

Spring Initializr

start.spring.io

Gmail Imagen YouTube Maps localhost:8080 / 12... Todos los marcadores

spring initializr

Project

☐ Gradle - Groovy

☐ Gradle - Kotlin

☒ Maven

Language

☒ Java

☐ Kotlin

☐ Groovy

Spring Boot

☐ 3.2.0 (SNAPSHOT)

☐ 3.2.0 (RC2)

☐ 3.1.6 (SNAPSHOT)

☒ 3.1.5

☐ 3.0.13 (SNAPSHOT)

☐ 3.0.12

☐ 2.7.18 (SNAPSHOT)

☐ 2.7.17

Project Metadata

Group

com.tutorialspoint

Artifact

SpringExample

Name

U6A1

Description

Demo project for Spring Boot

Package name

com.tutorialspoint.SpringExample

Packaging

☒ Jar

☐ War

Java

☐ 21

☒ 17

☐ 11

☐ 8

Dependencies

ADD ... CTRL + B

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

MySQL Driver

SQL

MySQL JDBC driver.

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

SpringExample main

Current File

Project

SpringExample C:\Users\Usuario\Desktop\2*DAMN\DDV

idea

mvn

src

main

java

com.tutorialspoint.SpringExample

U6A1Application

Student

Student

StudentDAO

resources

test

gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

External Libraries

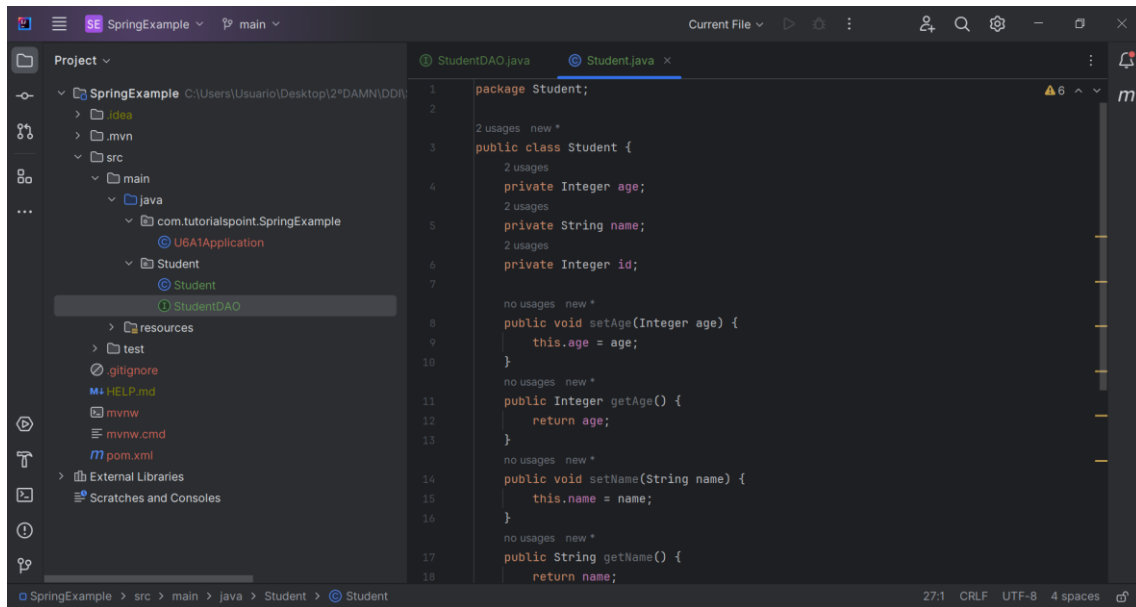
Scratches and Consoles

StudentDAO.java

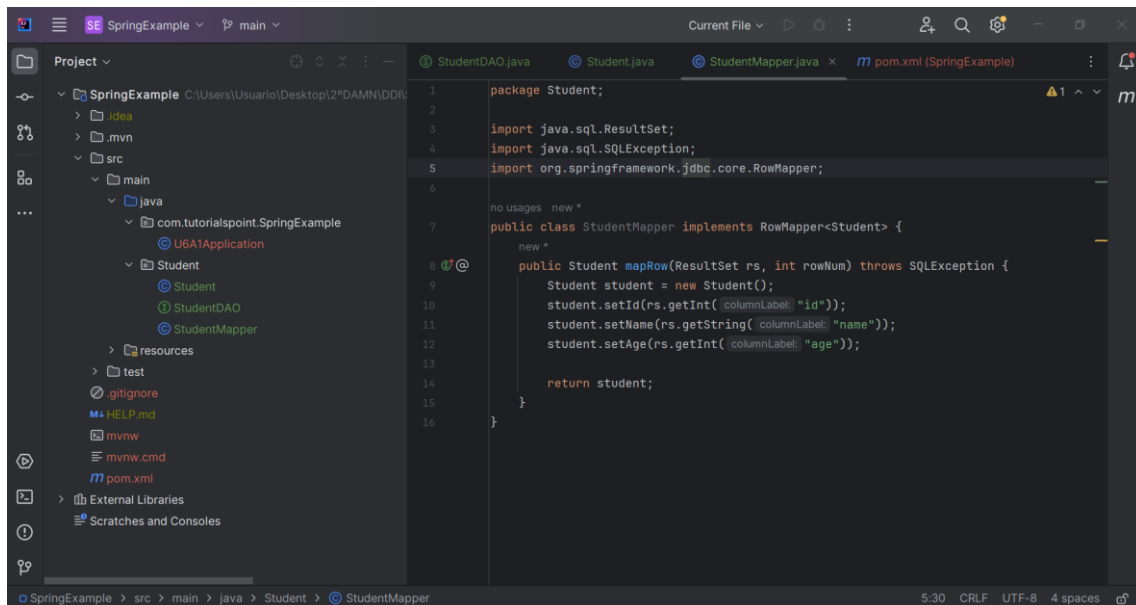
Student.java

```
1 package Student;
2
3 import java.util.List;
4 import javax.sql.DataSource;
5
6 no usages new *
7 public interface StudentDAO {
8
9     /**
10      * This is the method to be used to initialize
11      * database resources ie. connection.
12      */
13     no usages new *
14     public void setDataSource(DataSource ds);
15
16     /**
17      * This is the method to be used to create
18      * a record in the Student table.
19      */
20     no usages new *
21     public void create(String name, Integer age);
22
23     /**
24      * This is the method to be used to list down
25      * a record from the Student table corresponding
26      * to a passed student id.
```

SpringExample > src > main > java > Student > StudentDAO 6:18 CRLF UTF-8 4 spaces



```
1 package Student;
2
3 public class Student {
4     private Integer age;
5     private String name;
6     private Integer id;
7
8     public void setAge(Integer age) {
9         this.age = age;
10    }
11    public Integer getAge() {
12        return age;
13    }
14    public void setName(String name) {
15        this.name = name;
16    }
17    public String getName() {
18        return name;
19    }
20 }
```



```
1 package Student;
2
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import org.springframework.jdbc.core.RowMapper;
