МИНИСТЕРСТВО ТРАНСПОРТА РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»

Кафедра ЦТУТП

**Отчёт**

По лабораторной работе №7  
по дисциплине «Корпоративные информационные системы»

Тема: «Разработка REST-контроллера»

Вариант №15

Выполнил: Дронов А. А.

Группа: УИС-411

Преподаватель: доц. Кафедры ЦТУТП

Козьяков П. О.

­

Москва 2024 г.

**СОДЕРЖАНИЕ**

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc180325218)

[ХОД ВЫПОЛНЕНИЯ РАБОТЫ 4](#_Toc180325219)

[РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ 5](#_Toc180325220)

[КОД ПРОГРАММЫ 6](#_Toc180325221)

[ВЫВОД 10](#_Toc180325222)

# **ПОСТАНОВКА ЗАДАЧИ**

Для данных по университетам, выборка из БД которых реализована в лабораторной работе №6, реализовать контроллер на все операции Добавления, Удаления, Изменения и Выборки данных. Для документации и проверки запросов использовать Swagger.

# **ХОД ВЫПОЛНЕНИЯ РАБОТЫ**

Разработано приложение для управления данными об университетах, реализующее многослойную архитектуру на основе Spring Boot, MyBatis и Swagger. Модель University описывает сущность университета, включая такие поля, как идентификатор, название, местоположение, количество студентов и статус (государственный или частный). Аннотации Jackson используются для преобразования данных между Java-объектами и JSON-форматом.

Интерфейс UniversityMapper предоставляет методы для выполнения операций с базой данных PostgreSQL: добавление, обновление, выборка всех университетов и удаление записи по идентификатору. SQL-запросы описаны с использованием аннотаций MyBatis, а XML-файл маппинга содержит resultMap для сопоставления столбцов базы данных и атрибутов модели.

Сервис UniversityService отвечает за бизнес-логику и является промежуточным слоем между контроллером и маппером. Он реализует методы для добавления, обновления, получения и удаления данных об университетах.

Контроллер UniversityController предоставляет REST API для работы с данными. С использованием Swagger добавлено документирование всех операций. API поддерживает:

* **POST** запросы для добавления университетов.
* **PUT** запросы для обновления данных.
* **GET** запросы для получения всех записей.
* **DELETE** запросы для удаления записи по идентификатору.

Настройки для подключения к PostgreSQL, включая логирование запросов, описаны в файле application.properties. Приложение работает на порту 8081, предоставляя удобный доступ к сервису.

# **РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

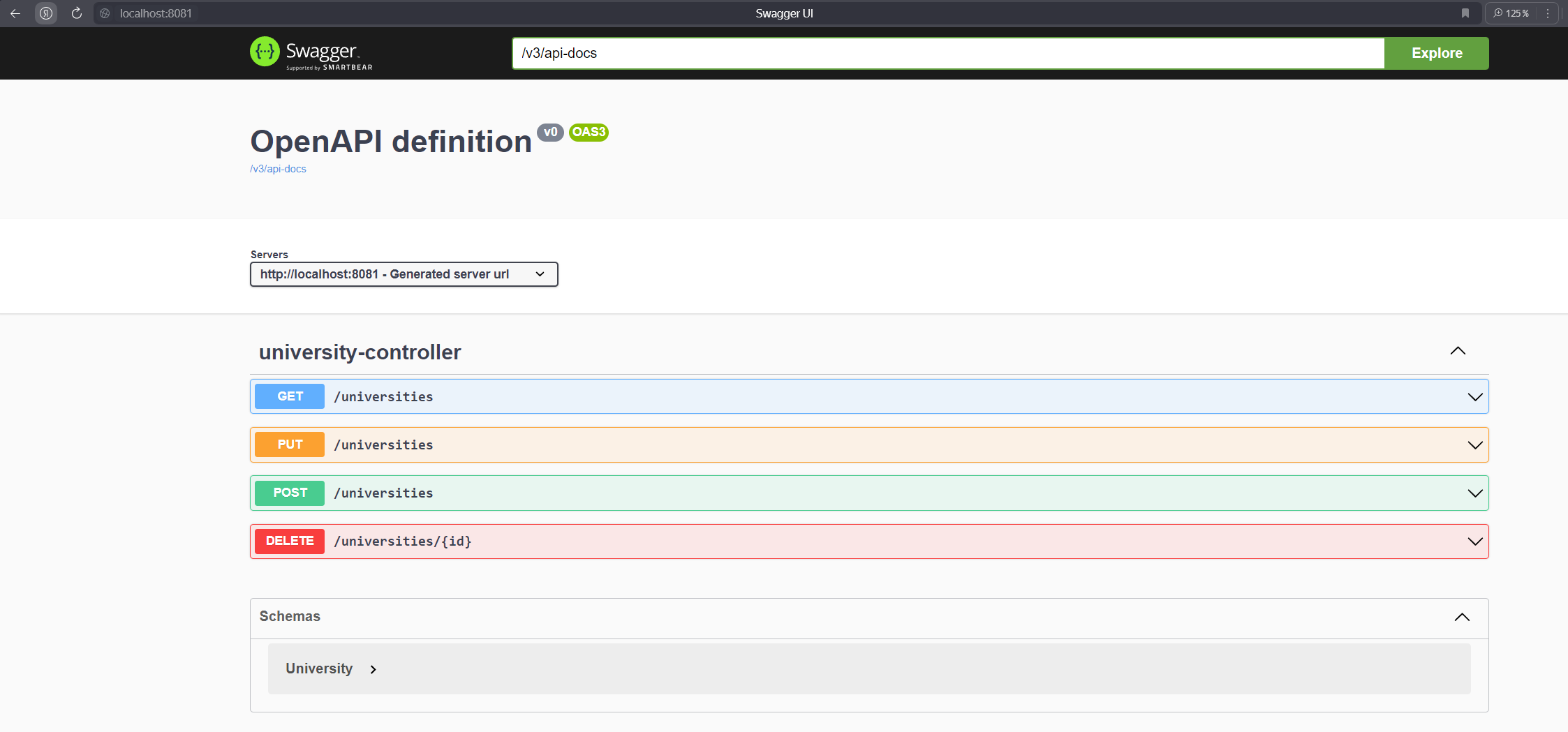


Рисунок 1 – Результат работы программы

# **КОД ПРОГРАММЫ**

package com.labsfrom6to8.Model;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class University {  
  
 private Long id;  
  
