Комитет по образованию Правительства Санкт-Петербурга

**САНКТ-ПЕТЕРБРУГСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ   
ТЕХНОЛОГИЙ**

**Отчет по практической работе  
МДК 01.02 «Разработка мобильных приложений»  
Разработка интерактивного графического приложения   
c использованием REST API**

Выполнил

студент группы 493:

Лукьянов И. А.

Преподаватель: Фомин А.В.

Санкт-Петербург 2022

Структура базы данных

База данных состоит из 5 таблиц:

1. account – хранит учетные записи пользователей.
2. graph – хранит графы пользователей.
3. link – хранит связи узлов.
4. node – хранит узлы графа.
5. session – хранит активные сессии пользователей.

ER диаграмма представлена на рисунке 1.

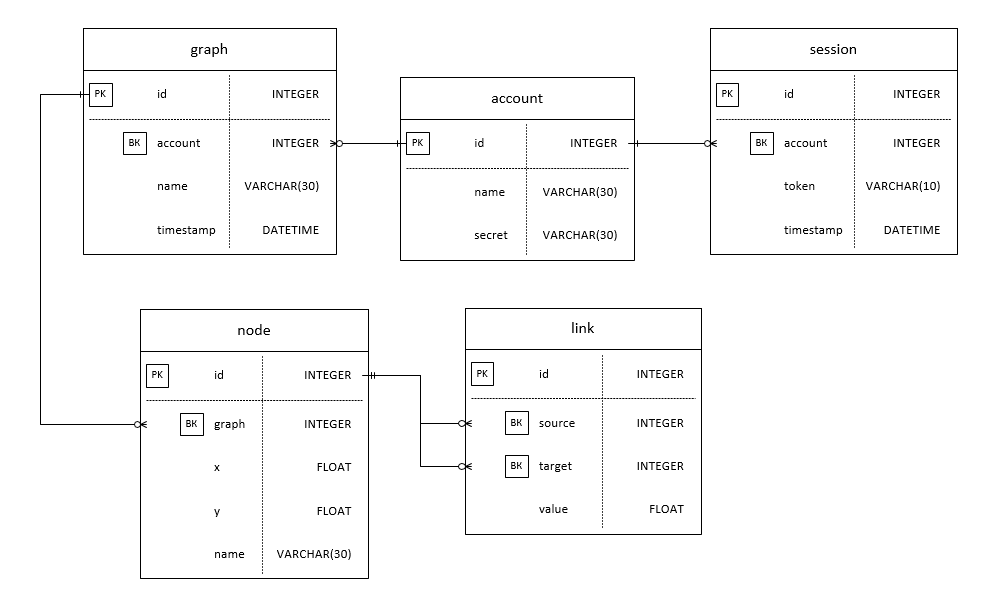


Рисунок – ER диаграмма базы данных

**Таблица account**

Содержит сведения о учетных записях пользователей. Таблица состоит из трёх столбцов:

1. id – уникальный идентификатор.
2. name – имя пользователя.
3. secret – пароль пользователя.

Подробное описание столбцов представлено на рисунке 2.

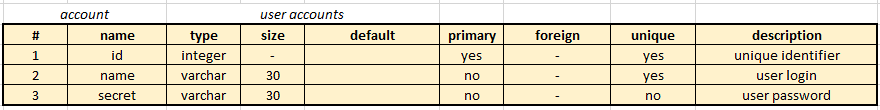


Рисунок – Описание столбцов таблицы account

Пример данных представлен на рисунке 3.

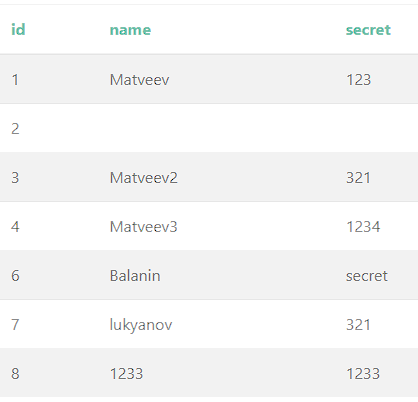


Рисунок – Пример данных таблицы account

**Таблица graph**

Содержит сведения о графах пользователя. Таблица состоит из четырех столбцов:

1. id – уникальный идентификатор.
2. account – идентификатор пользователя, которому принадлежит граф.
3. name – имя графа.
4. timestamp – дата и время создания графа.

