03 - Student Service and Repository

In the Spring Boot project, we have implemented a basic structure following the MVC pattern, that utilizes Spring JDBC for database interaction. The application demonstrates:

1. **Model Layer**: Represents the entity that maps to a table in the database.
2. **Repository Layer**: Manages interactions with the database for saving and retrieving data.
3. **Service Layer**: Contains the business logic, interacting with the repository.
4. **Application Layer**: Acts as the entry point and coordinates the process of adding and retrieving students.

Example:

Student.java:

```
package com.telusko.SpringJDBCEx.model;

@Component
@Scope("prototype")
public class Student {
    private int rollNo;
    private String name;
    private int marks;
    // Getters and setters...
}
```

StudentRepo.java:

```
package com.telusko.SpringJDBCEx.repo;

@Repository
public class StudentRepo {

    public void save(Student s) {
        System.out.println("Student added to the database.");
    }

    public List<Student> findAll() {
        List<Student> students = new ArrayList<>();
        return students;
    }
}
```

StudentService.java:

```
package com.telusko.SpringJDBCEx.service;

@Service
public class StudentService {
    private StudentRepo repo;
    @Autowired
    public void setRepo(StudentRepo repo) {
        this.repo = repo;
    }

    // Method to add a student to the repository
    public void addStudent(Student s) {
        repo.save(s);
    }

    // Method to retrieve all students from the repository
    public List<Student> getStudents() {
        return repo.findAll();
     }
}
```

Application.java:

```
package com.telusko.SpringJDBCEx;

@SpringBootApplication
public class Application {
   public static void main(String[] args) {
      ApplicationContext context = SpringApplication.run(Application.class, args);
      Student s = context.getBean(Student.class);
      StudentService service = context.getBean(StudentService.class);

      // Set the properties for the student
      s.setRollNo(101);
      s.setName("Navin");
      s.setMarks(78);

      // Add the student using the service
      service.addStudent(s);

      // Retrieve all students from the service
      List<Student> students = service.getStudents();
      System.out.println(students);
   }
}
```

Code Link:

https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/4%20Spring%20JDBC/4.3%20Student%20Service%20And%20Repo