 @JsonProperty("name")  
 private String name;  
  
 @JsonProperty("location")  
 private String location;  
  
 @JsonProperty("student\_count")  
 private Integer studentCount;  
  
 @JsonProperty("is\_public")  
 private Boolean isPublic;  
}

package com.labsfrom6to8.Mapper;  
  
import com.labsfrom6to8.Model.University;  
import org.apache.ibatis.annotations.\*;  
  
import java.util.List;  
  
@Mapper  
public interface UniversityMapper {  
  
 @Insert("INSERT INTO universities (name, location, student\_count, is\_public) VALUES (#{name}, #{location}, #{studentCount}, #{isPublic})")  
 void insertUniversity(University university);  
  
 @Update("UPDATE universities SET name = #{name}, location = #{location}, student\_count = #{studentCount}, is\_public = #{isPublic} WHERE id = #{id}")  
 void updateUniversity(University university);  
  
 @Select("SELECT \* FROM universities")  
 @Results({  
 @Result(property = "studentCount", column = "student\_count"),  
 @Result(property = "isPublic", column = "is\_public")  
 })  
 List<University> selectAllUniversities();  
  
 @Delete("DELETE FROM universities WHERE id = #{id}")  
 void deleteUniversity(Long id);  
}

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
  
<mapper namespace="com.example.Mapper.UniversityMapper">  
  
 <insert id="insertUniversity" parameterType="com.example.Model.University">  
 INSERT INTO universities (name, location, student\_count, is\_public)  
 VALUES (#{name}, #{location}, #{studentCount}, #{isPublic})  
 </insert>  
  
 <update id="updateUniversity" parameterType="com.example.Model.University">  
 UPDATE universities  
 SET name = #{name},  
 location = #{location},  
 student\_count = #{studentCount},  
 is\_public = #{isPublic}  
 WHERE id = #{id}  
 </update>  
  
 <resultMap id="UniversityResultMap" type="com.example.Model.University">  
 <result property="id" column="id"/>  
 <result property="name" column="name"/>  
 <result property="location" column="location"/>  
 <result property="studentCount" column="student\_count"/>  
 <result property="isPublic" column="is\_public"/>  
 </resultMap>  
  
 <select id="selectAllUniversities" resultMap="UniversityResultMap">  
 SELECT \* FROM universities  
 </select>  
  
 <delete id="deleteUniversity" parameterType="Long">  
 DELETE FROM universities  
 WHERE id = #{id}  
 </delete>  
</mapper>

package com.labsfrom6to8.Service;  
  
import com.labsfrom6to8.Mapper.UniversityMapper;  
import com.labsfrom6to8.Model.University;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class UniversityService {  
  
 private final UniversityMapper universityMapper;  
  
 public UniversityService(UniversityMapper universityMapper) {  
 this.universityMapper = universityMapper;  
 }  
  
 public void addUniversity(University university) {  
 universityMapper.insertUniversity(university);  
 }  
  
 public void updateUniversity(University university) {  
 universityMapper.updateUniversity(university);  
 }  
  
 public List<University> getAllUniversities() {  
 return universityMapper.selectAllUniversities();  
 }  
  
 public void deleteUniversityById(Long id) {  
 universityMapper.deleteUniversity(id);  
 }  
}

package com.labsfrom6to8.Controller;

import com.labsfrom6to8.Model.University;

import com.labsfrom6to8.Service.UniversityService;

import io.swagger.v3.oas.annotations.Operation;

import io.swagger.v3.oas.annotations.tags.Tag;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/universities")

@Tag(name = "University API", description = "API для управления университетами")

public class UniversityController {

private final UniversityService universityService;

public UniversityController(UniversityService universityService) {

this.universityService = universityService;

}

@PostMapping

@Operation(summary = "Добавить новый университет", description = "Добавляет новый университет в систему")

public void addUniversity(@RequestBody University university) {

universityService.addUniversity(university);

}

@PutMapping

@Operation(summary = "Обновить университет", description = "Обновляет информацию об университете")

public void updateUniversity(@RequestBody University university) {

universityService.updateUniversity(university);

}

@GetMapping

@Operation(summary = "Получить список всех университетов", description = "Возвращает список всех университетов в системе")

public List<University> getAllUniversities() {

return universityService.getAllUniversities();

}

@DeleteMapping("/{id}")

@Operation(summary = "Удалить университет", description = "Удаляет университет по его идентификатору")

public void deleteUniversity(@PathVariable Long id) {

universityService.deleteUniversityById(id);

}

}

# **ВЫВОД**

Приложение представляет собой полнофункциональный инструмент для управления данными об университетах. Использование Swagger упрощает документирование и тестирование API, а модульная структура приложения делает его легко поддерживаемым и масштабируемым. REST API обеспечивает простую интеграцию с другими системами, а использование MyBatis позволяет эффективно работать с базой данных.