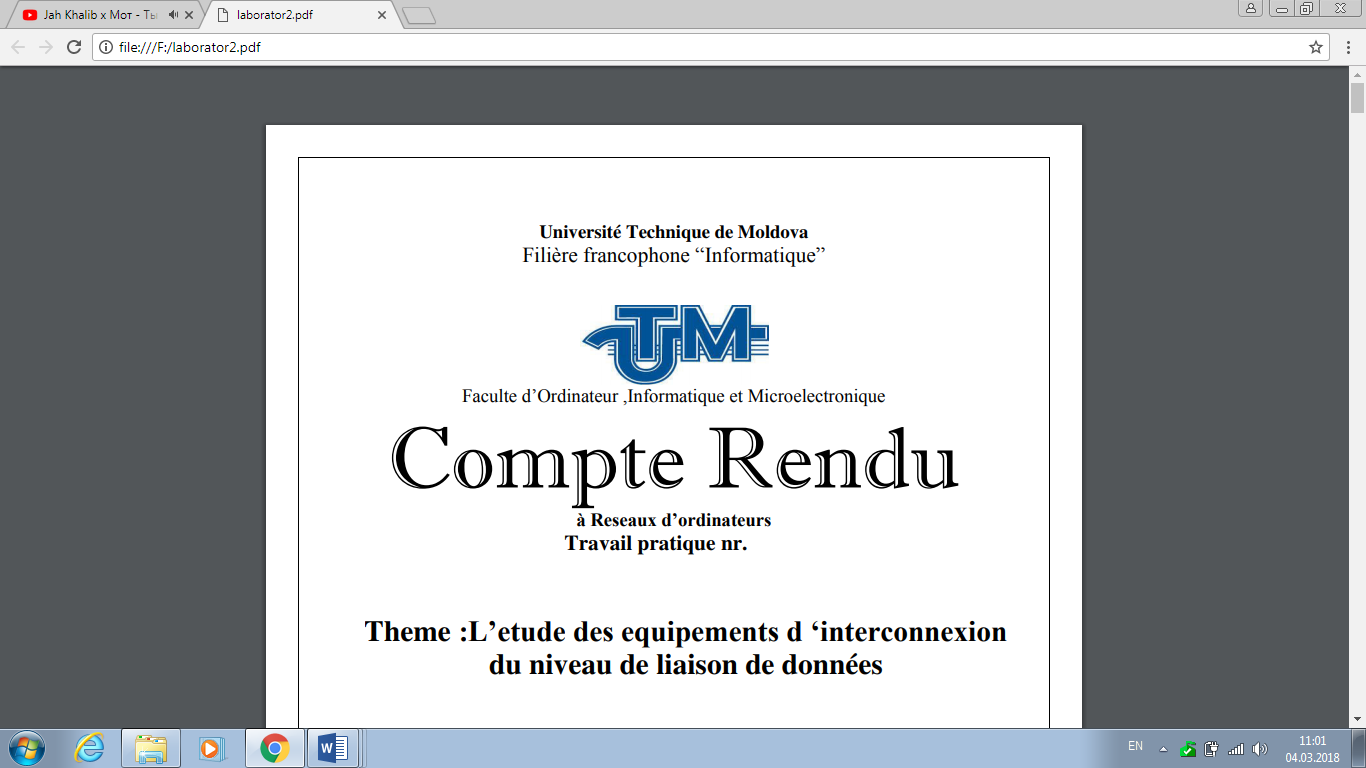
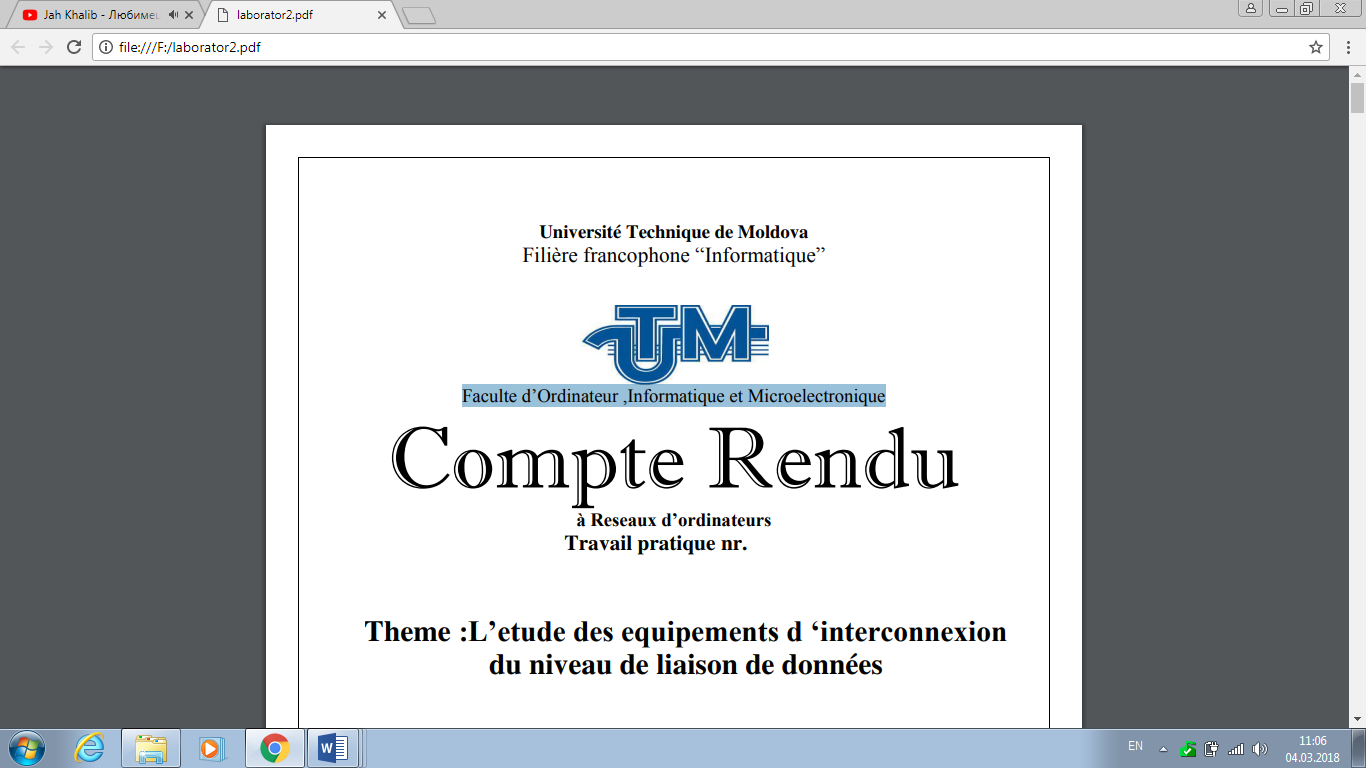
Ministère de l'éducation de la République de Moldova

