



Факультет программной инженерии и компьютерной техники  
Сервис ориентированная архитектура

Лабораторная работа №1

Вариант: 43435

Преподаватель: Исаев Илья Владимирович  
Выполнил: Васькин Алексей Андреевич Р34101

Санкт-Петербург  
2022

## 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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Double height; //Поле может быть null, Значение поля должно быть больше 0
    private String passportID; //Строка не может быть пустой, Длина строки должна быть не меньше 10, Длина строки не должна быть больше 34, Поле не может быть null
    private Color eyeColor; //Поле может быть null
    private Country nationality; //Поле может быть null
    private Location location; //Поле не может быть null
}
public class Coordinates {
    private Long x; //Максимальное значение поля: 273, Поле не может быть null
    private Long y; //Максимальное значение поля: 343, Поле не может быть null
}
public class Location {
    private Long x; //Поле не может быть null
    private Integer y; //Поле не может быть null
    private Integer z; //Поле не может быть null
}
public enum Color {
    BLACK,
    BLUE,
    YELLOW,
    BROWN;
}
public enum Country {
    INDIA,
    VATICAN,
    THAILAND,
    NORTH_KOREA,
    JAPAN;
}
```

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分页 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 **xml**.
- 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:

- Вернуть один (любой) объект, значение поля passportID которого является максимальным.
- Сгруппировать объекты по значению поля id, вернуть количество элементов в каждой группе.
- Вернуть количество объектов, значение поля height которых меньше 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:

- **/eye-color/{eye-color}/percentage** : вывести долю людей с заданным цветом глаз в общей популяции (в процентах)
- **/nationality/{nationality}/eye-color/{eye-color}/percentage** : вывести долю людей с заданным цветом глаз в пределах указанной национальности (в процентах)

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**.

### **Вопросы к защите лабораторной работы:**

1. Подходы к проектированию приложений. "Монолитная" и сервис-ориентированная архитектура.
2. Понятие сервиса. Общие свойства сервисов.

3. Основные принципы SOA. Подходы к реализации SOA, стандарты и протоколы.
4. Общие принципы построения и элементы сервис-ориентированных систем.
5. Понятие веб-сервиса. Определение, особенности, отличия от веб-приложений.
6. Категоризация веб-сервисов. RESTful и SOAP. Сходства и отличия, области применения.
7. RESTful веб-сервисы. Особенности подхода. Понятия ресурса, URI и полезной нагрузки (payload).
8. Виды RESTful-сервисов. Интерпретация методов HTTP в RESTful.
9. Правила именования ресурсов в RESTful сервисах.
10. Спецификация RESTful-сервисов. Стандарт OpenAPI.
11. Автодокументирование RESTful-сервисов. Swagger Editor, Swagger UI (и Swagger Codegen).

### Выполнение

Ссылка на Swagger UI: <https://se.ifmo.ru/~s265058/soa1/>

Ссылка на GitHub: <https://github.com/reddist/SOA/tree/main/lab1>

### Вывод

В ходе выполнения данной лабораторной работы я изучил спецификацию OpenAPI, инструмент Swagger, написал спецификацию OpenAPI для двух веб-сервисов и создал для неё интерактивную веб-документацию с помощью Swagger UI.