УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Сервис-Ориентированная архитектура»

**Лабораторная работа №1**

Вариант 1209

Студент

*Митрофанов Е. Ю.*

*Любкин А. С.*

*P34101*

Преподаватель

*Цопа Е. А.*

Санкт-Петербург, 2022 г.

Задание лабораторной работы

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 SpaceMarine, описание которого приведено ниже:

public class SpaceMarine {

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

private AstartesCategory category; //Поле может быть null

private Weapon weaponType; //Поле не может быть null

private MeleeWeapon meleeWeapon; //Поле не может быть null

private Chapter chapter; //Поле может быть null

}

public class Coordinates {

private Long x; //Поле не может быть null

private Integer y; //Поле не может быть null

}

public class Chapter {

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

private String parentLegion;

private Integer marinesCount; //Поле не может быть null, Значение поля должно быть больше 0, Максимальное значение поля: 1000

private String world; //Поле не может быть null

}

public enum AstartesCategory {

AGGRESSOR,

ASSAULT,

HELIX;

}

public enum Weapon {

COMBI\_FLAMER,

GRENADE\_LAUNCHER,

INFERNO\_PISTOL;

}

public enum MeleeWeapon {

CHAIN\_SWORD,

CHAIN\_AXE,

POWER\_FIST;

}

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 **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:

* Сгруппировать объекты по значению поля meleeWeapon, вернуть количество элементов в каждой группе.
* Вернуть массив объектов, значение поля name которых содержит заданную подстроку.
* Вернуть массив объектов, значение поля chapter которых больше заданного.

Эти операции должны размещаться на отдельных URL.

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

* /create/{id}/{name} : создать новый десантный корабль и сохранить его в БД
* /{starship-id}/load//space-marine-id : погрузить выбранного десантника на корабль

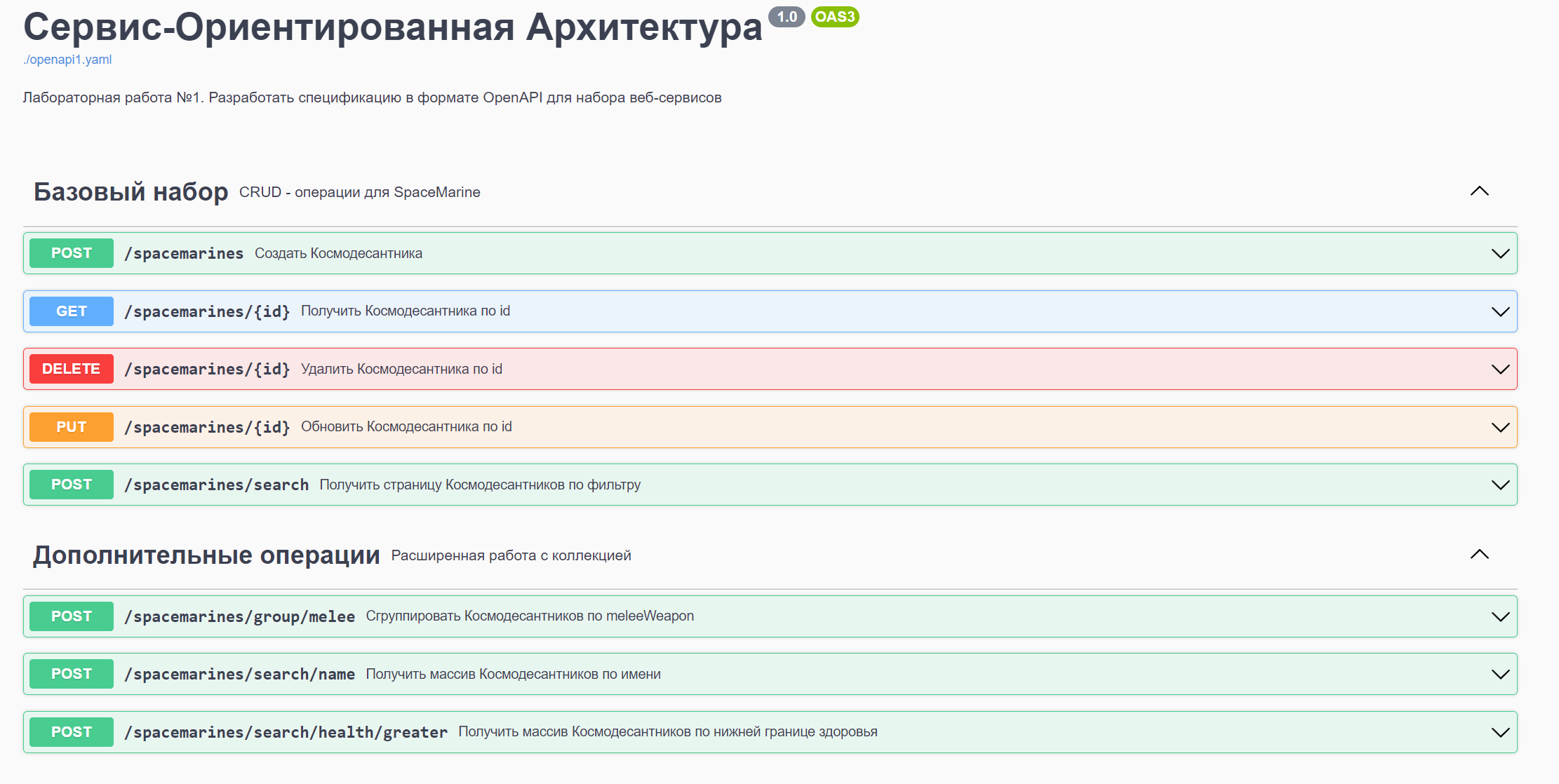
Эти операции также должны размещаться на отдельных URL.