Подробное описание столбцов представлено на рисунке 4.

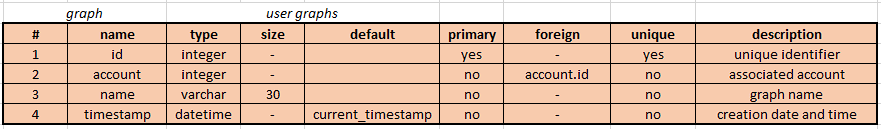


Рисунок – Описание столбцов таблицы graph

Пример данных представлен на рисунке 5.

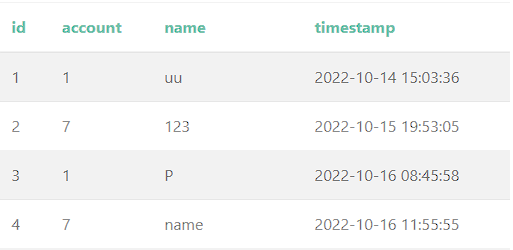


Рисунок – Пример данных таблицы graph

**Таблица node**

Содержит сведения об узлах графа. Таблица состоит из пяти столбцов:

1. id – уникальный идентификатор.
2. graph – номер графа, на котором находится узел.
3. x – координата по горизонтали.
4. y – координата по вертикали.
5. name – имя узла.

Подробное описание столбцов представлено на рисунке 6.

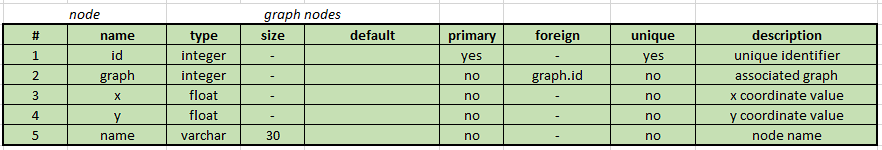


Рисунок – Описание столбцов таблицы node

Пример данных представлен на рисунке 7.

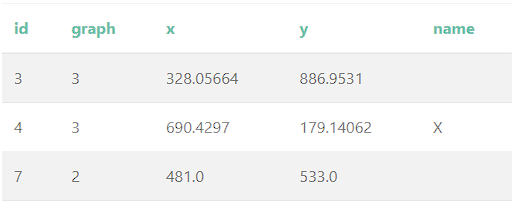


Рисунок – Пример данных таблицы node

**Таблица link**

Содержит сведения о связях узлов. Таблица состоит из четырех столбцов:

1. id – уникальный идентификатор.
2. source – идентификатор узла, от которого происходит связь.
3. target – идентификатор узла, к которому происходит связь.
4. value – значение связи.

Подробное описание столбцов представлено на рисунке 8.

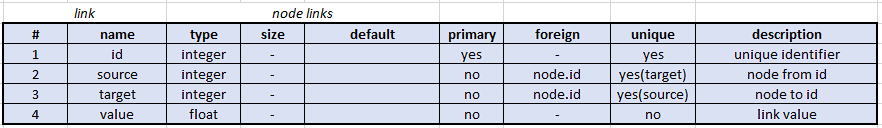


Рисунок – Описание столбцов таблицы link

Пример данных представлен на рисунке 9.

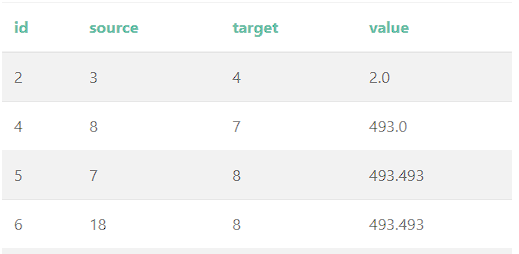


Рисунок – Пример данных таблицы link

**Таблица session**

Содержит сведения об активных сессиях пользователей. Таблица состоит из четырех столбцов:

1. id – уникальный идентификатор.
2. account – идентификатор пользователя, открывшего сессию.
3. token – уникальный ключ сессии.
4. timestamp – дата и время открытия сессии.