Université technique de Moldavie

Faculté d’Ordinateur, Informatique et Microélectronique

Filière francophone “Informatique”





**OOP**

Travail pratique nr.5

**Thème : Spring project**

Effectué par l’étudiant(e) de gr FI-181 : Vrabie Teodor

Vérifié par le professeur : Eladii Vadim

Chișinau 2019

**Student.java**

public class Student {  
 private int studentId;  
 private String firstName;  
 private String lastName;  
 private String gender;  
 private String group;  
 private double average;  
  
 public Student(int studentId, String firstName, String lastName, String gender, String group, double average) {  
 this.studentId = studentId;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.gender = gender;  
 this.group = group;  
 this.average = average;  
 }  
  
 public int getStudentId() {  
 return studentId;  
 }  
  
 public void setStudentId(int studentId) {  
 this.studentId = studentId;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getGender() {  
 return gender;  
 }  
  
 public void setGender(String gender) {  
 this.gender = gender;  
 }  
  
 public String getGroup() {  
 return group;  
 }  
  
 public void setGroup(String group) {  
 this.group = group;  
 }  
  
 public double getAverage() {  
 return average;  
 }  
  
 public void setAverage(double average) {  
 this.average = average;  
 }  
}

**Teacher.java**

public class Teacher {  
 private int teacherId;  
 private String firstName;  
 private String lastName;  
 private String telephoneNumber;  
 private String taughtCourse;  
  
