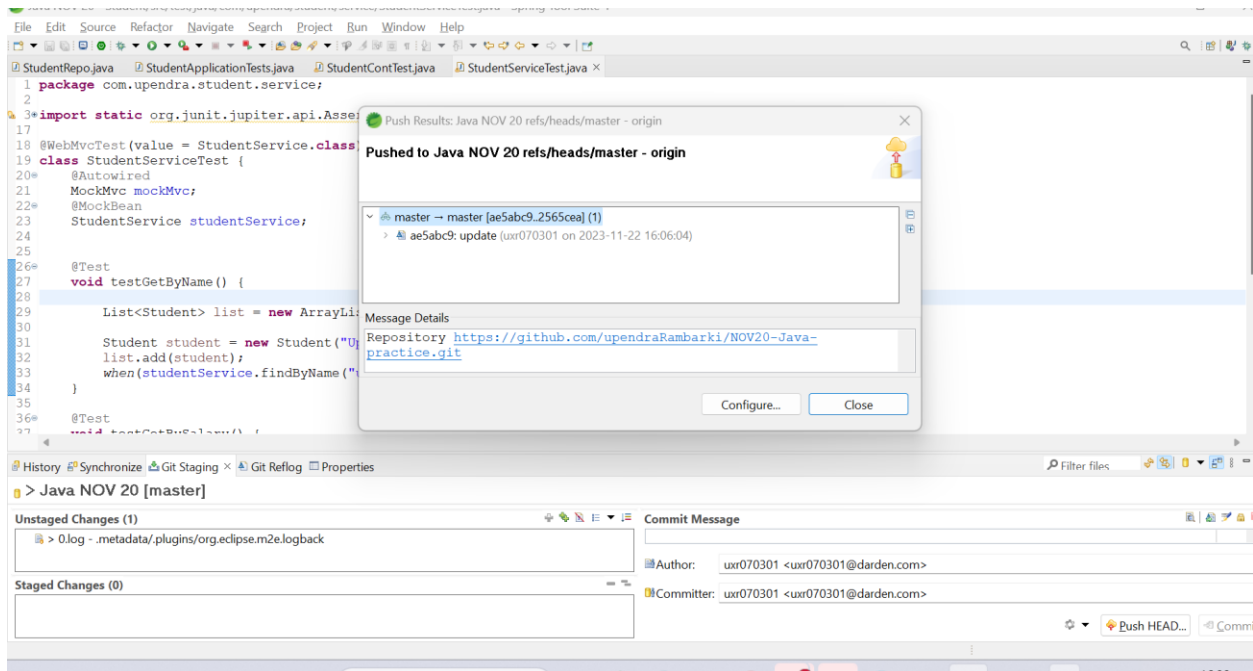
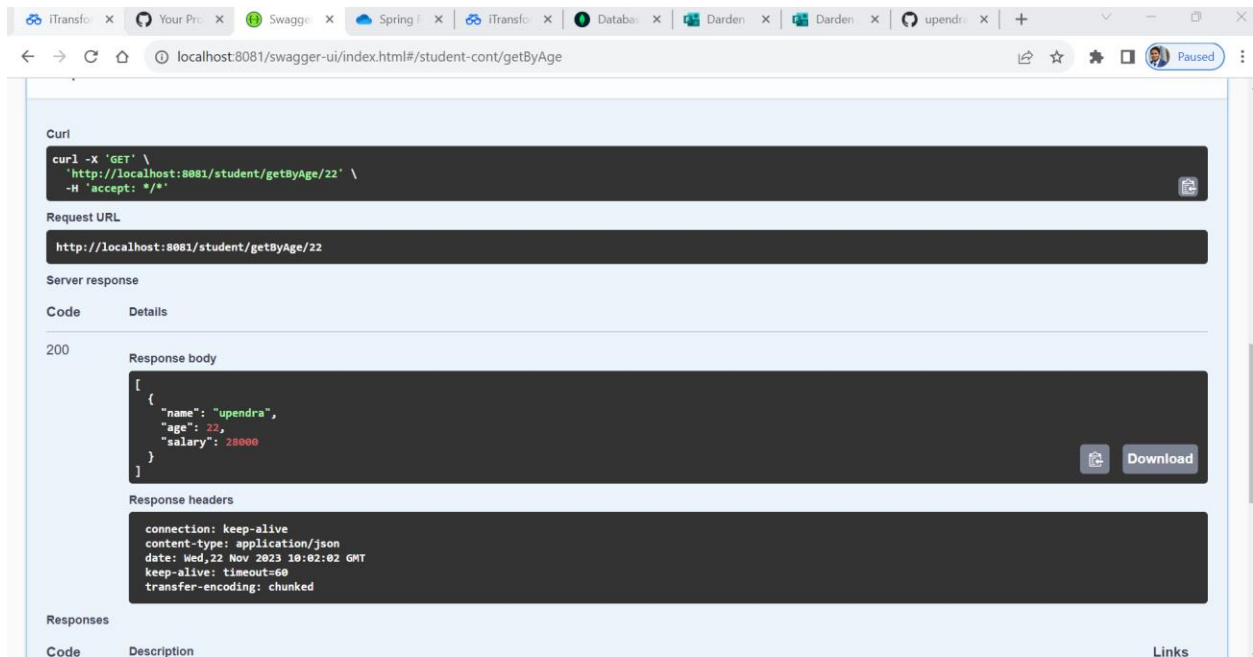
- Package Explorer:** Shows the project structure with tests passing (7/7 runs, 0 errors, 0 failures).
- Code Editor:** Displays `StudentServiceTest.java` with two test methods: `testGetByName()` and `testGetBySalary()`. Both methods are highlighted in green, indicating 100% coverage.
- Outline:** Lists the classes and methods in the project, including `StudentServiceTest` and its methods.
- Coverage Table:** A table at the bottom showing coverage details for the current file and its nested elements.