Подробное описание столбцов представлено на рисунке 10.

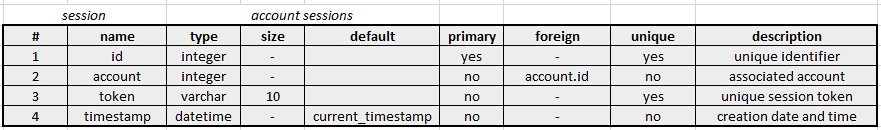


Рисунок – Описание столбцов таблицы session

Пример данных представлен на рисунке 11.

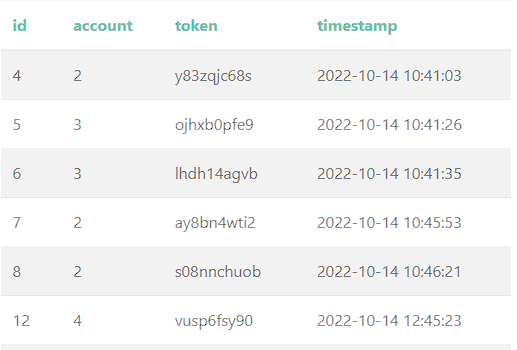


Рисунок – Пример данных таблицы session

Интерфейс приложения

Приложение состоит из 7 форм:

1. Auth Form: стартовая форма, служит для прохождения авторизации и открытия сессии.
2. Reg Form: служит для регистрации новых пользователей.
3. Main Form: служит для управления узлами и связями, а также переходу к форме графов.
4. Graph Form: форма, на которой находятся сохраненные и загруженные с REST API графы с возможностями сохранения, загрузки, переименования и удаления графа.
5. Session Form: форма, на которой можно изменить пароль пользователя, выбрать конечную точку API, выйти из текущей сессии, просмотреть список и закрыть выбранную сессию и удалить аккаунт.
6. Node: форма для задания свойств узла, а именно имени и координат.
7. Link: форма для задания значения связи.

**Форма Auth Form**

На рисунке 12 показан макет внешнего вида формы авторизации.

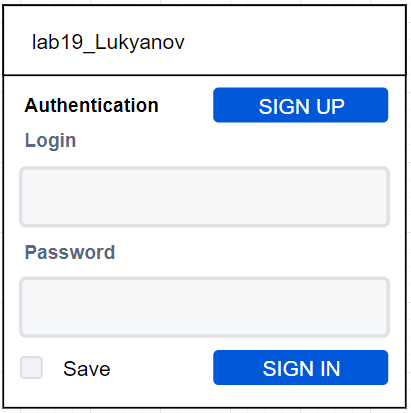


Рисунок – Макет формы Auth

На рисунке 13 показан внешний вид формы авторизации в приложении.

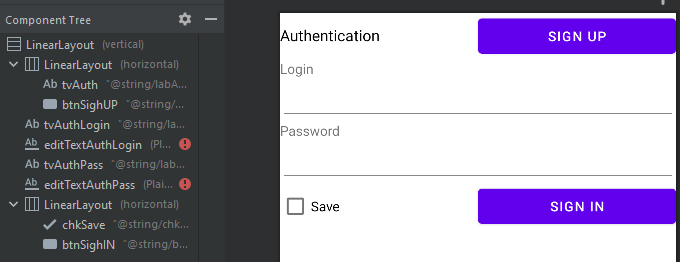


Рисунок – Форма Auth в приложении

**Форма Reg Form**

На рисунке 14 показан макет внешнего вида формы регистрации.

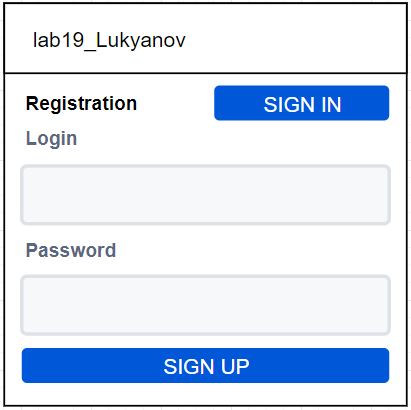


Рисунок – Макет формы Reg

На рисунке 15 показан внешний вид формы регистрации в приложении.

