Spring Boot Jdbc using MySQL CURD

Project Setup:-
#1. Lombok,JDBC,MySQL Driver/Oracle Driver
#2 MySQL version : added in pom.xml
#3 Connection Properties (DataSource)
--application.yml--

```
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://localhost:3306/nit
    username: root
    password: root
```

#4 Create table in MySQL Database.

```
show databases;
create database nit;
use nit;
show tables;
 CREATE TABLE STUDENT(
 SID INT,
 SNAME VARCHAR(25),
 SFEE DOUBLE,
 COURSE VARCHAR(25),
 DISCOUNT DOUBLE
 );
```

---Module Implementation----------
Module name : Student
1. Model class
2. IDao
3. DaoImpl
4. IService
5. ServiceImpl
6. RowMapper
7. Runner

a. Define one abstract method in IDao
b. Implement abstract method in DaoImpl using jt (update/query/queryForObject)

c. Copy abstract method from IDao to IService

d. Implement abstract method in ServiceImpl using dao.

****

pom.xml:-

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

		<!-- <dependency>
			<groupId>com.oracle.ojdbc</groupId>
			<artifactId>ojdbc8</artifactId>
			<scope>runtime</scope>
		</dependency> -->
		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<version>5.1.46</version><!--$NO-MVN-MAN-VER$-->
			<scope>runtime</scope>
		</dependency>
		<dependency>
			<groupId>org.projectlombok</groupId>
			<artifactId>lombok</artifactId>

		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
			<exclusions>
				<exclusion>
					<groupId>org.junit.vintage</groupId>
				<artifactId>junit-vintage-engine</artifactId>
				</exclusion>
			</exclusions>
		</dependency>
	</dependencies>
```

1. Model class

```
package in.nit.model;

import lombok.Data;

@Data
public class Student {

	private Integer stdId;
	private String stdName;
	private Double stdFee;
	private String stdCourse;
	private Double discount;


}
```

2. IStudentDao

```
package in.nit.dao;

import java.util.List;

import in.nit.model.Student;

public interface IStudentDao {

	public int saveStudent(Student s);
	public int updateStudent(Student s);
	public int deleteStudent(Integer id);


	public Student getOneStudent(Integer id);
	public List<Student> getAllStudent();


}
```

3. StudentDaoImpl

```
package in.nit.dao.impl;

import java.util.List;

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

```
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import in.nit.dao.IStudentDao;
import in.nit.model.Student;

@Repository
public class StudentDaoImpl
		implements IStudentDao
{
	@Autowired
	private JdbcTemplate jt;

	public int saveStudent(Student s) {
		String sql="INSERT INTO STUDENT VALUES(?,?,?,?,?)";
		//jt.update(String sql,Object... args)
		int count=jt.update(sql,
				s.getStdId(),s.getStdName(),
				s.getStdFee(),s.getStdCourse(),
				s.getDiscount());
		return count;
	}

	public int updateStudent(Student s) {
		String sql="UPDATE STUDENT SET
SNAME=?,SFEE=?,COURSE=?,DISCOUNT=? WHERE SID=?";
		int count=jt.update(sql,
				s.getStdName(),s.getStdFee(),
				s.getStdCourse(),s.getDiscount(),
				s.getStdId());
		return count;
	}

	public int deleteStudent(Integer id) {
		String sql="DELETE FROM STUDENT WHERE SID=?";
		int count=jt.update(sql,id);
		return count;
	}


	public Student getOneStudent(Integer id) {
		String sql="SELECT * FROM STUDENT WHERE SID=?";
		//Student s=jt.queryForObject(sql,new
StudentRowMapper(),id);
		Student sob=jt.queryForObject(sql,(rs,rowNum)->{
			Student s=new Student();
```

```
                s.setStdId(rs.getInt("SID"));
                s.setStdName(rs.getString("SNAME"));
                s.setStdFee(rs.getDouble("SFEE"));
                s.setStdCourse(rs.getString("COURSE"));
                s.setDiscount(rs.getDouble("DISCOUNT"));
                return s;
          },id);
          return sob;
    }

    public List<Student> getAllStudent() {
          String sql="SELECT * FROM STUDENT";
          List<Student> list=jt.query(sql,(rs,count)->{
                Student s=new Student();
                s.setStdId(rs.getInt("SID"));
                s.setStdName(rs.getString("SNAME"));
                s.setStdFee(rs.getDouble("SFEE"));
                s.setStdCourse(rs.getString("COURSE"));
                s.setDiscount(rs.getDouble("DISCOUNT"));
                return s;
          });
          return list;
    }

}
```

4. IStudentService

```
package in.nit.service;

import java.util.List;

import in.nit.model.Student;

public interface IStudentService {

    public int saveStudent(Student s);
    public int updateStudent(Student s);
    public int deleteStudent(Integer id);

    public Student getOneStudent(Integer id);
    public List<Student> getAllStudent();
}
```

5. StudentServiceImpl

```
package in.nit.service.impl;

import java.util.List;
import java.util.stream.Collectors;

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

import in.nit.dao.IStudentDao;
import in.nit.model.Student;
import in.nit.service.IStudentService;

@Service
public class StudentServiceImpl
	implements IStudentService
{
	@Autowired
	private IStudentDao dao;

	public int saveStudent(Student s) {
		double sfee=s.getStdFee();
		s.setDiscount(sfee * 5 /100.0);
		return dao.saveStudent(s);
	}

	public int updateStudent(Student s) {
		double sfee=s.getStdFee();
		s.setDiscount(sfee * 5 /100.0);
		return dao.updateStudent(s);
	}

	public int deleteStudent(Integer id) {
		return dao.deleteStudent(id);
	}

	public Student getOneStudent(Integer id) {
		return dao.getOneStudent(id);
	}
	@Override
	public List<Student> getAllStudent() {
		List<Student> list= dao.getAllStudent();
		list=list.stream()
		.sorted((o1,o2)-> o2.getStdId()-o1.getStdId())
		.collect(Collectors.toList());
```

```
            return list;
      }

}
```

6. StudentRowmapper

```
package in.nit.mapper;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import in.nit.model.Student;

public class StudentRowMapper
      implements RowMapper<Student>
{
      @Override
      public Student mapRow(ResultSet rs, int rowNum) throws
SQLException {
            Student s=new Student();
            s.setStdId(rs.getInt("SID"));
            s.setStdName(rs.getString("SNAME"));
            s.setStdFee(rs.getDouble("SFEE"));
            s.setStdCourse(rs.getString("COURSE"));
            s.setDiscount(rs.getDouble("DISCOUNT"));
            return s;
      }

}
```

7. application.yml

```
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://localhost:3306/nit
    username: root
    password: root
```

8. Runner class code:

```
package in.nit.runner;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;
```

```
import in.nit.service.IStudentService;

@Component
public class JdbcTestRunner
	implements CommandLineRunner
{
	@Autowired
	private IStudentService service;

	public void run(String... args)
	throws Exception
	{
		/*
		System.out.println("from Runner");
		Student s=new Student();
		s.setStdId(10);
		s.setStdName("SAM");
		s.setStdFee(1200.0);
		s.setStdCourse("CORE JAVA");
		int count=service.updateStudent(s);
		int count=service.deleteStudent(10);
		if(count>0)
			System.out.println("Updated");
		else
			System.out.println("not Updated");
		 */

		/*
		 * Student s=service.getOneStudent(101);
		System.out.println(s);
		 */
		service.getAllStudent()
		.forEach(System.out::println);

	}

}
```


FB: https://www.facebook.com/groups/thejavatemple/
email : javabyraghu@gmail.com