УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Сервис-Ориентированная архитектура»

**Лабораторная работа №1**

Вариант 1110

Студент

*Патутин В.М*

*P34101*

Преподаватель

*Исаев И.В.*

Санкт-Петербург, 2022 г.

Задание лабораторной работы

Разработать спецификацию в формате OpenAPI для набора веб-сервисов, реализующего следующую функциональность:

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 Product, описание которого приведено ниже:

public class Product {

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 Long price; //Поле не может быть null, Значение поля должно быть больше 0

private String partNumber; //Поле может быть null

private Integer manufactureCost; //Поле не может быть null

private UnitOfMeasure unitOfMeasure; //Поле может быть null

private Person owner; //Поле не может быть null

}

public class Coordinates {

private float x; //Максимальное значение поля: 791

private Float y; //Поле не может быть null

}

public class Person {

private String name; //Поле не может быть null, Строка не может быть пустой

private String passportID; //Длина строки не должна быть больше 40, Поле не может быть null

private Color eyeColor; //Поле не может быть null

private Color hairColor; //Поле может быть null

private Country nationality; //Поле может быть null

private Location location; //Поле может быть null

}

public class Location {

private int x;

private Double y; //Поле не может быть null

private double z;

}

public enum UnitOfMeasure {

KILOGRAMS,

METERS,

CENTIMETERS,

SQUARE\_METERS,

MILLILITERS;

}

public enum Color {

BLACK,

BLUE,

ORANGE,

BROWN;

}

public enum Country {

RUSSIA,

SPAIN,

NORTH\_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 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 owner которых больше заданного.
* Вернуть массив объектов, значение поля name которых содержит заданную подстроку.
* Вернуть массив уникальных значений поля owner по всем объектам.

Эти операции должны размещаться на отдельных URL.

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

* /filter/manufacturer/{manufacturer-id} : выбрать всю продукцию заданного owner
* /price/decrease/{decrease-percent} : снизить цену всей продукции на указанный процент

Эти операции также должны размещаться на отдельных URL.

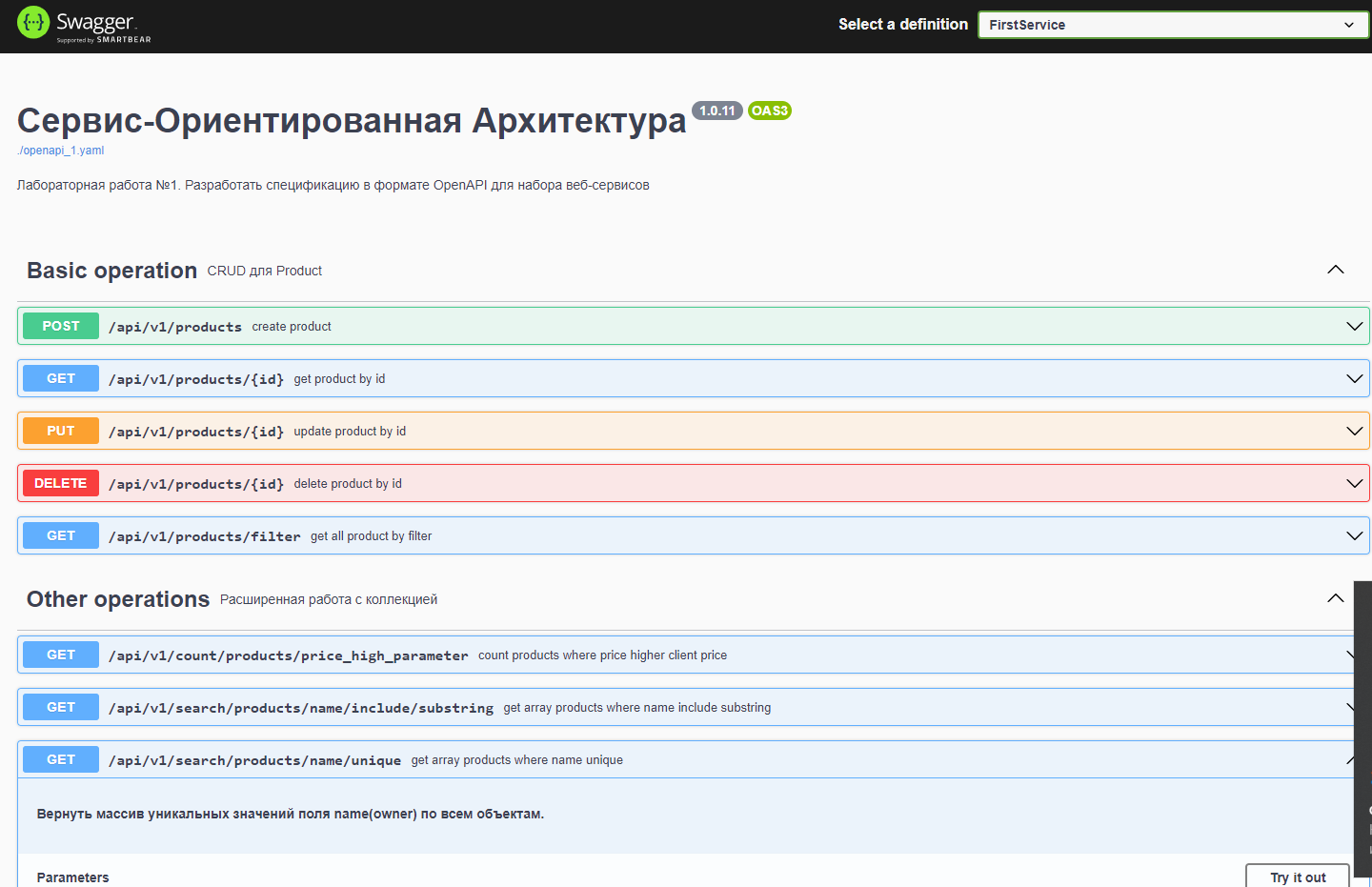
Для разработанной спецификации необходимо сгенерировать интерактивную веб-документацию с помощью Swagger UI. Документация должна содержать описание всех REST API обоих сервисов с текстовым описанием функциональности каждой операции. Созданную веб-документацию необходимо развернуть на сервере helios.