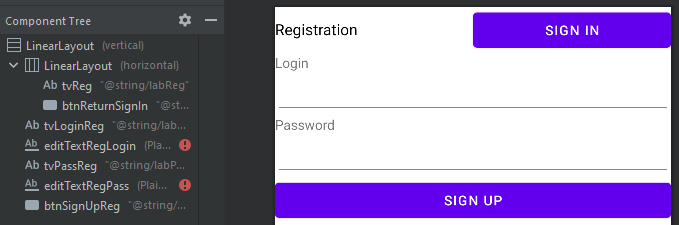


Рисунок – Форма Reg в приложении

**Форма Main Menu**

На рисунке 16 показан макет внешнего вида главной формы.

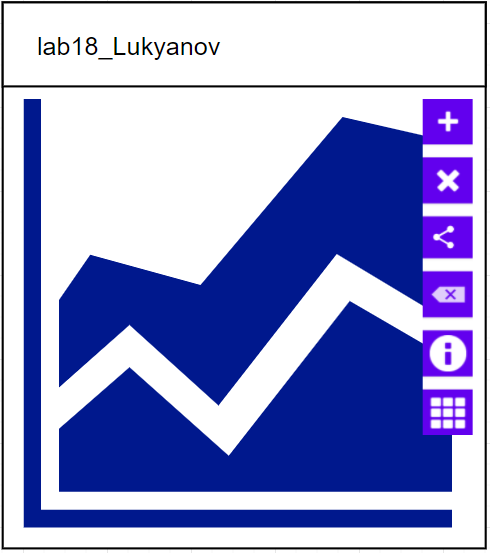


Рисунок 16 – Макет формы Main

На рисунке 17 показан внешний вид формы главного меню в приложении.

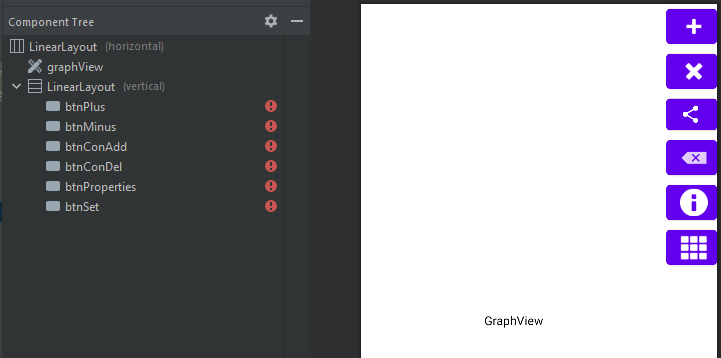


Рисунок – Форма Main Menu в приложении

**Форма Graph**

На рисунке 19 показан макет внешнего вида формы графов.

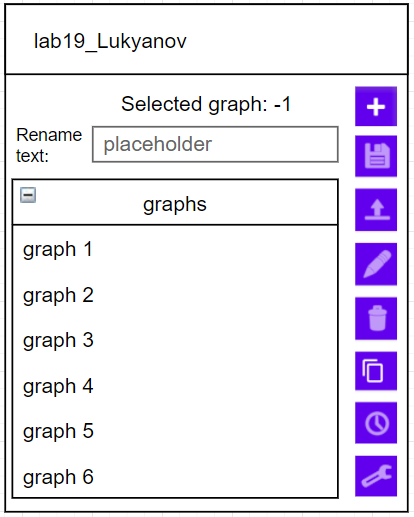


Рисунок 19 – Макет формы Graph

На рисунке 20 показан внешний вид формы графов в приложении.

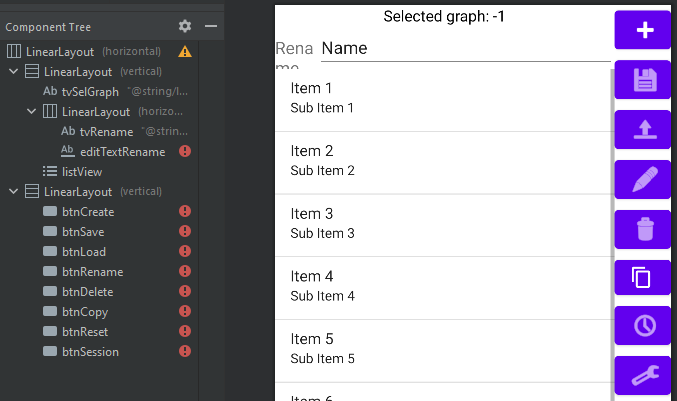


Рисунок 20 – Форма Graph в приложении

**Форма Session**

На рисунке 21 показан макет внешнего вида формы сессий.

