



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

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.

```
[
  {
    "path": "api/admin/",
    "methods": [
      {
        "method": "get",
        "res": {
          "content-type": "text/html",
          "content": "<html><body><h1>WELCOME, GOD OF DIGITAL WORLD</h1></body></html>",
          "code": 200
        }
      },
      {
        "method": "post",
        "res": {
          "msg": {
            "content-type": "application/json",
            "content": {
              "msg": "done creation tables"
            },
            "code": 200
          }
        },
        "method": "delete",
        "res": {
          "content-type": "application/json",
          "content": {
            "msg": "database cleared"
          },
          "code": 200
        }
      }
    ]
  },
  {
    "path": "api/dragons/grouped_by_type",
    "methods": [
      {
```

```

    "method": "get",
    "res": {
      "content-type": "application/json",
      "content": {
        "FIRE": "Int",
        "AIR": "Int",
        "UNKNOWN": "Int",
        "WATER": "Int"
      },
      "code": 200
    }
  ]
},
{
  "path": "api/dragons/find_with_killer_weaker_then",
  "methods": [
    {
      "method": "get",
      "params": [
        {
          "killer_id": {
            "required": "true",
            "type": "Long"
          }
        }
      ],
      "res": {
        "content-type": "application/json",
        "content": [
          {
            "id": "Long",
            "name": "Str",
            "coordinates": {
              "x": "Double",
              "y": "Double"
            },
            "creationDate": "Str",
            "age": "Int",
            "wingspan": "Double",
            "color": "Str",
            "type": "Str",
            "killerID": "Long"
          }
        ]
      }
    }
  ]
}

```

```

        }
    ],
    "code": 200
}
}
]
},
{
    "path": "api/dragons/",
    "methods": [
        {
            "method": "get",
            "params": [
                {
                    "offset": {
                        "required": "false",
                        "type": "Int"
                    },
                    "limit": {
                        "required": "false",
                        "type": "Int"
                    },
                    "filters": {
                        "required": "false",
                        "type": "Map<str, str>"
                    },
                    "sorters": {
                        "required": "false",
                        "type": "Map<str, str>"
                    }
                }
            ],
            "res": {
                "content-type": "application/json",
                "content": [
                    {
                        "id": "Long",
                        "name": "Str",
                        "coordinates": {
                            "x": "Double",
                            "y": "Double"
                        },
                        "creationDate": "Str",

```

```

        "age": "Int",
        "wingspan": "Double",
        "color": "Str",
        "type": "Str",
        "killerID": "Long"
    }
],
"code": 200
}
},
{
    "method": "post",
    "req": {
        "content-type": "application/json",
        "required": "true",
        "content": {
            "id": "Long",
            "name": "Str",
            "coordinates": {
                "x": "Double",
                "y": "Double"
            },
            "creationDate": "Str",
            "age": "Int",
            "wingspan": "Double",
            "color": "Str",
            "type": "Str",
            "killerID": "Long"
        }
    },
    "res": {
        "content-type": "application/json",
        "content": {
            "id": "Long"
        },
        "code": 200
    }
}
]
},
{
    "path": "api/dragons/{id}/",
    "methods": [

```

```

{
  "method": "put",
  "req": {
    "content-type": "application/json",
    "required": "optional",
    "content": {
      "id": "Long",
      "name": "Str",
      "coordinates": {
        "x": "Double",
        "y": "Double"
      },
      "creationDate": "Str",
      "age": "Int",
      "wingspan": "Double",
      "color": "Str",
      "type": "Str",
      "killerID": "Long"
    }
  },
  "res": {
    "content-type": "application/json",
    "content": {
      "msg": "successfully modified",
      "id": "Long"
    },
    "code": 200
  }
},
{
  "method": "delete",
  "res": {
    "content-type": "application/json",
    "content": {
      "msg": "successfully removed",
      "id": "Long"
    },
    "code": 200
  }
}
],
{

```

```

"path": "api/persons/find_person_included_in_name/",
"methods": [
  {
    "method": "get",
    "params": [
      {
        "name": {
          "required": "true",
          "type": "Str"
        }
      }
    ],
    "res": {
      "content-type": "application/json",
      "content": [
        {
          "name": "Str",
          "height": "Int",
          "weight": "Int",
          "passportID": "Str",
          "hairColor": "Str"
        }
      ]
    },
    "code": 200
  }
],
},
{
  "path": "api/persons/",
  "methods": [
    {
      "method": "get",
      "params": [
        {
          "offset": {
            "required": "false",
            "type": "Int"
          },
          "limit": {
            "required": "false",
            "type": "Int"
          }
        },

```



```

    "filters": {
      "required": "false",
      "type": "Map<str, str>"
    },
    "sorters": {
      "required": "false",
      "type": "Map<str, str>"
    }
  },
  "res": {
    "content-type": "application/json",
    "content": [
      {
        "name": "Str",
        "height": "Int",
        "weight": "Int",
        "passportID": "Str",
        "hairColor": "Str"
      }
    ],
    "code": 200
  }
},
{
  "method": "post",
  "req": {
    "content-type": "application/json",
    "required": "true",
    "content": {
      "name": "Str",
      "height": "Int",
      "weight": "Int",
      "passportID": "Str",
      "hairColor": "Str"
    }
  },
  "res": {
    "content-type": "application/json",
    "content": {
      "id": "Long"
    },
    "code": 200
  }
}

```

```

    }
  }
]
},
{
  "path": "api/persons/{id}/",
  "methods": [
    {
      "methods": "put",
      "req": {
        "content-type": "application/json",
        "required": "optional",
        "content": {
          "name": "Str",
          "height": "Int",
          "weight": "Int",
          "passportID": "Str",
          "hairColor": "Str"
        }
      },
      "res": {
        "content-type": "application/json",
        "content": {
          "msg": "successfully modified",
          "id": "Long"
        },
        "code": 200
      }
    },
    {
      "method": "delete",
      "res": {
        "content-type": "application/json",
        "content": {
          "msg": "successfully removed",
          "id": "Long"
        },
        "code": 200
      }
    }
  ]
}
]

```

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.