

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ»



Факультет Программной Инженерии и Компьютерной Техники

Дисциплина:
«Сервис-ориентированная архитектура»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Вариант 5192

Выполнили:
Студенты гр. Р34151
Соловьев Артемий Александрович
Понамареv Степан Андреевич

Проверил:
Цона Евгений Алексеевич

Санкт-Петербург
2024г.

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 int id; //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.util.Date creationDate; //Поле не может быть null, Значение этого поля
    должно генерироваться автоматически
    private long oscarsCount; //Значение поля должно быть больше 0
    private int usaBoxOffice; //Значение поля должно быть больше 0
    private MovieGenre genre; //Поле может быть null
    private Mpaarating mpaarating; //Поле может быть null
    private Person screenwriter;
}
public class Coordinates {
    private long x;
    private Long y; //Максимальное значение поля: 479, Поле не может быть null
}
public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Double height; //Поле может быть null, Значение поля должно быть больше 0
    private Color eyeColor; //Поле может быть null
    private Color hairColor; //Поле не может быть null
    private Country nationality; //Поле не может быть null
    private Location location; //Поле не может быть null
}
public class Location {
    private Double x; //Поле не может быть null
    private float y;
    private String name; //Поле не может быть null
}
public enum MovieGenre {
    WESTERN,
    MUSICAL,
    ADVENTURE,
    TRAGEDY;
}
public enum Mpaarating {
    G,
    PG,
    PG_13,
    NC_17;
}
public enum Color {
    GREEN,
    RED,
    BLACK,
```

```

    WHITE;
}
public enum Color {
    GREEN,
    YELLOW,
    ORANGE,
    WHITE,
    BROWN;
}
public enum Country {
    GERMANY,
    VATICAN,
    ITALY,
    SOUTH_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 **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 `usaBoxOffice` для всех объектов.
- Вернуть количество объектов, значение поля `screenwriter` которых больше заданного.
- Вернуть массив объектов, значение поля `name` которых содержит заданную подстроку.

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:

- `/screenwriters/get-loosers` : получить список сценаристов, ни один фильм которых не получил ни одного "Оскара"
- `/directors/humiliate-by-genre/{genre}` : отобразить все "Оскары" у всех фильмов режиссёров, снявших хотя бы 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.

Сгенерированный SwaggerUI

<https://se.ifmo.ru/~s334645/soa>

Исходный код спецификации

<https://github.com/Fer0ty/SOA>

Вывод

Мы научились писать доку