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 №12

Выполнил: Бугаева А. Л.

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 6](#_Toc180325220)

[КОД ПРОГРАММЫ 7](#_Toc180325221)

[ВЫВОД 11](#_Toc180325222)

# **ПОСТАНОВКА ЗАДАЧИ**

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

# **ХОД ВЫПОЛНЕНИЯ РАБОТЫ**

Для управления данными о компьютерах был разработан сервис на основе Spring Boot с использованием MyBatis. Класс Computer определяет модель данных, включающую информацию о бренде, модели, процессоре, объёме оперативной памяти, объёме хранилища и доступности компьютера. Реализация операций с базой данных выполнена с использованием интерфейса ComputerMapper, который содержит методы для добавления, обновления, выборки и удаления записей. Методы связываются с SQL-запросами через аннотации MyBatis.

Для управления запросами к базе данных также используется XML-файл, включающий маппинг результата и SQL-запросы. Слой сервиса ComputerService обрабатывает бизнес-логику и взаимодействует с маппером. Контроллер ComputerController предоставляет REST API для работы с данными через HTTP-запросы.

Подключение к базе данных PostgreSQL осуществляется с использованием настроек в файле application.properties. Логирование запросов MyBatis включено для удобства отладки. Сервис запускается на порту 8081, обеспечивая CRUD-операции с данными о компьютерах.

# **РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

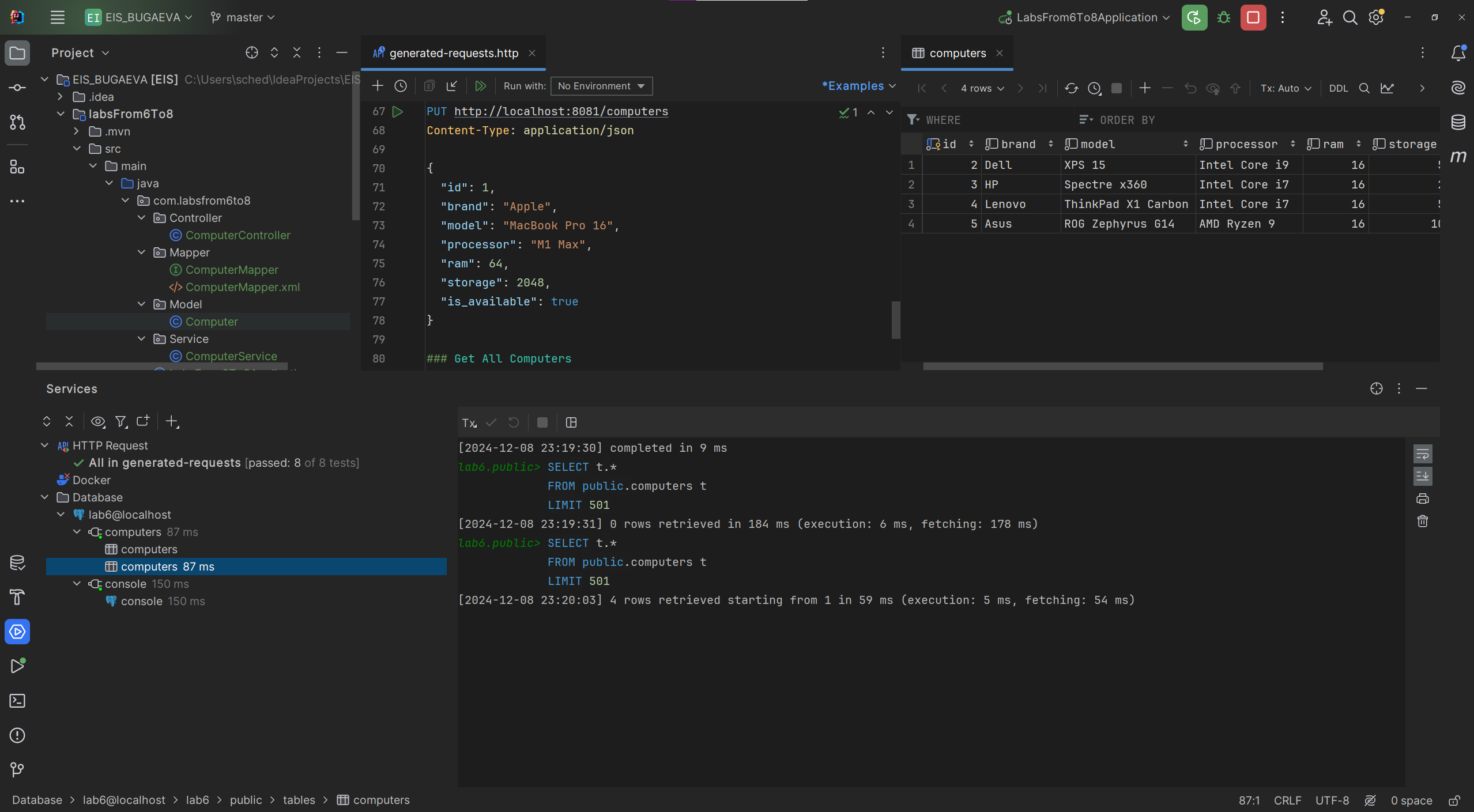


Рисунок 1 – Результат работы программы

# **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 Computer {  
  
 private Long id;  
  
