

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерных технологий

Сервисно-ориентированная архитектура

Лабораторная работа № 1

Вариант 1006

Выполнил
студент

Демичев Даниил Дмитриевич

Группа № Р34112

Преподаватель: Усков Иван Владимирович

г. Санкт-Петербург

2021

Задание

Разработать веб-сервис на базе сервлета, реализующий управление коллекцией объектов, и клиентское веб-приложение, предоставляющее интерфейс к разработанному веб-сервису. В коллекции необходимо хранить объекты класса **HumanBeing**, описание которого приведено ниже:

```
public class HumanBeing {
    private Long 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.LocalDateTime creationDate; //Поле не может быть null,
    Значение этого поля должно генерироваться автоматически
    private Boolean realHero; //Поле не может быть null
    private boolean hasToothpick;
    private Float impactSpeed; //Поле не может быть null
    private String soundtrackName; //Поле не может быть null
    private WeaponType weaponType; //Поле может быть null
    private Mood mood; //Поле может быть null
    private Car car; //Поле не может быть null
}
public class Coordinates {
    private Integer x; //Максимальное значение поля: 416, Поле не может быть null
    private Long y; //Значение поля должно быть больше -700, Поле не может быть
    null
}
public class Car {
    private String name; //Поле может быть null
    private Boolean cool; //Поле может быть null
}
public enum WeaponType {
    AXE,
    RIFLE,
    MACHINE_GUN,
    BAT;
}
public enum Mood {
    SORROW,
    CALM,
    RAGE,
    FRENZY;
}
```

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.
- Данные коллекции, которыми управляет веб-сервис, должны храниться в реляционной базе данных.
- 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.
- Веб-сервис должен быть "упакован" в веб-приложение, которое необходимо развернуть на сервере приложений **Jetty**.

Помимо базового набора, веб-сервис должен поддерживать следующие операции над объектами коллекции:

- Вернуть количество объектов, значение поля `weaponType` которых больше заданного.
- Вернуть массив объектов, значение поля `weaponType` которых больше заданного.
- Вернуть массив уникальных значений поля `impactSpeed` по всем объектам.

Эти операции должны размещаться на отдельных URL.

Требования к клиентскому приложению:

- Клиентское приложение может быть написано на любом веб-фреймворке, который можно запустить на сервере `helios`.
- Клиентское приложение должно обеспечить полный набор возможностей по управлению объектами коллекции, предоставляемых веб-сервисом -- включая сортировку, фильтрацию и постраничный вывод.
- Клиентское приложение должно преобразовывать передаваемые сервисом данные в человеко-читаемый вид -- параграф текста, таблицу и т.д.
- Клиентское приложение должно информировать пользователя об ошибках, возникающих на стороне сервиса, в частности, о том, что сервису были отправлены невалидные данные.

Веб-сервис и клиентское приложение должны быть развёрнуты на сервере helios.

Описание API

URL: /human-beings

Method: GET

Описание: получить коллекцию объектов HumanBeing с возможностью сортировки, фильтрации и分页ного вывода

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/human-beings?pageNumber=1&pageSize=50&orderBy=+impactSpeed&filterBy=name:PFF;weaponType:AXE;mood:CALM

Ответ:

```
{
  "humanBeingList": [
    {
      "id": "8",
      "name": "PFF",
      "coordinates": {
        "id": "15",
        "x": "20",
        "y": "50"
      },
      "creationDate": "null",
      "realHero": "false",
      "hasToothpick": "true",
      "impactSpeed": "35.5",
      "soundtrackName": "Belupacito",
      "weaponType": "AXE",
      "mood": "CALM",
      "car": {
        "id": "13",
        "name": "Lada",
        "cool": "true"
      }
    },
    {
      "id": "6",
```

```
"name": "PFF",
"coordinates": {
  "id": "40",
  "x": "200",
  "y": "3"
},
"creationDate": "null",
"realHero": "false",
"hasToothpick": "true",
"impactSpeed": "35.5",
"soundtrackName": "Belupacito",
"weaponType": "AXE",
"mood": "CALM",
"car": {
  "id": "52",
  "name": "Tesla",
  "cool": "true"
}
},
{
  "id": "7",
  "name": "PFF",
  "coordinates": {
    "id": "41",
    "x": "200",
    "y": "50"
  },
  "creationDate": "null",
  "realHero": "false",
  "hasToothpick": "true",
  "impactSpeed": "35.5",
  "soundtrackName": "Belupacito",
  "weaponType": "AXE",
  "mood": "CALM",
  "car": {
    "id": "53",
    "name": "Lada",
    "cool": "true"
  }
}
],
"count": 3
}
```

URL : /human-beings/{id}

Method: GET

Описание: получить объект коллекции HumanBeing с определенным id.

