Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Технологии веб-сервисов

Лабораторная работа 5

Выполнили:
Кривоносов Егор Дмитриевич

Группа: P4214

Преподаватель:
Сафронов Андрей Геннадьевич

2025 г.

Санкт-Петербург

Оглавление

Техническое задание

Необходимо выполнить задание из второй работы, но с использованием REST-сервиса. Таблицу базы данных, а также код для работы с ней можно оставить без изменений.

Постановка задачи

Разработать REST-сервис для работы с сущностью Person с функционалом:

- Создание новой записи.
- Обновление существующей записи по id.
- Удаление записи по id.
- Предоставление данных в формате JSON.

Также реализовать клиент для взаимодействия с REST-сервисом.

Этапы выполнения

1. Реализован REST-контроллер `PersonRestController` с тремя новыми методами:
    a. Удаление записи по id;
    b. Создание новой записи;
    c. Обновление существующей записи по id;
2. Добавить новые методы на клиенте. Для вызова выше перечисленных функций и их отображения пользователю.

Комментарии к архитектурным и функциональным аспектам реализации

Контроллер:

1. Методы реализуют стандартные CRUD-операции.
2. Используются HTTP-методы `POST`, `PUT`, `DELETE`.
3. Данные проверяются перед записью в базу с помощью сервиса `PersonValidationService`.

```
@POST
public Response createPerson(PersonDto personDto) {
    PersonValidationService.validatePersonDto(personDto);
    int id = personService.createPerson(personDto);
```

```
        return Response.status(Response.Status.CREATED).entity(id).build();
    }

    @PUT
    @Path("/{id}")
    public Response updatePerson(@PathParam("id") int id, PersonDto personDto) {
        PersonValidationService.validatePersonDto(personDto);
        boolean updated = personService.updatePerson(id, personDto);
        return updated ? Response.ok().build() :
Response.status(Response.Status.NOT_FOUND).build();
    }

    @DELETE
    @Path("/{id}")
    public Response deletePersonById(@PathParam("id") int id) {
        boolean deleted = personService.deletePersonById(id);
        return deleted ? Response.ok().build() :
Response.status(Response.Status.NOT_FOUND).build();
    }
```

Rest-клиент:

1. Используются HTTP-запросы для взаимодействия с сервером.
2. Реализован функционал для поиска, создания, обновления и удаления записей.

```
    public int createPerson(PersonDto person) throws Exception {
        Response response = client.target(baseUrl)
                .path("/persons")
                .request(MediaType.APPLICATION_JSON)
                .post(Entity.entity(person, MediaType.APPLICATION_JSON));

        if (response.getStatus() == Response.Status.CREATED.getStatusCode()) {
            return response.readEntity(Integer.class);
        } else {
            throw new RuntimeException("Error creating person: " +
response.readEntity(String.class));
        }
    }

    public boolean updatePerson(int id, PersonDto person) throws Exception {
        Response response = client.target(baseUrl)
                .path("/persons/{id}")
                .resolveTemplate("id", id)
                .request(MediaType.APPLICATION_JSON)
                .put(Entity.entity(person, MediaType.APPLICATION_JSON));
        if (response.getStatus() == Response.Status.OK.getStatusCode()) {
            return true;
        } else if (response.getStatus() == Response.Status.NOT_FOUND.getStatusCode()) {
            return false;
        } else {
            throw new RuntimeException("Error updating person: " +
response.readEntity(String.class));
        }
    }

    public boolean deletePerson(int id) throws Exception {
        Response response = client.target(baseUrl)
                .path("/persons/{id}")
                .resolveTemplate("id", id)
                .request(MediaType.APPLICATION_JSON)
                .delete();
```

```
        if (response.getStatus() == Response.Status.OK.getStatusCode()) {
            return true;
        } else if (response.getStatus() == Response.Status.NOT_FOUND.getStatusCode()) {
            return false;
        } else {
            throw new RuntimeException("Error deleting person: " +
response.readEntity(String.class));
        }
    }
```

Ссылка на код

https://github.com/RedGry/TVS-LABS/tree/lab456

Вывод

В рамках работы был реализован REST-сервис с клиентом для работы с сущностью Person. Сервис поддерживает полный набор CRUD-операций и предоставляет данные в формате JSON. Клиент позволяет удобно взаимодействовать с сервисом из сторонних приложений.