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 №14

Выполнил: Сафонов Г. К.

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**

Для данных по населённым пунктам, 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**

Создано приложение для управления данными о населённых пунктах с использованием Spring Boot и MyBatis. Модель Settlement описывает сущность населённого пункта с такими атрибутами, как идентификатор, название, численность населения, регион и статус активности. Аннотации Jackson позволяют преобразовывать данные между объектами Java и JSON-форматом.

Интерфейс SettlementMapper предоставляет методы для выполнения операций с базой данных, таких как добавление, обновление, выборка активных населённых пунктов и удаление по идентификатору. SQL-запросы описаны с помощью аннотаций MyBatis, а XML-маппинг используется для отображения результатов запросов в объекты модели.

Слой сервиса SettlementService инкапсулирует бизнес-логику приложения, включая обработку данных перед их передачей в SettlementMapper или возвращением результата из него. Реализованы методы для добавления, обновления, получения и удаления записей.

Контроллер SettlementController предоставляет REST API для взаимодействия с клиентами. С помощью Swagger аннотаций (@Tag и @Operation) добавлено документирование API, что упрощает работу разработчиков и тестирование приложения. API поддерживает POST-запросы для добавления населённого пункта, PUT-запросы для обновления данных, GET-запросы для получения активных записей и DELETE-запросы для удаления по идентификатору.

Конфигурация подключения к базе данных PostgreSQL выполнена в файле application.properties. Включено логирование SQL-запросов для отладки и анализа. Приложение запускается на порту 8081.

# **РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

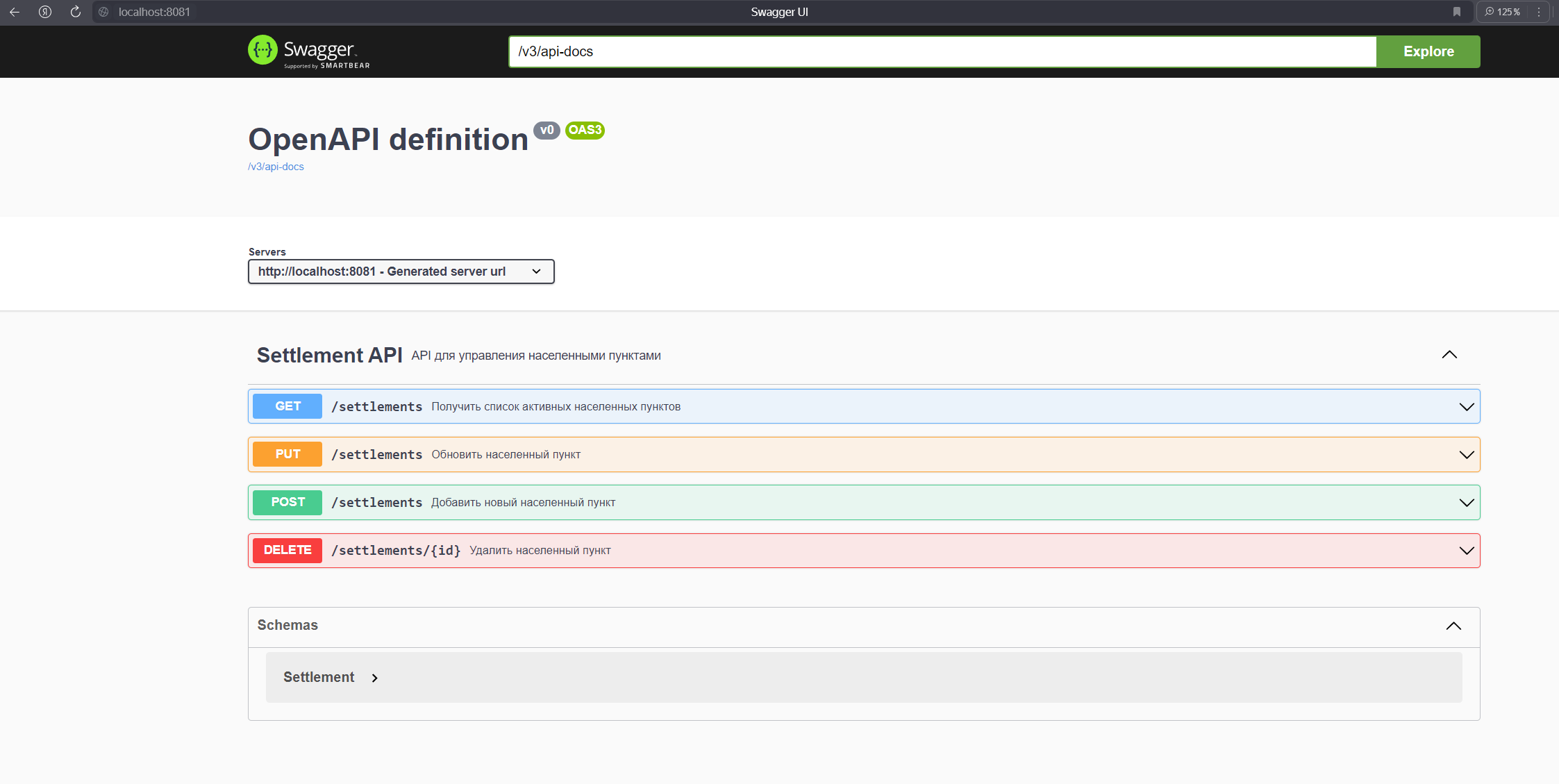


Рисунок 1 – Результат работы программы

# **КОД ПРОГРАММЫ**

package com.labsfrom6to8.Model;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Settlement {  
  
 private Long id;  
  