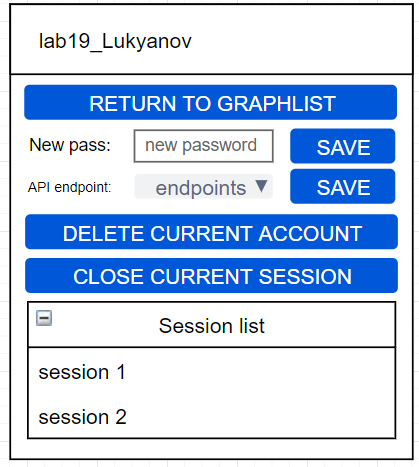


Рисунок 21 – Макет формы Session

На рисунке 22 показан внешний вид формы сессий в приложении.

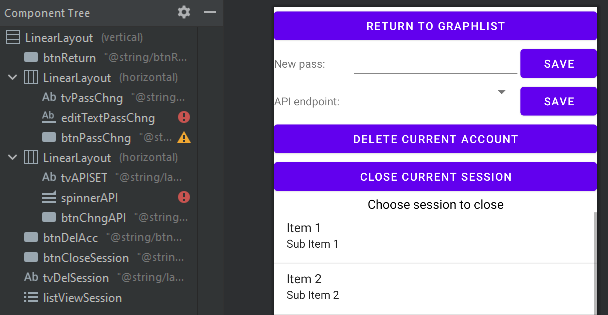


Рисунок 22 – Форма Session в приложении

**Форма Node**

На рисунке 23 показан макет внешнего вида формы узла.

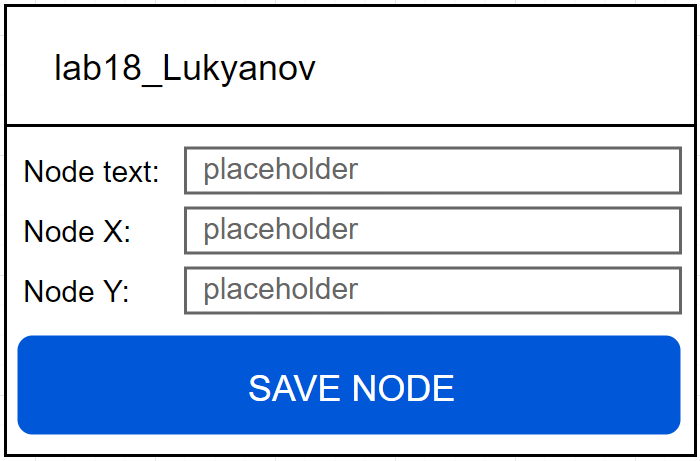


Рисунок 23 – Макет формы Node

На рисунке 24 показан внешний вид формы сообщения в приложении.

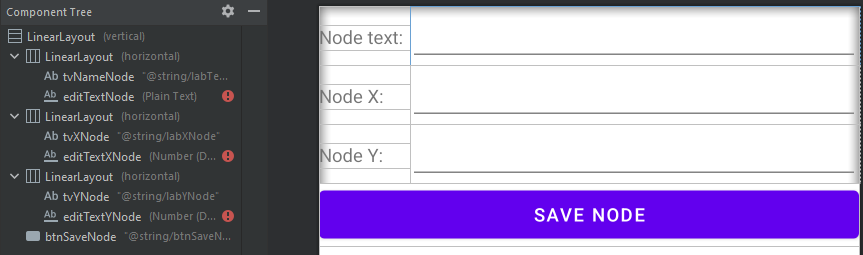


Рисунок 24 – Форма Node в приложении

**Форма Link**

На рисунке 25 показан макет внешнего вида формы связи.