6
7 public class StudentMapper implements RowMapper<Student> {
8     @Override
9     public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
10         Student student = new Student();
11         student.setId(rs.getInt("id"));
12         student.setName(rs.getString("name"));
13         student.setAge(rs.getInt("age"));
14
15         return student;
16     }
17 }
```

```
15 public void create(String name, Integer age) {
16     String SQL = "insert into Student (name, age) values (?, ?)";
17     jdbcTemplateObject.update( SQL, name, age);
18     System.out.println("Created Record Name = " + name + " Age = " + age);
19     return;
20 }
21
22 public Student getStudent(Integer id) {
23     String SQL = "select * from Student where id = ?";
24     Student student = jdbcTemplateObject.queryForObject(SQL,
25         new Object[]{id}, new StudentMapper());
26     return student;
27 }
28
29 public List<Student> listStudents() {
30     String SQL = "select * from Student";
31     List<Student> students = jdbcTemplateObject.query(SQL, new StudentMapper());
32     return students;
33 }
34
35 public void delete(Integer id) {
36     String SQL = "delete from Student where id = ?";
37     jdbcTemplateObject.update(SQL, id);
38     System.out.println("Deleted Record with ID = " + id );
39 }
```

```
1 package com.tutorialspoint.SpringExample;
2 import java.util.List;
3
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.support.ClassPathXmlApplicationContext;
6 import com.tutorialspoint.SpringExample.StudentJDBCTemplate;
7
8 public class MainApp {
9     public static void main(String[] args) {
10         ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");
11
12         StudentJDBCTemplate studentJDBCTemplate =
13             (StudentJDBCTemplate)context.getBean( name: "studentJDBCTemplate");
14
15         System.out.println("-----Records Creation-----" );
16         studentJDBCTemplate.create( name: "Zara", age: 11);
17         studentJDBCTemplate.create( name: "Nuha", age: 2);
18         studentJDBCTemplate.create( name: "Ayan", age: 15);
19
20         System.out.println("-----Listing Multiple Records-----" );
21         List<Student> students = studentJDBCTemplate.listStudents();
22
23         for (Student record : students) {
24             System.out.print("ID : " + record.getId() );
25             System.out.print(", Name : " + record.getName() );
26             System.out.println(", Age : " + record.getAge());
27         }
28     }
29 }
```