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.

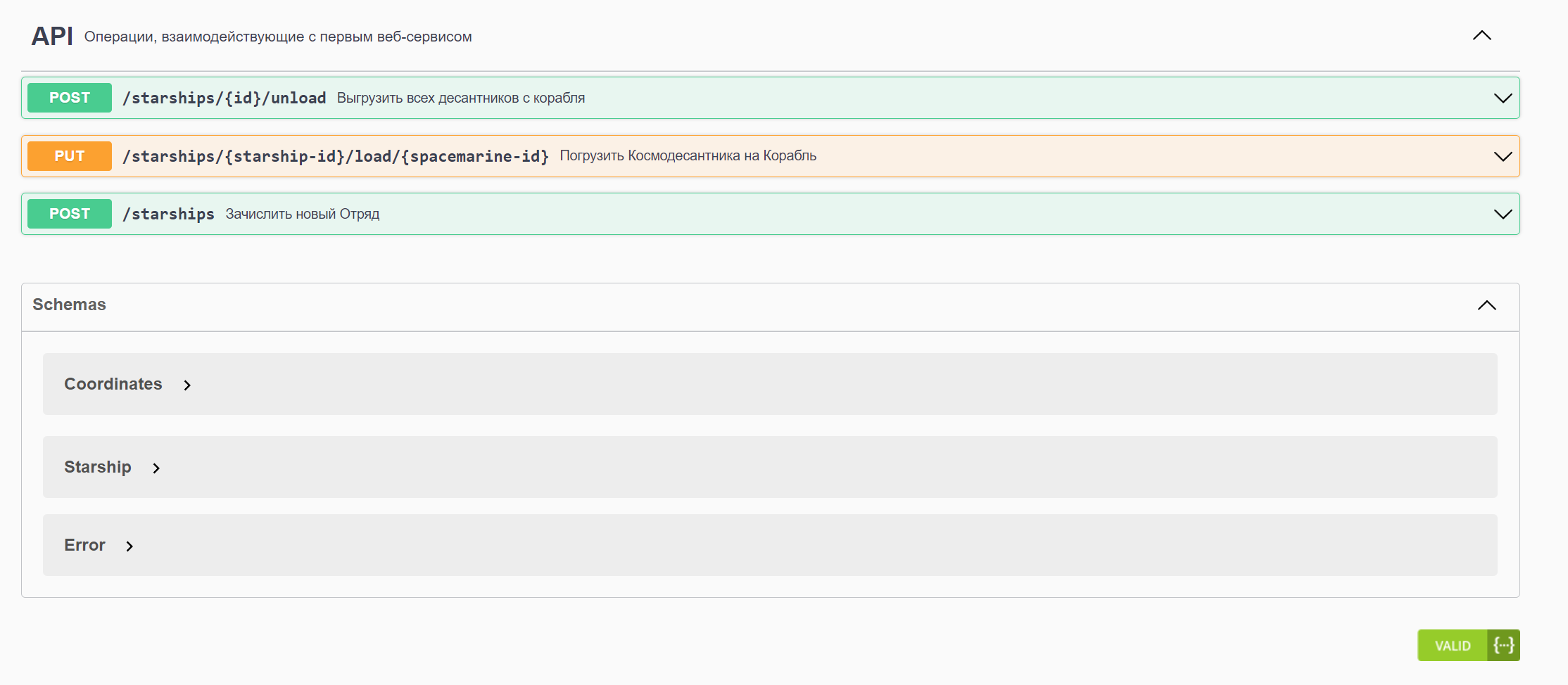
Спецификация REST API

Спецификация доступна по ссылке: <https://se.ifmo.ru/~s284731/SOA>

*Первый веб – сервис:*



*Второй веб – сервис:*



Вывод

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