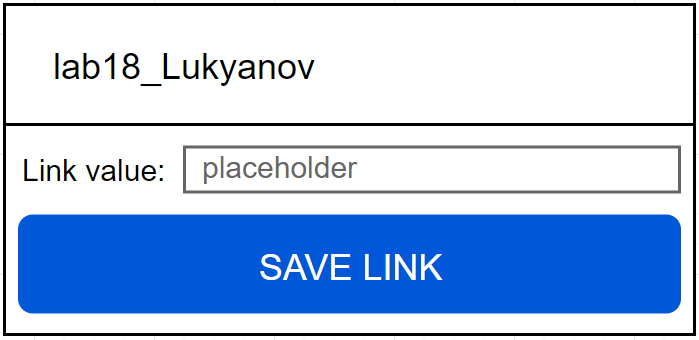


Рисунок 25 – Макет формы Link

На рисунке 26 показан внешний вид формы связи в приложении.

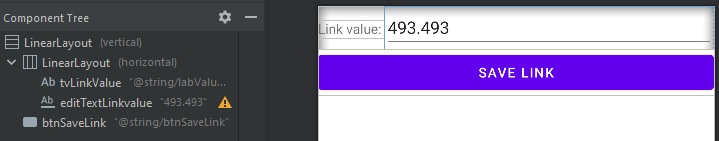


Рисунок 26 – Форма Link в приложении

Описание протокола взаимодействия

Для взаимодействия с базой данных мобильное приложение использует API на основе хранимых процедур и сервера, который организует вызов процедур по протоколу HTTP и обмен данными.

Всего доступно 18 функций, список которых представлен на рисунке 27.

1. /account/create – создание новой учетной записи.
2. /account/delete – удаление учетной записи.
3. /account/update – смена пароля учетной записи.
4. /session/close – закрытие сессии.
5. /session/list – вывод списка сессий пользователя.
6. /session/open – открытие сессии.
7. /graph/create – создание графа.
8. /graph/delete – удаление графа.
9. /graph/list – вывод списка графов пользователя.
10. /graph/update – смена имени графа.
11. /node/create – создание узла.
12. /node/delete – удаление узла.
13. /node/list – вывод списка узлов графа.
14. /node/update – обновление параметров узла.
15. /link/create – создание связи.
16. /link/delete – удаление связи.
17. /link/list – вывод списка связей графа.
18. /link/update – изменение значения связи.