 public Teacher(int teacherId, String firstName, String lastName, String telephoneNumber, String taughtCourse) {  
 this.teacherId = teacherId;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.telephoneNumber = telephoneNumber;  
 this.taughtCourse = taughtCourse;  
 }  
  
 public int getTeacherId() {  
 return teacherId;  
 }  
  
 public void setTeacherId(int teacherId) {  
 this.teacherId = teacherId;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getTelephoneNumber() {  
 return telephoneNumber;  
 }  
  
 public void setTelephoneNumber(String telephoneNumber) {  
 this.telephoneNumber = telephoneNumber;  
 }  
  
 public String getTaughtCourse() {  
 return taughtCourse;  
 }  
  
 public void setTaughtCourse(String taughtCourse) {  
 this.taughtCourse = taughtCourse;  
 }

**Exam.java**

import java.util.Date;  
  
public class Exam {  
 private int examId;  
 private String examName;  
 private Date examDate;  
 private int examGrade;  
  
 public Exam(int examId, String examName, Date examDate, int examGrade) {  
 this.examId = examId;  
 this.examName = examName;  
 this.examDate = examDate;  
 this.examGrade = examGrade;  
 }  
  
 public int getExamId() {  
 return examId;  
 }  
  
 public void setExamId(int examId) {  
 this.examId = examId;  
 }  
  
 public String getExamName() {  
 return examName;  
 }  
  
 public void setExamName(String examName) {  
 this.examName = examName;  
 }  
  
 public Date getExamDate() {  
 return examDate;  
 }  
  
 public void setExamDate(Date examDate) {  
 this.examDate = examDate;  
 }  
  
 public int getExamGrade() {  
 return examGrade;  
 }  
  
 public void setExamGrade(int examGrade) {  
 this.examGrade = examGrade;  
 }  
}

**StudentWithExam.java**

import java.util.Date;

public class StudentWithExam extends Student {

private String examName;

private Date examDate;

private int examGrade;

public StudentWithExam(int studentId, String firstName, String lastName, String gender, String group, double average, String examName, Date examDate, int examGrade) {

super(studentId, firstName, lastName, gender, group, average);

this.examName = examName;

this.examDate = examDate;

this.examGrade = examGrade;

}

public StudentWithExam(int studentId, String firstName, String lastName, String gender, String group, double average, String examName) {

super(studentId, firstName, lastName, gender, group, average);

this.examName = examName;

}

public String getExamName() {

return examName;

}

public void setExamName(String examName) {

this.examName = examName;

}

public Date getExamDate() {

return examDate;

}

public void setExamDate(Date examDate) {

this.examDate = examDate;

}

public int getExamGrade() {

return examGrade;

}

public void setExamGrade(int examGrade) {

this.examGrade = examGrade;

}

}

**StudentRepository.java**

import com.school.project.classes.Student;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.jdbc.core.BeanPropertyRowMapper;  
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class StudentRepository {  
 @Autowired  
 private JdbcTemplate jdbcTemplate;  
  
 public List<Student> findAll() {  
 return jdbcTemplate.query(  
 "SELECT *\** FROM Student",  
 (reponse, rownumber) ->  
 new Student(reponse.getInt("studentid"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("gender"),  
 reponse.getString("group"),  
 reponse.getDouble("average")));  
 }  
  
 public Student findById(int studentId) {  
 String sql = "SELECT *\**FROM Student WHERE studentId = ?;";  
 Object[] inputs = new Object[] {studentId};  
  
 return jdbcTemplate.queryForObject(  
 sql,  
 inputs,  
 (reponse, rownumber) ->new Student (reponse.getInt("studentid"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("gender"),  
 reponse.getString("group"),  
 reponse.getDouble("average")));  
 }  
  
 public void save(Student student) {  
 jdbcTemplate.update(  
 "INSERT INTO Student(firstName, lastName, gender, \"group\", average) VALUES(?,?,?,?,?)",  
 student.getFirstName(), student.getLastName(), student.getGender(), student.getGroup(), student.getAverage());  
 }  
  
 public void update(Student student, String firstName) {  
 jdbcTemplate.update(  
 "UPDATE Student\n" +  
 "SET firstName = ?, lastName = ?,gender = ?,\"group\" = ?, average = ?\n" +  
 "WHERE firstName = ?",  
 student.getFirstName(), student.getLastName(), student.getGender(), student.getGroup(), student.getAverage(),firstName);  
 }  
  
 public void delete(String firstName) {  
 jdbcTemplate.update(  
 "DELETE FROM Student WHERE firstName = ?", firstName);  
 }  
  
