



УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники
Сервис-ориентированная архитектура

Лабораторная работа №1
Вариант: 1012

Преподаватель: Усков Иван Владимирович
Выполнил: Кульбако Артемий Юрьевич Р34112

Санкт-Петербург
2021

Задание

Разработать веб-сервис на базе сервлета, реализующий управление коллекцией объектов, и клиентское веб-приложение, предоставляющее интерфейс к разработанному веб-сервису:

```
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 int age; //Значение поля должно быть больше 0
    private double wingspan; //Значение поля должно быть больше 0
    private Color color; //Поле не может быть null
    private DragonType type; //Поле может быть null
    private Person killer; //Поле может быть null
}
public class Coordinates {
    private double x;
    private long y;
}
public class Person {
    private String name; //Поле не может быть null, Строка не может быть
    пустой
    private Double height; //Поле не может быть null, Значение поля должно
    быть больше 0
    private Double weight; //Поле может быть null, Значение поля должно быть
    больше 0
    private String passportID; //Строка не может быть пустой, Длина строки
    должна быть не меньше 5, Длина строки не должна быть больше 22, Поле может
    быть null
    private Color hairColor; //Поле может быть null
}
public enum Color {
    GREEN,
    YELLOW,
    BROWN;
}
public enum DragonType {
    WATER,
    AIR,
    FIRE;
}
```

Схема API

Прикрутить Swagger к сервлетам проблематично, поэтому описано вручную.

1	api
2	admin
3	[GET] - проверить работоспособности api
4	[POST] - создать таблицы БД
5	[DELETE] - удалить все сущность из таблиц
6	persons
7	find_person_included_in_name<name>
8	[GET] - получить персонажей в имени которых содержится <name>
9	id
10	[GET] - получить определённого персонажа по id
11	[GET] - получить всех персонажей
12	[POST] - создать персонажа
13	[PUT] - модифицировать персонажа (достаточно предоставить модифицированные данные)
14	[DELETE] - удалить персонажа
15	dragons
16	grouped_by_type
17	[GET] - получить статистику драконов по типам
18	find_with_killer_weaker_then<killer_id>
19	[GET] - получить драконов, у которых убийца слабее <killer_id>
20	id
21	[GET] - получить определённого дракона по id
22	[GET] - получить всех драконов
23	[POST] - создать дракона
24	[PUT] - модифицировать дракона (достаточно предоставить модифицированные данные)
25	[DELETE] - удалить дракона

Код

<https://github.com/testpassword/Service-oriented-architecture/tree/master/lab1-05.07.21>

Вывод

В рамках лабораторной работы был написан пример монолитного сервиса на javax.servlet и React. Новых практических навыков, к сожалению, эта лабораторная не даёт, но позволяет посмотреть, как страдали программисты раньше, разрабатывая на чистых сервлетах и деплоя свои приложения с использованием контейнеров сервлетов. К тому же, технология сервлетов соответствует принципам DI, IOC, что, на мой взгляд, зачастую только усложняет разработку ввиду отсутствия полноценного контроля управления программой. Разработанный мною бекенд, поддерживает работу в двух режимах, с использованием встроенного в jar сервера и в форме war веб-приложения: разрабатывать и разворачивать первый было значительно проще.