 @JsonProperty("brand")  
 private String brand;  
  
 @JsonProperty("model")  
 private String model;  
  
 @JsonProperty("processor")  
 private String processor;  
  
 @JsonProperty("ram")  
 private Integer ram;  
  
 @JsonProperty("storage")  
 private Integer storage;  
  
 @JsonProperty("is\_available")  
 private Boolean isAvailable;  
}

package com.labsfrom6to8.Mapper;  
  
import com.labsfrom6to8.Model.Computer;  
import org.apache.ibatis.annotations.\*;  
  
import java.util.List;  
  
@Mapper  
public interface ComputerMapper {  
  
 @Insert("INSERT INTO computers (brand, model, processor, ram, storage, is\_available) VALUES (#{brand}, #{model}, #{processor}, #{ram}, #{storage}, #{isAvailable})")  
 void insertComputer(Computer computer);  
  
 @Update("UPDATE computers SET brand = #{brand}, model = #{model}, processor = #{processor}, ram = #{ram}, storage = #{storage}, is\_available = #{isAvailable} WHERE id = #{id}")  
 void updateComputer(Computer computer);  
  
 @Select("SELECT \* FROM computers")  
 @Results({  
 @Result(property = "isAvailable", column = "is\_available")  
 })  
 List<Computer> selectAllComputers();  
  
 @Delete("DELETE FROM computers WHERE id = #{id}")  
 void deleteComputer(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.ComputerMapper">  
  
 <insert id="insertComputer" parameterType="com.example.Model.Computer">  
 INSERT INTO computers (brand, model, processor, ram, storage, is\_available)  
 VALUES (#{brand}, #{model}, #{processor}, #{ram}, #{storage}, #{isAvailable})  
 </insert>  
  
 <update id="updateComputer" parameterType="com.example.Model.Computer">  
 UPDATE computers  
 SET brand = #{brand},  
 model = #{model},  
 processor = #{processor},  
 ram = #{ram},  
 storage = #{storage},  
 is\_available = #{isAvailable}  
 WHERE id = #{id}  
 </update>  
  
 <resultMap id="ComputerResultMap" type="com.example.Model.Computer">  
 <result property="id" column="id"/>  
 <result property="brand" column="brand"/>  
 <result property="model" column="model"/>  
 <result property="processor" column="processor"/>  
 <result property="ram" column="ram"/>  
 <result property="storage" column="storage"/>  
 <result property="isAvailable" column="is\_available"/>  
 </resultMap>  
  
 <select id="selectAllComputers" resultMap="ComputerResultMap">  
 SELECT \* FROM computers  
 </select>  
  
 <delete id="deleteComputer" parameterType="Long">  
 DELETE FROM computers  
 WHERE id = #{id}  
 </delete>  
</mapper>

package com.labsfrom6to8.Service;  
  
import com.labsfrom6to8.Mapper.ComputerMapper;  
import com.labsfrom6to8.Model.Computer;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class ComputerService {  
  
 private final ComputerMapper computerMapper;  
  
 public ComputerService(ComputerMapper computerMapper) {  
 this.computerMapper = computerMapper;  
 }  
  
 public void addComputer(Computer computer) {  
 computerMapper.insertComputer(computer);  
 }  
  
 public void updateComputer(Computer computer) {  
 computerMapper.updateComputer(computer);  
 }  
  
 public List<Computer> getAllComputers() {  
 return computerMapper.selectAllComputers();  
 }  
  
 public void deleteComputerById(Long id) {  
 computerMapper.deleteComputer(id);  
 }  
}

package com.labsfrom6to8.Controller;  
  
import com.labsfrom6to8.Model.Computer;  
import com.labsfrom6to8.Service.ComputerService;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/computers")  
public class ComputerController {  
  
 private final ComputerService computerService;  
  
 public ComputerController(ComputerService computerService) {  
 this.computerService = computerService;  
 }  
  
 @PostMapping  
 public void addComputer(@RequestBody Computer computer) {  
 computerService.addComputer(computer);  
 }  
  
 @PutMapping  
 public void updateComputer(@RequestBody Computer computer) {  
 computerService.updateComputer(computer);  
 }  
  
 @GetMapping  
 public List<Computer> getAllComputers() {  
 return computerService.getAllComputers();  
 }  
  
 @DeleteMapping("/{id}")  
 public void deleteComputer(@PathVariable Long id) {  
 computerService.deleteComputerById(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**

Разработанное приложение предоставляет функциональный сервис для работы с данными о компьютерах. Использование Spring Boot и MyBatis обеспечивает модульную архитектуру, упрощающую поддержку и расширение. REST API позволяет легко интегрировать сервис с другими системами. Решение демонстрирует корректную работу с базой данных PostgreSQL и удобство использования благодаря чёткой структуре кода.