

# УНИВЕРСИТЕТ ИТМО

СЕРВИС ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА

## Лабораторная работа №1

ФИО студентов:

Готовко Алексей Владимирович

Руденко Илья Александрович

Вариант: 197

Направление подготовки: 09.03.04 (СППО)

Учебная группа: Р34101

ФИО преподавателя: Кривоносов Егор Дмитриевич

Санкт-Петербург

2024г.

## Содержание

<b>1</b>	<b>Задание</b>	<b>2</b>
<b>2</b>	<b>Развернутый API</b>	<b>3</b>
<b>3</b>	<b>Исходный код</b>	<b>3</b>
<b>4</b>	<b>Вывод</b>	<b>3</b>

# 1 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 **LabWork**, описание которого приведено ниже:

---

```
1 public class LabWork {
2     // Поле не может быть null,
3     // Значение поля должно быть больше 0,
4     // Значение этого поля должно быть уникальным,
5     // Значение этого поля должно генерироваться автоматически
6     private Long id;
7     // Поле не может быть null, Строка не может быть пустой
8     private String name;
9     // Поле не может быть null
10    private Coordinates coordinates;
11    // Поле не может быть null,
12    // Значение этого поля должно генерироваться автоматически
13    private java.time.LocalDate creationDate;
14    // Значение поля должно быть больше 0
15    private double minimalPoint;
16    // Поле не может быть null
17    private Difficulty difficulty;
18    // Поле не может быть null
19    private Discipline discipline;
20 }
21 public class Coordinates {
22     private long x; //Значение поля должно быть больше -895
23     private Integer y; //Значение поля должно быть больше -529, Поле не может быть null
24 }
25 public class Discipline {
26     private String name; //Поле не может быть null, Строка не может быть пустой
27     private int selfStudyHours;
28 }
29 public enum Difficulty {
30     VERY_EASY,
31     EASY,
32     HARD,
33     VERY_HARD,
34     HOPELESS;
35 }
```

---

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ного 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 `discipline` которого эквивалентно заданному.
- Рассчитать сумму значений поля `minimalPoint` для всех объектов.
- Вернуть массив объектов, значение поля `'name'` которых содержит заданную подстроку.

Эти операции должны размещаться на отдельных URL.

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

- `/labwork/labwork-id/difficulty/increase/steps-count`: повысить сложность заданной лабораторной работы на указанное число "шагов".
- `/discipline/discipline-id/make-hardcore`: добавить в программу указанной дисциплины 10 самых сложных лабораторных работ.

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`.

## 2 Развернутый API

Доступен по [ссылке](https://se.ifmo.ru/~s335151/) (`https://se.ifmo.ru/~s335151/`).

## 3 Исходный код

Доступен по [ссылке](https://github.com/xGodness/itmo/tree/main/4-year/service-oriented-architecture/1-lab) (`https://github.com/xGodness/itmo/tree/main/4-year/service-oriented-architecture/1-lab`).

## 4 Вывод

В результате выполнения лабораторной работы был в достаточной степени освоен набор инструментов Swagger. Было получено неимоверное удовольствие.