 @JsonProperty("name")  
 private String name;  
  
 @JsonProperty("population")  
 private Integer population;  
  
 @JsonProperty("region")  
 private String region;  
  
 @JsonProperty("is\_active")  
 private Boolean isActive;  
}

package com.labsfrom6to8.Mapper;  
  
import com.labsfrom6to8.Model.Settlement;  
import org.apache.ibatis.annotations.\*;  
  
import java.util.List;  
  
@Mapper  
public interface SettlementMapper {  
  
 @Insert("INSERT INTO settlements (name, population, region, is\_active) VALUES (#{name}, #{population}, #{region}, #{isActive})")  
 void insertSettlement(Settlement settlement);  
  
 @Update("UPDATE settlements SET name = #{name}, population = #{population}, region = #{region}, is\_active = #{isActive} WHERE id = #{id}")  
 void updateSettlement(Settlement settlement);  
  
 @Select("SELECT \* FROM settlements WHERE is\_active = TRUE")  
 List<Settlement> selectActiveSettlements();  
  
 @Delete("DELETE FROM settlements WHERE id = #{id}")  
 void deleteSettlement(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.SettlementMapper">  
  
 <insert id="insertSettlement" parameterType="com.example.Model.Settlement">  
 INSERT INTO settlements (name, population, region, is\_active)  
 VALUES (#{name}, #{population}, #{region}, #{isActive})  
 </insert>  
  
 <update id="updateSettlement" parameterType="com.example.Model.Settlement">  
 UPDATE settlements  
 SET name = #{name},  
 population = #{population},  
 region = #{region},  
 is\_active = #{isActive}  
 WHERE id = #{id}  
 </update>  
  
 <resultMap id="SettlementResultMap" type="com.example.Model.Settlement">  
 <result property="id" column="id"/>  
 <result property="name" column="name"/>  
 <result property="population" column="population"/>  
 <result property="region" column="region"/>  
 <result property="isActive" column="is\_active"/>  
 </resultMap>  
  
 <select id="selectActiveSettlements" resultMap="SettlementResultMap">  
 SELECT \* FROM settlements WHERE is\_active = TRUE  
 </select>  
  
 <delete id="deleteSettlement" parameterType="Long">  
 DELETE FROM settlements  
 WHERE id = #{id}  
 </delete>  
</mapper>

package com.labsfrom6to8.Service;  
  
  
import com.labsfrom6to8.Mapper.SettlementMapper;  
import com.labsfrom6to8.Model.Settlement;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class SettlementService {  
  
 private final SettlementMapper settlementMapper;  
  
 public SettlementService(SettlementMapper settlementMapper) {  
 this.settlementMapper = settlementMapper;  
 }  
  
 public void addSettlement(Settlement settlement) {  
 settlementMapper.insertSettlement(settlement);  
 }  
  
 public void updateSettlement(Settlement settlement) {  
 settlementMapper.updateSettlement(settlement);  
 }  
  
 public List<Settlement> getActiveSettlements() {  
 return settlementMapper.selectActiveSettlements();  
 }  
  
 public void deleteSettlementById(Long id) {  
 settlementMapper.deleteSettlement(id);  
 }  
}

package com.labsfrom6to8.Controller;  
  
import com.labsfrom6to8.Model.Settlement;  
import com.labsfrom6to8.Service.SettlementService;  
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("/settlements")  
@Tag(name = "Settlement API", description = "API для управления населенными пунктами")  
public class SettlementController {  
  
 private final SettlementService settlementService;  
  
 public SettlementController(SettlementService settlementService) {  
 this.settlementService = settlementService;  
 }  
  
 @PostMapping  
 @Operation(summary = "Добавить новый населенный пункт", description = "Добавляет новый населенный пункт в систему")  
 public void addSettlement(@RequestBody Settlement settlement) {  
 settlementService.addSettlement(settlement);  
 }  
  
 @PutMapping  
 @Operation(summary = "Обновить населенный пункт", description = "Обновляет информацию о населенном пункте")  
 public void updateSettlement(@RequestBody Settlement settlement) {  
 settlementService.updateSettlement(settlement);  
 }  
  
 @GetMapping  
 @Operation(summary = "Получить список активных населенных пунктов", description = "Возвращает список всех активных населенных пунктов")  
 public List<Settlement> getActiveSettlements() {  
 return settlementService.getActiveSettlements();  
 }  
  
 @DeleteMapping("/{id}")  
 @Operation(summary = "Удалить населенный пункт", description = "Удаляет населенный пункт по его идентификатору")  
 public void deleteSettlement(@PathVariable Long id) {  
 settlementService.deleteSettlementById(id);  
 }  
}

# **ВЫВОД**

Реализованное приложение обеспечивает функциональность управления населёнными пунктами, предлагая хорошо документированный REST API с поддержкой Swagger. Использование MyBatis позволяет эффективно работать с базой данных, а архитектура приложения соответствует принципам модульности, что упрощает его поддержку и расширение. Приложение подходит для интеграции в большие системы управления данными.