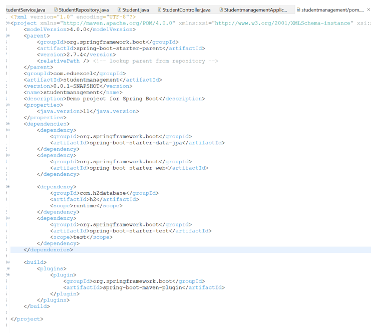
Assignment 3: Student Management microservices

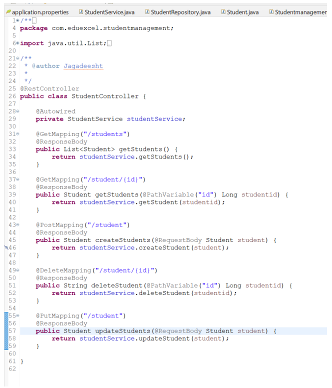
With Spring boot setup having spring starter web, jpa and h2database below microservice has been developed

Pom.xml: Describing what all the spring start dependencies have required



```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:
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.4</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.eduexcel</groupId>
    <artifactId>studentmanagement</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>studentmanagement</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>11</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>com.h2database</groupId>
            <artifactId>h2</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>
```

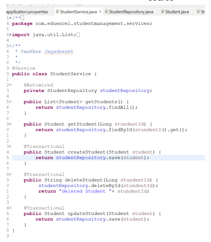
Request dispatcher called Controller with @RestController annotation used to have all the APIs as resources



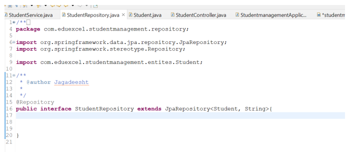
```java
/**
package com.eduexcel.studentmanagement;

import java.util.List;

/**
 * @author Jagadeesht
 *
 */
@RestController
public class StudentController {

    @Autowired
    private StudentService studentService;

    @GetMapping("/students")
    @ResponseBody
    public List<Student> getStudents() {
        return studentService.getStudents();
    }

    @GetMapping("/student/{id}")
    @ResponseBody
    public Student getStudents(@PathVariable("id") Long studentid) {
        return studentService.getStudent(studentid);
    }

    @PostMapping("/student")
    @ResponseBody
    public Student createStudents(@RequestBody Student student) {
        return studentService.createStudent(student);
    }

    @DeleteMapping("/student/{id}")
    @ResponseBody
    public String deleteStudent(@PathVariable("id") Long studentid) {
        return studentService.deleteStudent(studentid);
    }

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

}
```

In Controller class, service calls will be invoked by autowire Service class to get the API responses

Below is the service class to process the request and get the response from data source



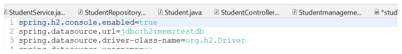
```java
/**
package com.eduexcel.studentmanagement.services;

import java.util.List;

/**
 * @author Jagadeesht
 *
 */
@Service
public class StudentService {

    @Autowired
    private StudentRepository studentRepository;

    public List<Student> getStudents() {
        return studentRepository.findAll();
    }

    public Student getStudent(Long stundentId) {
        return studentRepository.findById(stundentId).get();
    }

    @Transactional
    public Student createStudent(Student student) {
        return studentRepository.save(student);
    }

    @Transactional
    public String deleteStudent(Long stundentId) {
         studentRepository.deleteById(stundentId);
         return "deleted Student "+ stundentId;
    }

    @Transactional
    public Student updateStudent(Student student) {
        return studentRepository.save(student);
    }
}
```



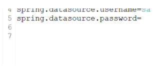
```java
/**
package com.eduexcel.studentmanagement.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.eduexcel.studentmanagement.entites.Student;

/**
 * @author Jagadeesht
 *
 */
@Repository
public interface StudentRepository extends JpaRepository<Student, String>{



}
```

Here we make use of JPA specification to make the DB operations on the h2 database, JPARepository will provide default methods to call CRUD operations like findAll, findById, deleteById, save and etc.

