

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет
по лабораторной работе № 1 по дисциплине «Сервис-ориентированная
архитектура»

Вариант 192

Автор: Иванов Андрей Вячеславович

Факультет: ПИиКТ

Группа: Р34101

Преподаватель: Кривоносов Егор Дмитриевич



Санкт-Петербург, 2024

Задание:

Лабораторная работа #1

Введите вариант: 192

Внимание! У разных вариантов разный текст задания!

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 Long id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значен
    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 long oscarsCount; //Значение поля должно быть больше 0
    private MovieGenre genre; //Поле не может быть null
    private Mpaarating mpaarating; //Поле не может быть null
    private Person screenwriter;
}

public class Coordinates {
    private double x;
    private int y;
}

public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private java.time.LocalDate birthday; //Поле может быть null
    private float height; //Значение поля должно быть больше 0
    private Color hairColor; //Поле может быть null
    private Country nationality; //Поле может быть null
}

public enum MovieGenre {
    ACTION,
    MUSICAL,
    ADVENTURE,
    FANTASY;
}

public enum Mpaarating {
    G,
    PG,
    R,
    NC_17;
}

public enum Color {
    BLACK,
    YELLOW,
    ORANGE,
    WHITE;
}

public enum Country {
    UNITED_KINGDOM,
    FRANCE,
    SPAIN,
    INDIA;
}
```

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 ID, 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 **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 screenwriter которого является максимальным.
- Вернуть массив объектов, значение поля name которых содержит заданную подстроку.
- Вернуть массив объектов, значение поля oscarCount которых меньше заданного.

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"
- `/movies/honor-by-length/{min-length}/oscar-to-add` : дополнительно наградить все фильмы с продолжительностью больше заданной указанным числом "Оскаров"

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**.

Ссылка на спецификацию:

<https://se.ifmo.ru/~s336587>

Ссылка на репозиторий с кодом:

<https://github.com/ANVISERO/ITMO/tree/main/4course/SOA/labs/lab1/dist>

Инструкция по развёртыванию:

Сначала нужно скачать архив по данной [ссылке](#) и разархивировать его. Затем ищем папку dist и в неё кладём наши файлы с описанием. Далее в файле swagger-initializer.js меняем путь до файлов, чтобы он указывал на файлы с описанием OpenAPI.

Пример:

```
urls: [  
  {url: "./MovieService.yaml", name: "MovieService"},  
  {url: "./OscarService.yaml", name: "OscarService"},  
]
```

Вывод:

Во время выполнения лабораторной работы познакомился со спецификацией

Open API инструментами написания документации Swagger UI и Swagger Editor, разработав спецификацию для набора веб-сервисов в соответствии с вариантом задания.