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/human-beings/6

Ответ:

```
{
  "humanBeingList": [
    {
      "id": "6",
      "name": "PFF",
      "coordinates": {
        "id": "40",
        "x": "200",
        "y": "3"
      },
      "creationDate": "null",
      "realHero": "false",
      "hasToothpick": "true",
      "impactSpeed": "35.5",
      "soundtrackName": "Belupacito",
      "weaponType": "AXE",
      "mood": "CALM",
      "car": {
        "id": "52",
        "name": "Tesla",
        "cool": "true"
      }
    }
  ],
  "count": 1
}
```

URL: /human-beings

Method: POST

Описание: Добавление нового объекта HumanBeing

Пример запроса:

http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/human-beings

Тело запроса:

```
[
  {
    "name": "User",
    "coordinates": {
      "x": "2",
      "y": "3"
    },
    "realHero": "true",
    "hasToothpick": "false",
    "impactSpeed": "1",
    "soundtrackName": "Castle",
    "weaponType": "RIFLE",
    "mood": "FRENZY",
    "car": {
      "name": "Toyota",
      "cool": "false"
    }
  }
]
```

Ответ:

HTTP Status 200

URL: /human-beings/{id}

Method: PUT

Описание: Изменение существующего объекта HumanBeing с определенным id

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/human-beings/30

Тело запроса:

```
[
  {
    "name": "User",
    "coordinates": {
```

```
    "x": "20",
    "y": "30"
  },
  "realHero": "true",
  "hasToothpick": "true",
  "impactSpeed": "1.0",
  "soundtrackName": "Castle",
  "weaponType": "RIFLE",
  "mood": "FRENZY",
  "car": {
    "name": "Toyota",
    "cool": "false"
  }
}
]
```

Ответ:

HPPP Status 200

URL: /human-beings/{id}

Method: DELETE

Описание: удаление объекта HumanBeing с определенным id

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/human-beings/30

Ответ:

HTTP Status 200

URL: /additional?weaponTypeArray={value}

Method: GET

Описание: вернуть количество и массив объектов, значение поля weaponType которых больше заданного.

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/additional?weaponTypeArray=AXE

Ответ:

```
{
```



```
"humanBeingList": [
{
  "id": "16",
  "name": "Max",
  "coordinates": {
    "id": "23",
    "x": "4",
    "y": "4"
  },
  "creationDate": "null",
  "realHero": "true",
  "hasToothpick": "false",
  "impactSpeed": "3.0",
  "soundtrackName": "Pop",
  "weaponType": "BAT",
  "mood": "SORROW",
  "car": {
    "id": "21",
    "name": "BTR",
    "cool": "false"
  }
},
{
  "id": "19",
  "name": "PFF",
  "coordinates": {
    "id": "32",
    "x": "12",
    "y": "12"
  },
  "creationDate": "null",
  "realHero": "true",
  "hasToothpick": "false",
  "impactSpeed": "123.0",
  "soundtrackName": "ads",
  "weaponType": "BAT",
  "mood": "CALM",
  "car": {
    "id": "33",
    "name": "cvsd",
    "cool": "false"
  }
},
{

```

```
"id": "24",
"name": "Skittle",
"coordinates": {
  "id": "35",
  "x": "7",
  "y": "8"
},
"creationDate": "2021-11-19T13:40:11.486",
"realHero": "false",
"hasToothpick": "false",
"impactSpeed": "1.0",
"soundtrackName": "Pop",
"weaponType": "BAT",
"mood": "RAGE",
"car": {
  "id": "36",
  "name": "Zzz",
  "cool": "false"
}
},
"count": 3
}
```

URL: /additional?uniqueImpactSpeed

Method: GET

Описание: вернуть массив уникальных значений поля impactSpeed по всем объектам.

Пример запроса: http://localhost:8080/SOA_Lab1-1.0-SNAPSHOT/additional?uniqueImpactSpeed

Ответ:

```
[
  105.0,
  321.0,
  3.0,
  35.5,
  75.0,
  21.0,
  123.0,
  1.0,
```

2.0,
12.0
]

Вывод

В ходе выполнения этой лабораторной работы я разработал клиентскую часть приложения на ReactJs и серверную, используя сервлеты, настроил их взаимодействие для хранения, модификации и управления объектами коллекции.