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`
- Branch:** `master` (2 branches, 0 tags)
- Status:** This branch is 11 commits ahead, 2 commits behind main.
- Files:** A list of files and folders updated recently, including `.metadata`, `Customer`, `Day1`, `Day2`, `Person`, `Student`, and `.gitignore`.
- Releases:** No releases published. [Create a new release](#)
- Packages:** No packages published. [Publish your first package](#)
- Help:** A button to [Add a README](#) to help people understand the project.

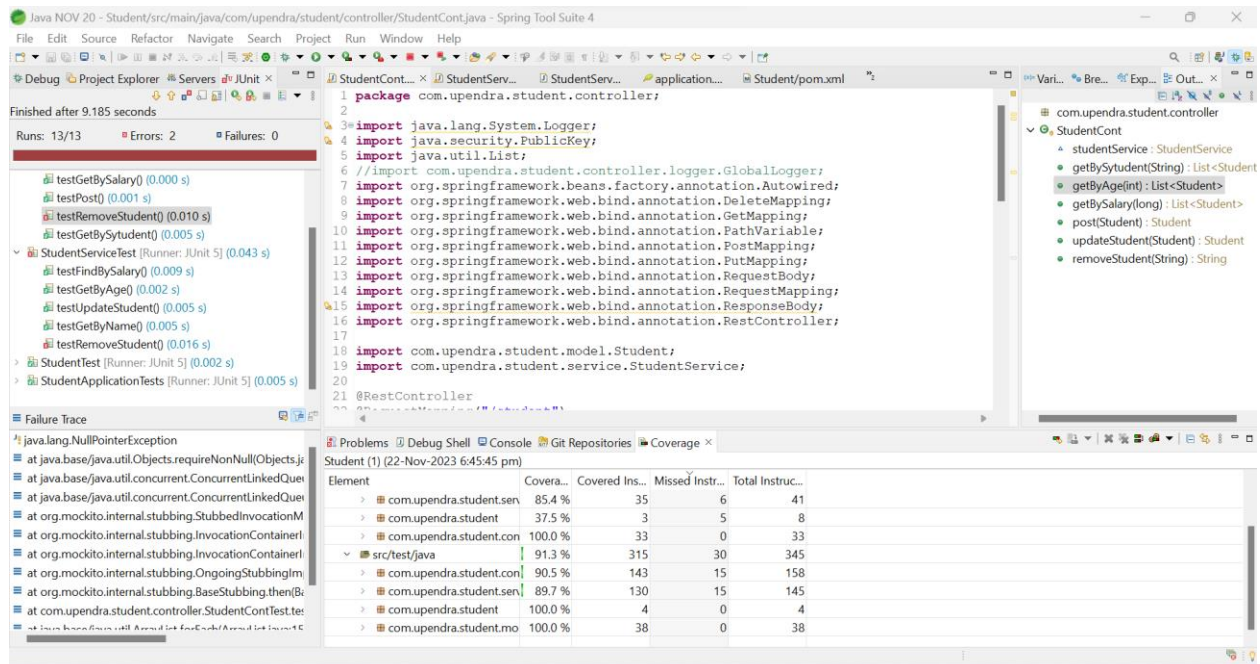


## Task 2

Github link

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





## Controller

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

```
import java.lang.System.Logger;
import java.security.PublicKey;
import java.util.List;
//import com.upendra.student.controller.logger.GlobalLogger;
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.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
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 {
    // private Logger log = GlobalLogger.getLogger(StudentCont.class);
    @Autowired
    StudentService studentService;

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

```

//         logger.trace("getting by 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);
    }

    @PutMapping("/update")
    public Student updateStudent(@RequestBody Student student)
    {
        return studentService.updateStudent(student);
    }

    @DeleteMapping("/delete")
    public String removeStudent(String name)
    {
        return studentService.deleteStudent(name);
    }
}

```

## 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);

}

public Student updateStudent(Student student) {

// TODO Auto-generated method stub

return repo.save(student);

}

public String deleteStudent(String name) {

```
        // TODO Auto-generated method stub
        List<Student> student = repo.findByName(name);
        repo.deleteAll(student);
        return "deleted";
    }
}
```