Below is the properties used in application.properties to configure the h2database

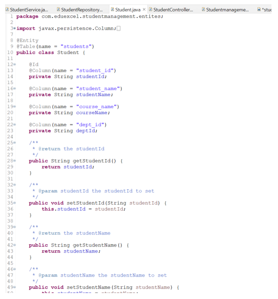


```properties
spring.h2.console.enabled=true
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driver-class-name=org.h2.Driver
```



```
spring.datasource.username=sa
spring.datasource.password=
```

Student object is model object used for student management service

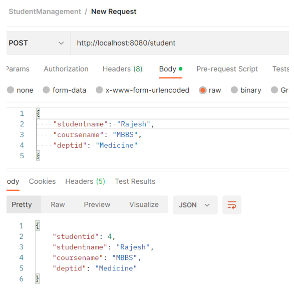


StudentService.ja... StudentRepository... Student.java StudentController... Studentmanageme... *stuc

```
package com.eduexcel.studentmanagement.entites;

import javax.persistence.Column;

@Entity
@Table(name = "students")
public class Student {

    @Id
    @Column(name = "student_id")
    private String studentId;

    @Column(name = "student_name")
    private String studentName;

    @Column(name = "course_name")
    private String courseName;

    @Column(name = "dept_id")
    private String deptId;

    /**
     * @return the studentId
     */
    public String getStudentId() {
        return studentId;
    }

    /**
     * @param studentId the studentId to set
     */
    public void setStudentId(String studentId) {
        this.studentId = studentId;
    }

    /**
     * @return the studentName
     */
    public String getStudentName() {
        return studentName;
    }

    /**
     * @param studentName the studentName to set
     */
    public void setStudentName(String studentName) {
```

Testing above operations

1. create a student



StudentManagement / New Request

POST http://localhost:8080/student

Params Authorization Headers (8) Body Pre-request Script Tests

none form-data x-www-form-urlencoded [✓ raw] binary Gr

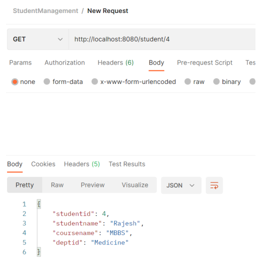
```
{
    "studentname": "Rajesh",
    "coursename": "MBBS",
    "deptid": "Medicine"
}
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "studentid": 4,
    "studentname": "Rajesh",
    "coursename": "MBBS",
    "deptid": "Medicine"
}
```

2. get a student



StudentManagement / New Request

GET http://localhost:8080/student/4

Params Authorization Headers (6) Body Pre-request Script Tes

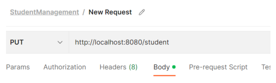
[✓ none] form-data x-www-form-urlencoded raw binary

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "studentid": 4,
    "studentname": "Rajesh",
    "coursename": "MBBS",
    "deptid": "Medicine"
}
```

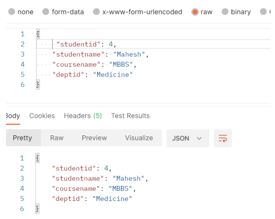
3. update a student



StudentManagement / New Request

PUT http://localhost:8080/student

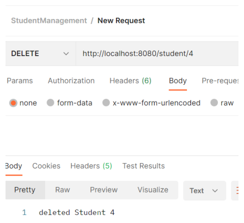
Params Authorization Headers (8) Body Pre-request Script Tes



```
{
    "studentid": 4,
    "studentname": "Mahesh",
    "coursename": "MBBS",
    "deptid": "Medicine"
}
```

```
{
    "studentid": 4,
    "studentname": "Mahesh",
    "coursename": "MBBS",
    "deptid": "Medicine"
}
```

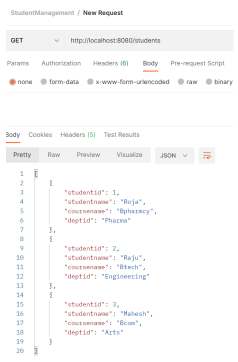
4. delete a student


StudentManagement / New Request

DELETE http://localhost:8080/student/4

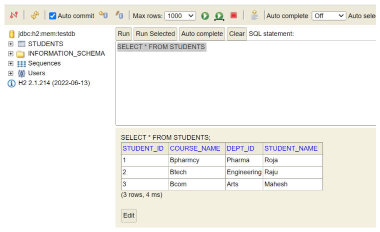
```
deleted Student 4
```

5. Get All Students


StudentManagement / New Request

GET http://localhost:8080/students

```
[
    {
        "studentid": 1,
        "studentname": "Roja",
        "coursename": "Bpharmcy",
        "deptid": "Pharma"
    },
    {
        "studentid": 2,
        "studentname": "Raju",
        "coursename": "Btech",
        "deptid": "Engineering"
    },
    {
        "studentid": 3,
        "studentname": "Mahesh",
        "coursename": "Bcom",
        "deptid": "Arts"
    }
]
```



```
SELECT * FROM STUDENTS
```

SELECT * FROM STUDENTS;

STUDENT_ID	COURSE_NAME	DEPT_ID	STUDENT_NAME
1	Bpharmcy	Pharma	Roja
2	Btech	Engineering	Raju
3	Bcom	Arts	Mahesh

(3 rows, 4 ms)