

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Сервис-ориентированная архитектура»

Лабораторная работа №1

Вариант 9189

Студент

Чебоксаров Я.М.

Иванов А.А

Р34091

Преподаватель

Цопа Е.А.

Санкт-Петербург, 2024 г.

Оглавление

Описание задания	2
Тесты.....	Ошибка! Закладка не определена.
Таблица \arcsin	Ошибка! Закладка не определена.
Выводы.....	5

Описание 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 Dragon, описание:

```
public class Dragon {  
  
    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.time.LocalDate creationDate; //Поле не может быть null,  
    Значение этого поля должно генерироваться автоматически  
  
    private Integer age; //Значение поля должно быть больше 0, Поле  
    может быть null  
  
    private Long wingspan; //Значение поля должно быть больше 0, Поле  
    может быть null  
  
    private boolean speaking;  
  
    private Color color; //Поле не может быть null  
  
    private DragonHead head;  
  
}  
  
public class Coordinates {  
  
    private Integer x; //Максимальное значение поля: 411, Поле не может  
    быть null  
  
    private Double y; //Значение поля должно быть больше -296, Поле не  
    может быть null  
  
}  
  
public class DragonHead {  
  
    private Double eyesCount; //Поле может быть null  
  
}  
  
public enum Color {  
  
    RED,  
  
    BLACK,  
  
    BLUE,  
  
    WHITE;  
  
}
```

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 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 количество объектов, значение поля head которых равно заданному.
- Вернуть количество объектов, значение поля color 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 /killer, и реализовывать ряд дополнительных операций, связанных с вызовом API первого сервиса:


- /dragon/find-by-cave-depth/{max} : найти дракона, живущего в самой глубокой (или мелкой) пещере
- /team/{team-id}/move-to-cave/{cave-id} : отправить команду убийц драконов в указанную пещеру

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 Swagger UI

[Swagger Docs](#)

 **Swagger**
Supported by SMARTBEAR

Select a definition Dragon

Dragon Management API 1.0.0 OAS 3.0

./first_service.yaml

API для управления коллекцией объектов типа Dragon

default ^

GET	/dragons	Получить все объекты Dragon с поддержкой сортировки, фильтрации и пагинации	^
POST	/dragons	Создать новый объект Dragon	^
GET	/dragons/{id}	Получить объект Dragon по ID	^
PUT	/dragons/{id}	Обновить объект Dragon по ID	^
DELETE	/dragons/{id}	Удалить объект Dragon по ID	^
GET	/dragons/count	Вернуть количество объектов Dragon с заданным количеством глаз у головы	^
GET	/dragons/countByColor	Вернуть количество объектов Dragon с цветом больше заданного	^
GET	/dragons/searchByName	Вернуть массив объектов Dragon, имя которых содержит заданную подстроку	^
POST	/dragonHunter	Добавить нового охотника на драконов	^
GET	/dragonHunter	Получить массив охотников на драконов	^
GET	/dragonHunter/{id}	Получить охотника по ID	^
PUT	/dragonHunter/{id}	Обновить охотника по ID	^
DELETE	/dragonHunter/{id}	Удалить охотника по ID	^

Файлы с исходной конфигурацией

[Ссылка на репозиторий с конфигурацией](#)

Выводы

В ходе выполнения лабораторной работы была разработана спецификация REST API в формате OpenAPI для набора веб-сервисов.