  
}

**TeacherRepository.java**

import com.school.project.classes.Teacher;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class TeacherRepository {  
 @Autowired  
 private JdbcTemplate jdbcTemplate;  
  
 public List<Teacher> findAll() {  
 return jdbcTemplate.query(  
 "SELECT *\** FROM Teacher",  
 (reponse, rownumber) ->  
 new Teacher(reponse.getInt("teacherId"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("telephoneNumber"),  
 reponse.getString("taughtCourse")));  
 }  
  
 public Teacher findById(int teacherId) {  
 String sql = "SELECT *\**FROM Teacher WHERE teacherId = ?;";  
 Object[] inputs = new Object[] {teacherId};  
  
 return jdbcTemplate.queryForObject(  
 sql,  
 inputs,  
 (reponse, rownumber) ->new Teacher (reponse.getInt("teacherId"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("telephoneNumber"),  
 reponse.getString("taughtCourse")));  
 }  
  
  
 public void save(Teacher teacher) {  
 jdbcTemplate.update(  
 "INSERT INTO Teacher(firstName, lastName, telephoneNumber, taughtCourse) VALUES(?,?,?,?)",  
 teacher.getFirstName(), teacher.getLastName(), teacher.getTelephoneNumber(), teacher.getTaughtCourse());  
 }  
  
 public void update(Teacher teacher, String firstName) {  
 jdbcTemplate.update(  
 "UPDATE Teacher \n" +  
 "SET firstName = ?, lastName = ?,telephoneNumber = ?, taughtCourse = ?\n" +  
 "WHERE firstName = ?",  
 teacher.getFirstName(), teacher.getLastName(), teacher.getTelephoneNumber(), teacher.getTaughtCourse(),firstName);  
 }  
  
 public void delete(String firstName) {  
 jdbcTemplate.update(  
 "DELETE FROM Teacher WHERE firstName = ?", firstName);  
 }  
}

**ExamRepository.java**

import com.school.project.classes.Exam;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class ExamRepository {  
 @Autowired  
 JdbcTemplate jdbcTemplate;  
  
 public List<Exam> findAll() {  
 return jdbcTemplate.query(  
 "SELECT *\** FROM Exam",  
 (reponse, rownumber) ->  
 new Exam(reponse.getInt("examId"),  
 reponse.getString("examName"),  
 reponse.getDate("examDate"),  
 reponse.getInt("examGrade")));  
 }  
  
 public Exam findById(int examId) {  
 String sql = "SELECT *\**FROM Exam WHERE examId = ?;";  
 Object[] inputs = new Object[] {examId};  
  
 return jdbcTemplate.queryForObject(  
 sql,  
 inputs,  
 (reponse, rownumber) ->new Exam (reponse.getInt("examId"),  
 reponse.getString("examName"),  
 reponse.getDate("examDate"),  
 reponse.getInt("examGrade")));  
 }  
  
 public void save(Exam exam) {  
 jdbcTemplate.update(  
 "INSERT INTO Exam(examName, examDate, examGrade) VALUES(?,?,?)",  
 exam.getExamName(), exam.getExamDate(), exam.getExamGrade());  
 }  
  
 public void update(Exam exam, String examName) {  
 jdbcTemplate.update(  
 "UPDATE Exam \n" +  
 "SET examName = ?, examDate = ?,examGrade = ?\n" +  
 "WHERE examName = ?",  
 exam.getExamName(), exam.getExamDate(), exam.getExamGrade(),examName);  
 }  
  
 public void delete(String examName) {  
 jdbcTemplate.update(  
 "DELETE FROM Exam WHERE examName = ?", examName );  
 }  
}

