

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное автономное образование учреждение  
высшего образования “Национальный исследовательский университет  
ИТМО”

**ОТЧЁТ**

По лабораторной работе №1  
По дисциплине “Сервис-ориентированная архитектура”  
Вариант: 8413

**Выполнил:**

Трошкин Александр Евгеньевич

**Факультет:**

ПИиКТ

**Преподаватель:**

Егошин Алексей Васильевич



Санкт-Петербург, 2025

# 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 **Movie**, описание которого приведено ниже:

```
public class Movie {
    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 Integer oscarsCount; //Значение поля должно быть больше 0, Поле
    может быть null
    private Double totalBoxOffice; //Поле может быть null, Значение поля должно
    быть больше 0
    private Long length; //Поле не может быть null, Значение поля должно быть
    больше 0
    private MovieGenre genre; //Поле может быть null
    private Person operator; //Поле может быть null
}
public class Coordinates {
    private Integer x; //Значение поля должно быть больше -651, Поле не может
    быть null
    private float y; //Значение поля должно быть больше -613
}
public class Person {
    private String name; //Поле не может быть null, Строка не может быть
    пустой
    private String passportID; //Длина строки должна быть не меньше 8, Строка не
    может быть пустой, Значение этого поля должно быть уникальным, Поле не может
    быть null
    private Color eyeColor; //Поле может быть null
    private Color hairColor; //Поле не может быть null
    private Country nationality; //Поле может быть null
    private Location location; //Поле не может быть null
}
public class Location {
    private double x;
    private Long y; //Поле не может быть null
```

```

    private Integer z; //Поле не может быть null
}
public enum MovieGenre {
    COMEDY,
    ADVENTURE,
    TRAGEDY,
    SCIENCE_FICTION;
}
public enum Color {
    GREEN,
    RED,
    BLACK,
    ORANGE;
}
public enum Color {
    GREEN,
    RED,
    BLACK,
    BLUE;
}
public enum Country {
    FRANCE,
    VATICAN,
    SOUTH_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 **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:

- Рассчитать сумму значений поля length для всех объектов.
- Вернуть количество объектов, значение поля genre которых равно заданному.
- Вернуть массив объектов, значение поля name которых начинается с заданной подстроки.

Эти операции должны размещаться на отдельных URL.

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

- `/directors/get-loosers` : получить список режиссёров, ни один фильм которых не получил ни одного "Оскара"
- `/directors/humiliate-by-genre/{genre}` : отобрать все "Оскары" у всех фильмов режиссёров, снявших хоть один фильм в указанном жанре

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

Спецификация в формате OpenAPI

<https://github.com/k1nd-cat/service-oriented-architecture-itmo/blob/main/open-api/movie-api-openapi.yaml>

Интерактивная веб-документация через Swagger UI

<https://se.ifmo.ru/~s333304/swagger-ui/swagger.html>

# Вывод

Я научился вручную создавать спецификацию в формате OpenAPI, а также разворачивать её на helios