Федеральное государственное автономное образовательное учреждение высшего образования

 Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «*Сервис-ориентированная архитектура*»

**Отчёт**

**По лабораторной работе №1**

**Вариант: 8702**

Студенты:

*Барсуков М. А.*

группа *P3415*

*Коломиец Н. С.*

группа *P3410*

Преподаватель:

*Коновалов А. А.*

г. Санкт-Петербург, 2025 г.

**Задание**

Разработать спецификацию в формате OpenAPI для набора веб-сервисов, реализующего следующую функциональность:

**Первый веб-сервис** должен осуществлять управление коллекцией объектов. В коллекции необходимо хранить объекты класса Person, описание которого приведено ниже:

| public class Person {  private Integer id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически  private String name; //Поле не может быть null, Строка не может быть пустой  private Coordinates coordinates; //Поле не может быть null  private java.time.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически  private Float height; //Поле может быть null, Значение поля должно быть больше 0  private Color eyeColor; //Поле не может быть null  private Color hairColor; //Поле может быть null  private Country nationality; //Поле может быть null  private Location location; //Поле не может быть null  }  public class Coordinates {  private int x;  private int y;  }  public class Location {  private Integer x; //Поле не может быть null  private Long y; //Поле не может быть null  private Integer z; //Поле не может быть null  private String name; //Длина строки не должна быть больше 704, Поле может быть null  }  public enum Color {  RED, BLUE, YELLOW, ORANGE;  }  public enum Color {  GREEN, RED, YELLOW, ORANGE, BROWN;  }  public enum Country {  CHINA, INDIA, ITALY, NORTH\_KOREA;  } |
| --- |

Веб-сервис должен удовлетворять следующим требованиям:

* API, реализуемый сервисом, должен соответствовать рекомендациям подхода RESTful.
* Необходимо реализовать следующий базовый набор операций с объектами коллекции: добавление нового элемента, получение элемента по ИД, обновление элемента, удаление элемента, получение массива элементов.
* Операция, выполняемая над объектом коллекции, должна определяться методом HTTP-запроса.
* Операция получения массива элементов должна поддерживать возможность сортировки и фильтрации по любой комбинации полей класса, а также возможность постраничного вывода результатов выборки с указанием размера и порядкового номера выводимой страницы.
* Все параметры, необходимые для выполнения операции, должны передаваться в URL запроса.
* Информация об объектах коллекции должна передаваться в формате **json**.
* В случае передачи сервису данных, нарушающих заданные на уровне класса ограничения целостности, сервис должен возвращать код ответа http, соответствующий произошедшей ошибке.

Помимо базового набора, веб-сервис должен поддерживать следующие операции над объектами коллекции:

* Удалить все объекты, значение поля nationality которого эквивалентно заданному.
* Удалить один (любой) объект, значение поля location которого эквивалентно заданному.
* Вернуть массив объектов, значение поля location которых больше заданного.

Эти операции должны размещаться на отдельных URL.

**Второй веб-сервис** должен располагаться на URL /demography, и реализовывать ряд дополнительных операций, связанных с вызовом API первого сервиса:

* /hair-color/{hair-color}/percentage : вывести долю людей с заданным цветом волос в общей популяции (в процентах)
* /eye-color/{eye-color}: вывести количество людей с заданным цветом глаз

Эти операции также должны размещаться на отдельных URL.

Для разработанной спецификации необходимо сгенерировать интерактивную веб-документацию с помощью Swagger UI. Документация должна содержать описание всех REST API обоих сервисов с текстовым описанием функциональности каждой операции. Созданную веб-документацию необходимо развернуть на сервере helios.

**Выполнение**

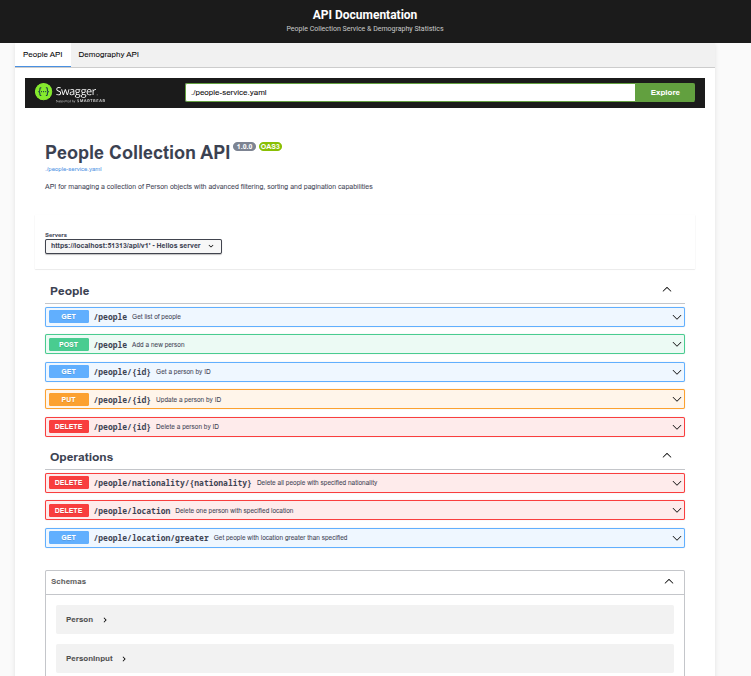
**Исходный код спецификации**

[https://github.com/maxbarsukov/itmo/blob/master/7 соа/лабораторные/lab1/people-service.yaml](https://github.com/maxbarsukov/itmo/blob/master/7%20%D1%81%D0%BE%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab1/people-service.yaml)

[https://github.com/maxbarsukov/itmo/blob/master/7 соа/лабораторные/lab1/demography-service.yaml](https://github.com/maxbarsukov/itmo/blob/master/7%20%D1%81%D0%BE%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab1/demography-service.yaml)

**Сгенерированный Swagger UI**

<https://se.ifmo.ru/~s367081/>



*Рис 1. Swagger UI для people-service*

**Выводы**

Во время выполнения лабораторной работы была разработана спецификация в формате OpenAPI для набора веб-сервисов.