Спецификация REST API

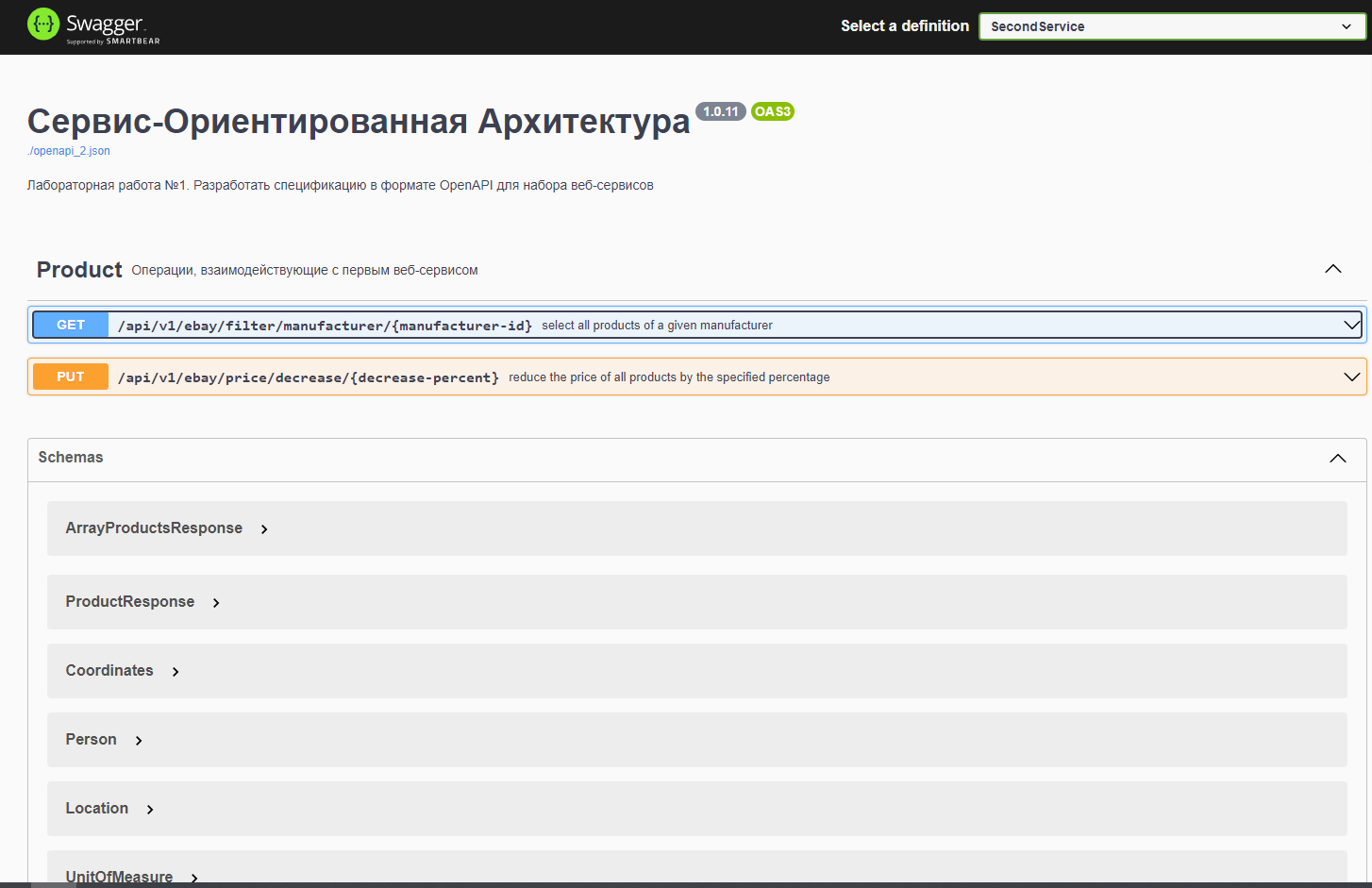
Спецификация доступна по ссылке:

<https://se.ifmo.ru/~s282606/SOA/Lab1/>

*Первый веб – сервис:*



*Второй веб – сервис:*



Вывод

Во время выполнения лабораторной работы мы изучили микросервисный подход к проектированию приложений и глубже познакомились со спецификацией OpenAPI и инструментами написания документации Swagger UI и Swagger Editor, разработав спецификацию в соответствии с вариантов задания