**StudentWithExam.Repository.java**

import com.school.project.classes.Student;  
import com.school.project.classes.StudentWithExam;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.jdbc.core.BeanPropertyRowMapper;  
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class StudentWithExamRepository {  
 @Autowired  
 private JdbcTemplate jdbcTemplate;  
  
 public List<StudentWithExam> findAll() {  
  
 String sql = "SELECT Student.*\**, Exam.examName, Exam.examDate, Exam.examGrade FROM Student\n" +  
 "INNER JOIN Exam\n" +  
 "ON Exam.examId = Student.studentId;";  
  
 return jdbcTemplate.query(  
 sql,  
 (reponse, rownumber) ->  
 new StudentWithExam(reponse.getInt("studentid"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("gender"),  
 reponse.getString("group"),  
 reponse.getDouble("average"),  
 reponse.getString("examName"),  
 reponse.getDate("examDate"),  
 reponse.getInt("examGrade")));  
 }  
  
 public List<StudentWithExam> findByGenderAndGrade(String gender, int examGrade) {  
 String sql = "SELECT Student.*\**, Exam.examName, Exam.examDate, Exam.examGrade FROM Student\n" +  
 "INNER JOIN Exam\n" +  
 "ON Exam.examId = Student.studentId\n" +  
 "WHERE gender = ? AND examGrade = ?;";  
  
 Object[] inputs = new Object[] {gender, examGrade};  
  
 return jdbcTemplate.query(  
 sql,  
 inputs,  
 (reponse, rownumber) ->new StudentWithExam (reponse.getInt("studentid"),  
 reponse.getString("firstName"),  
 reponse.getString("lastname"),  
 reponse.getString("gender"),  
 reponse.getString("group"),  
 reponse.getDouble("average"),  
 reponse.getString("examName"),  
 reponse.getDate("examDate"),  
 reponse.getInt("examGrade")));  
 }  
  
}

**StudentCotroller.java**

import com.school.project.classes.Student;  
import com.school.project.repositories.StudentRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/school")  
public class StudentController {  
 @Autowired  
 StudentRepository studentRepository;  
  
 @GetMapping("student")  
 public List<Student> findAll() {  
 return studentRepository.findAll();  
 }  
  
 @GetMapping("student/{studentId}")  
 public Student findById(@PathVariable int studentId){  
 return studentRepository.findById(studentId);  
 }  
  
 @PostMapping("student")  
 public void save(@RequestBody Student student) {  
 studentRepository.save(student);  
 }  
  
 @PutMapping("student/{firstName}")  
 public void update(@RequestBody Student student, @PathVariable String firstName) {  
 studentRepository.update(student,firstName);  
 }  
  
 @DeleteMapping("student/{firstName}")  
 public void delete(@PathVariable String firstName) {  
 studentRepository.delete(firstName);  
 }  
}

**TeacherCotroller.java**

import com.school.project.classes.Teacher;  
import com.school.project.repositories.TeacherRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("school")  
public class TeacherController {  
 @Autowired  
 TeacherRepository teacherRepository;  
  
 @GetMapping("teacher")  
 public List<Teacher> findAll() {  
 return teacherRepository.findAll();  
 }  
  
 @GetMapping("teacher/{teacherId}")  
 public Teacher findById(@PathVariable int teacherId){  
 return teacherRepository.findById(teacherId);  
 }  
  
 @PostMapping("teacher")  
 public void save(@RequestBody Teacher teacher) {  
 teacherRepository.save(teacher);  
 }  
  
 @PutMapping("teacher/{firstName}")  
 public void update(@RequestBody Teacher teacher, @PathVariable String firstName) {  
 teacherRepository.update(teacher,firstName);  
 }  
  
 @DeleteMapping("teacher/{firstName}")  
 public void delete(@PathVariable String firstName) {  
 teacherRepository.delete(firstName);  
 }  
}

**ExamCotroller.java**

@RestController  
@RequestMapping("school")  
public class ExamController {  
 @Autowired  
 ExamRepository examRepository;  
  
 @GetMapping("exam")  
 public List<Exam> findAll() {  
 return examRepository.findAll();  
 }  
  
