

Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный
исследовательский университет ИТМО»

Лабораторная работа 1

по дисциплине «Сервис-ориентированная архитектура»

Вариант 3003

Выполнил:
Бобряков Кирилл, гр. № Р34122
Проверил: Усков Иван
Владимирович

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 `LabWork`, описание которого приведено ниже:

Вариант - 3003

```
public class LabWork {
    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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Integer minimalPoint; //Поле может быть null, Значение поля должно быть больше 0
    private Difficulty difficulty; //Поле может быть null
    private Discipline discipline; //Поле не может быть null
}

public class Coordinates {
    private long x;
    private int y; //Значение поля должно быть больше -895
}

public class Discipline {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Long lectureHours; //Поле может быть null
}

public enum Difficulty {
    VERY_EASY,
    NORMAL,
    IMPOSSIBLE;
}
```

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分页 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:

- Удалить один (любой) объект, значение поля difficulty которого эквивалентно заданному.
- Вернуть количество объектов, значение поля difficulty которых больше заданного.
- Вернуть массив объектов, значение поля name которых содержит заданную подстроку.

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 страничный 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 развернуты на сервере helios.

Ссылка на репозиторий:

Веб-сервис

<https://github.com/kekens/soa-back>

Клиент

<https://github.com/kekens/soa-front>

Спецификация API

1. Получить объект LabWork

GET /lab-work/{id}

Возвращает:

- 200 — OK: Возвращает объект
- 404 — NOT_FOUND: Объект с таким id не найден

2. Получить количество объектов LabWork, у которых свойство difficulty выше заданного

GET /lab-work/difficulty/count

Параметры:

- difficulty — Сложность лабораторной работы (Обязательный параметр)

Возвращает:

- 200 — OK: Возвращает количество объектов
- 400 — BAD_REQUEST: Неправильно задан параметр difficulty

3. Получить объекты LabWork, имя которых содержит заданную подстроку

GET /lab-work/name/substr

Параметры:

- name — Имя объекта (Обязательный параметр)

Возвращает:

- 200 — OK: Возвращает список объектов

4. Получить список объектов LabWork, применив фильтрацию

GET /lab-work/

Параметры:

- name — Имя объекта (Необязательный параметр)
- coordinates_x — Координата X (Необязательный параметр)
- coordinates_y — Координата Y (Необязательный параметр)
- minimalPoint — Минимальная оценка (Необязательный параметр)
- disciplineName — Имя дисциплины (Необязательный параметр)
- difficulty — Сложность лабораторной работы (Необязательный параметр)
- creationDate — Дата создания (Необязательный параметр)
- sort — Сортировка по параметру (Необязательный параметр)
- page — Индекс страницы (Необязательный параметр)
- count — Количество объектов на странице (Необязательный параметр)

Возвращает:

- 200 — OK: Возвращает список объектов
- 400 — BAD_REQUEST: Ошибка в параметре

5. Обновить объект LabWork

PUT /lab-work/{id}

Тело запроса:

LabWork — объект

Возвращает:

- 200 — OK
- 400 — BAD_REQUEST: Список ошибок валидации

- 404 — NOT_FOUND: Объект не найден

6. Добавить объект LabWork

POST /lab-work/

Тело запроса:

LabWork — объект

Возвращает:

- 200 — OK
- 400 — BAD_REQUEST: Список ошибок валидации

7. Удалить объект по ID LabWork

DELETE /lab-work/{id}

Возвращает:

- 200 — OK
- 404 — NOT_FOUND: Не найден объект с заданным id

8. Удалить случайный объект LabWork с заданной сложностью

DELETE /lab-work/difficulty

Параметры:

- difficulty — Сложность объекта (Обязательный параметр)

Возвращает:

- 200 — OK

- 400 — BAD_REQUEST: Неправильно задана сложность

9. Получить объект Discipline

GET /discipline/{id}

Возвращает:

- 200 — OK: Возвращает объект
- 404 — NOT_FOUND: Объект с таким id не найден

10. Обновить объект Discipline

PUT /discipline/{id}

Тело запроса:

Discipline — объект

Возвращает:

- 200 — OK
- 400 — BAD_REQUEST: Список ошибок валидации
- 404 — NOT_FOUND: Объект не найден

11. Добавить объект Discipline

POST /discipline/

Тело запроса:

Discipline — объект

Возвращает:

- 200 — OK
- 400 — BAD_REQUEST: Список ошибок валидации

11. Удалить объект по ID Discipline

DELETE /discipline/{id}

Возвращает:

- 200 — OK
- 404 — NOT_FOUND: Не найден объект с заданным id

12. Получить список объектов LabWork, применив фильтрацию

GET /lab-work/

Возвращает:

- 200 — OK: Возвращает список объектов