МИНИСТЕРСТВО ТРАНСПОРТА РФ

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

Кафедра ЦТУТП

**Отчёт**

По лабораторной работе №6  
по дисциплине «Корпоративные информационные системы»

Тема: «Работа с библиотекой MyBatis»

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 9](#_Toc180325222)

# **ПОСТАНОВКА ЗАДАЧИ**

Описать класс Университет. В БД Postgres создать таблицу «университеты». С помощью MyBatis реализовать операции Добавления, Удаления, Изменения и Выборки данных из таблицы «университеты». Продемонстрировать операции работы с БД.

# **ХОД ВЫПОЛНЕНИЯ РАБОТЫ**

Разработано приложение для управления данными о университетах с использованием Spring Boot и MyBatis. На уровне модели создан класс University, который содержит такие свойства, как идентификатор, название университета, местоположение, количество студентов и публичный статус. Этот класс настроен для преобразования в JSON с использованием аннотаций Jackson.

Интерфейс UniversityMapper определяет методы для выполнения CRUD-операций. Аннотации MyBatis используются для описания SQL-запросов, таких как вставка, обновление, выборка всех записей и удаление по идентификатору. В XML-файле маппинга добавлены настройки resultMap для отображения столбцов базы данных на свойства модели.

Сервис UniversityService содержит бизнес-логику и является связующим звеном между контроллером и маппером. Он предоставляет методы для добавления, обновления, получения списка университетов и удаления записей.

Контроллер UniversityController реализует REST API для взаимодействия с клиентами. Пользователь может отправлять POST-запросы для добавления новых университетов, PUT-запросы для их обновления, GET-запросы для получения списка всех университетов и DELETE-запросы для удаления записи по идентификатору.

Приложение настроено на подключение к базе данных PostgreSQL. Логирование запросов включено через настройки MyBatis для упрощения отладки. Сервис работает на порту 8081, предоставляя доступ к API через HTTP-запросы.

# **РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

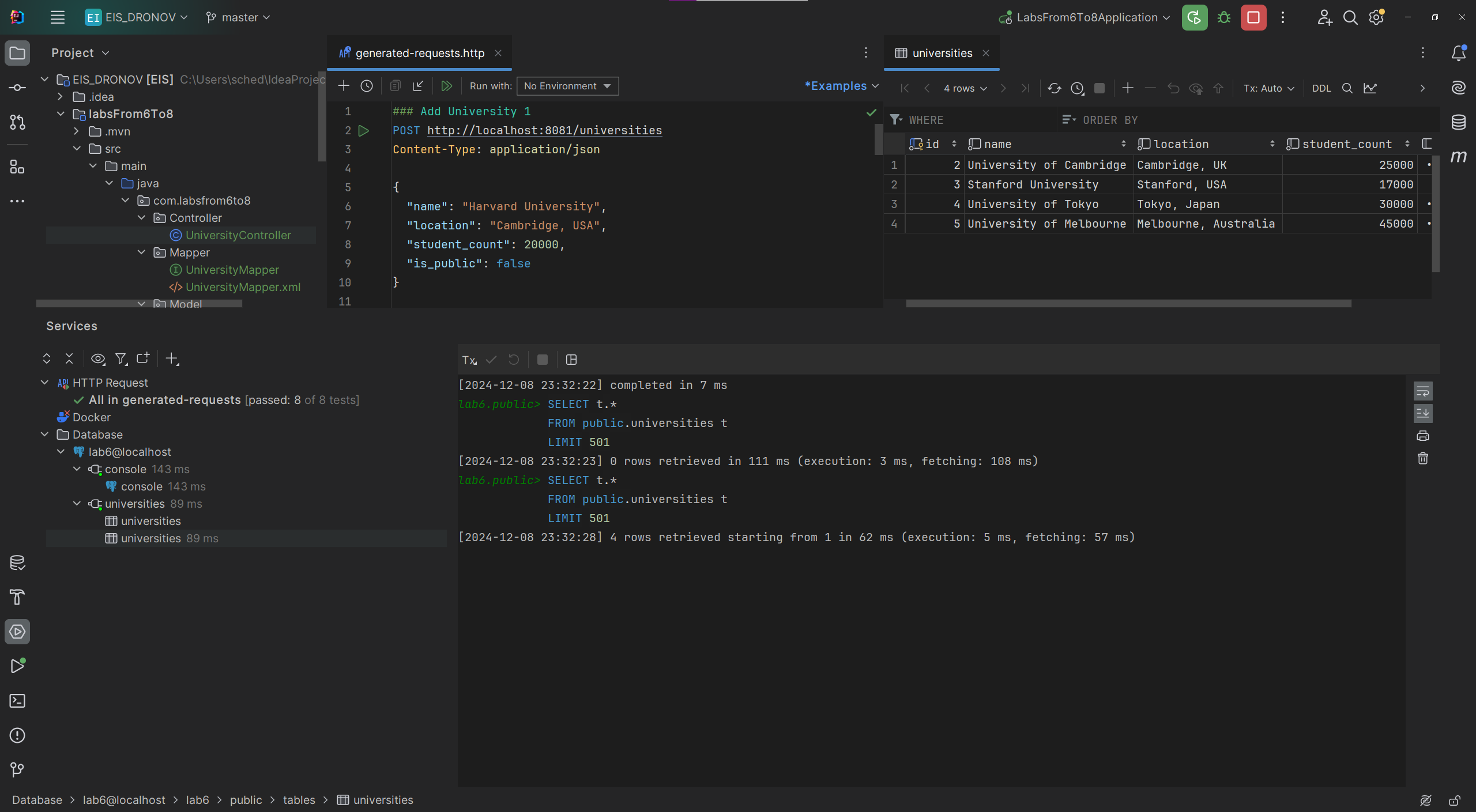


Рисунок 1 – Результат работы программы

# **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 org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/universities")  
public class UniversityController {  
  
 private final UniversityService universityService;  
  
 public UniversityController(UniversityService universityService) {  
 this.universityService = universityService;  
 }  
  
 @PostMapping  
 public void addUniversity(@RequestBody University university) {  
 universityService.addUniversity(university);  
 }  
  
 @PutMapping  
 public void updateUniversity(@RequestBody University university) {  
 universityService.updateUniversity(university);  
 }  
  
 @GetMapping  
 public List<University> getAllUniversities() {  
 return universityService.getAllUniversities();  
 }  
  
 @DeleteMapping("/{id}")  
 public void deleteUniversity(@PathVariable Long id) {  
 universityService.deleteUniversityById(id);  
 }  
}

spring.application.name=labsFrom6To8  
  
spring.datasource.url=jdbc:postgresql://localhost:5432/lab6  
spring.datasource.username=postgres  
spring.datasource.password=123  
spring.datasource.driver-class-name=org.postgresql.Driver  
  
logging.level.org.mybatis=DEBUG  
logging.level.org.springframework.jdbc.core=DEBUG  
logging.level.org.springframework.jdbc.datasource=DEBUG  
logging.level.org.apache.ibatis.logging.stdout.StdOutImpl=TRACE  
mybatis.configuration.log-impl=org.apache.ibatis.logging.stdout.StdOutImpl  
  
spring.mvc.view.prefix=  
spring.mvc.view.suffix=  
server.port=8081

# **ВЫВОД**

Приложение успешно реализует функционал управления данными о университетах, обеспечивая гибкость и удобство работы с базой данных. Архитектура приложения соответствует принципам модульности, что упрощает его поддержку и масштабирование. REST API делает сервис удобным для интеграции с внешними системами и приложениями.