 @GetMapping("exam/{examId}")  
 public Exam findById(@PathVariable int examId){  
 return examRepository.findById(examId);  
 }  
  
 @PostMapping("exam")  
 public void save(@RequestBody Exam exam) {  
 examRepository.save(exam);  
 }  
  
 @PutMapping("exam/{examName}")  
 public void update(@RequestBody Exam exam, @PathVariable String examName) {  
 examRepository.update(exam,examName);  
 }  
  
 @DeleteMapping("exam/{examName}")  
 public void delete(@PathVariable String examName) {  
 examRepository.delete(examName);  
 }  
}

**StudentWithExamCotroller.java**

import com.school.project.classes.StudentWithExam;  
import com.school.project.repositories.StudentWithExamRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/school")  
public class StudentWithExamController {  
 @Autowired  
 StudentWithExamRepository studentWithExamRepository;  
  
 @GetMapping("studentWithExam")  
 public List<StudentWithExam> findAll() {  
 return studentWithExamRepository.findAll();  
 }  
  
 @GetMapping("studentGrade")  
 public List<StudentWithExam> findByGenderAndGrade(@RequestParam String gender, @RequestParam int examGrade){  
 return studentWithExamRepository.findByGenderAndGrade(gender,examGrade);  
 }  
  
}

**SwaggerConfig.java**

import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;  
import springfox.documentation.builders.PathSelectors;  
import springfox.documentation.builders.RequestHandlerSelectors;  
import springfox.documentation.spi.DocumentationType;  
import springfox.documentation.spring.web.plugins.Docket;  
import springfox.documentation.swagger2.annotations.EnableSwagger2;  
  
@Configuration  
@EnableSwagger2  
public class SwaggerConfig {  
 @Bean  
 public Docket api() {  
 return new Docket(DocumentationType.*SWAGGER\_2*)  
 .select()  
 .apis(RequestHandlerSelectors.*basePackage*("com.school.project"))  
 .paths(PathSelectors.*any*())  
 .build();  
 }  
}

**Database.sql**

CREATE DATABASE schoolRelational;  
CREATE TABLE Student  
(  
 studentId serial primary key,  
 firstName varchar(15) not null,  
 lastName varchar(15) not null,  
 gender varchar(15) not null,  
 "group" varchar(15) not null,  
 average float not null  
);  
CREATE TABLE Teacher  
(  
 teacherId serial primary key,  
 firstName varchar(15) not null,  
 lastName varchar(15) not null,  
 telephoneNumber varchar(15) not null,  
 taughtCourse varchar(15) not null  
);  
  
CREATE TABLE Exam  
(  
 examId serial primary key,  
 examName varchar(15) not null,  
 examDate date not null,  
 examGrade int not null,  
 studentId int UNIQUE ,  
 FOREIGN KEY (studentId) REFERENCES Student (studentId)  
);  
  
INSERT INTO Student(firstName, lastName, gender, "group", average) VALUES ('Teodor','Vrabie','male','Fi-181',9);  
INSERT INTO Student(firstName, lastName, gender, "group", average) VALUES ('Andrei','Lovru','male','Fi-181',6);  
INSERT INTO Exam(examName, examDate, examGrade,studentId) VALUES ('informatica','2019-06-01',10,2);  
INSERT INTO Teacher(firstName, lastName, telephoneNumber, taughtCourse) VALUES ('Anatolie','Balabanov','079897754','CDE');  
INSERT INTO Exam(examName, examDate, examGrade,studentId) VALUES ('matematica','2019-06-01',10,1);

**Pom.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:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.2.2.RELEASE</version>  
 <relativePath/> *<!-- lookup parent from repository -->* </parent>  
 <groupId>com.school</groupId>  
 <artifactId>project</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>project</name>  
 <description>Demo project for school</description>  
  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>io.springfox</groupId>  
 <artifactId>springfox-swagger2</artifactId>  
 <version>2.9.2</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>io.springfox</groupId>  
 <artifactId>springfox-swagger-ui</artifactId>  
 <version>2.9.2</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-jdbc</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.postgresql</groupId>  
 <artifactId>postgresql</artifactId>  
 <scope>runtime</scope>  
 </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>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>