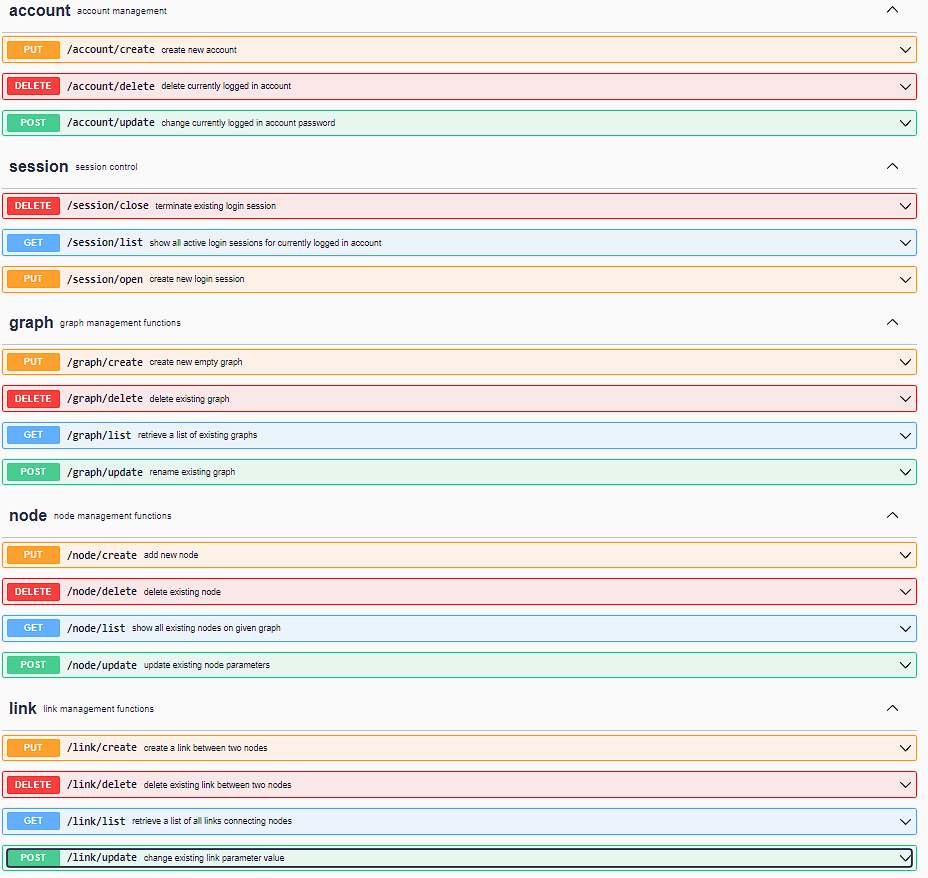


Рисунок 27 – Список функций

1. **Управление учетными записями**

Данный набор функций позволяет управлять учетными записями пользователей.

* 1. **Функция /account/create**

Обеспечивает возможность создания пользователя с использованием имени и пароля. Далее происходит сохранение его в таблицу account.

Входные параметры:

* name – имя пользователя (текстовая строчка)
* secret – пароль учетной записи (текстовая строчка)

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 28.

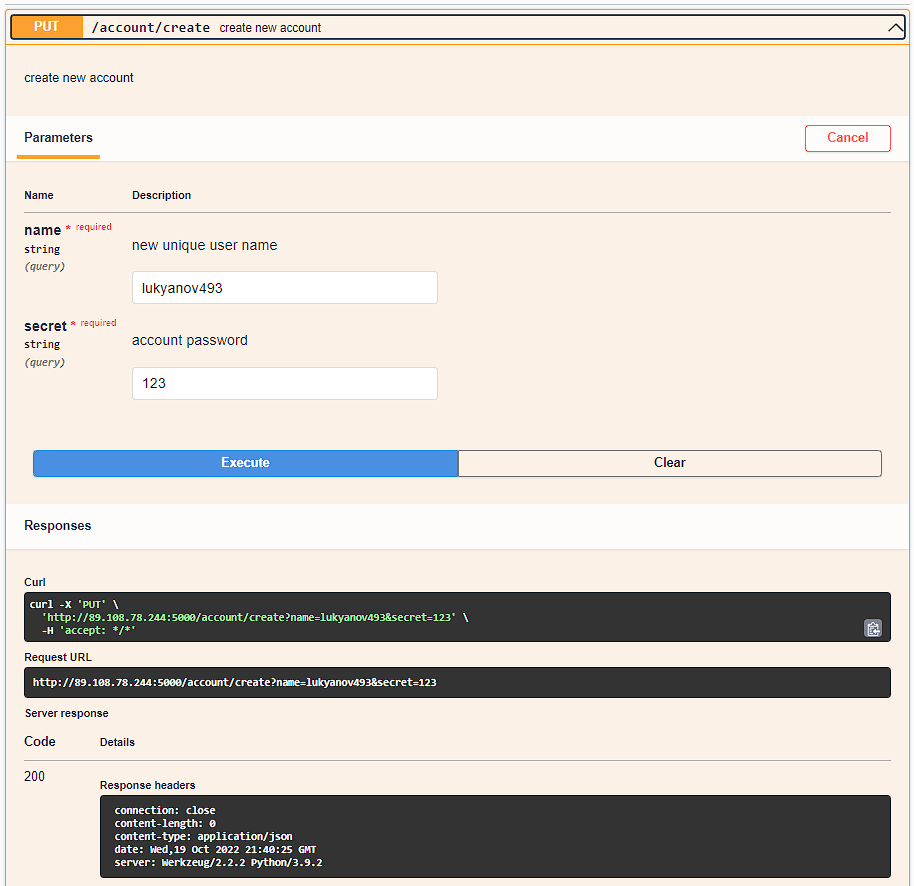


Рисунок 28 – Вызов функции /account/create

* 1. **Функция /account/delete**

Обеспечивает возможность удаления пользователя с использованием ключа сессии открытой этим пользователем. Далее происходит удаление его из таблицы account.

Входные параметры:

* token – ключ сессии.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 29.

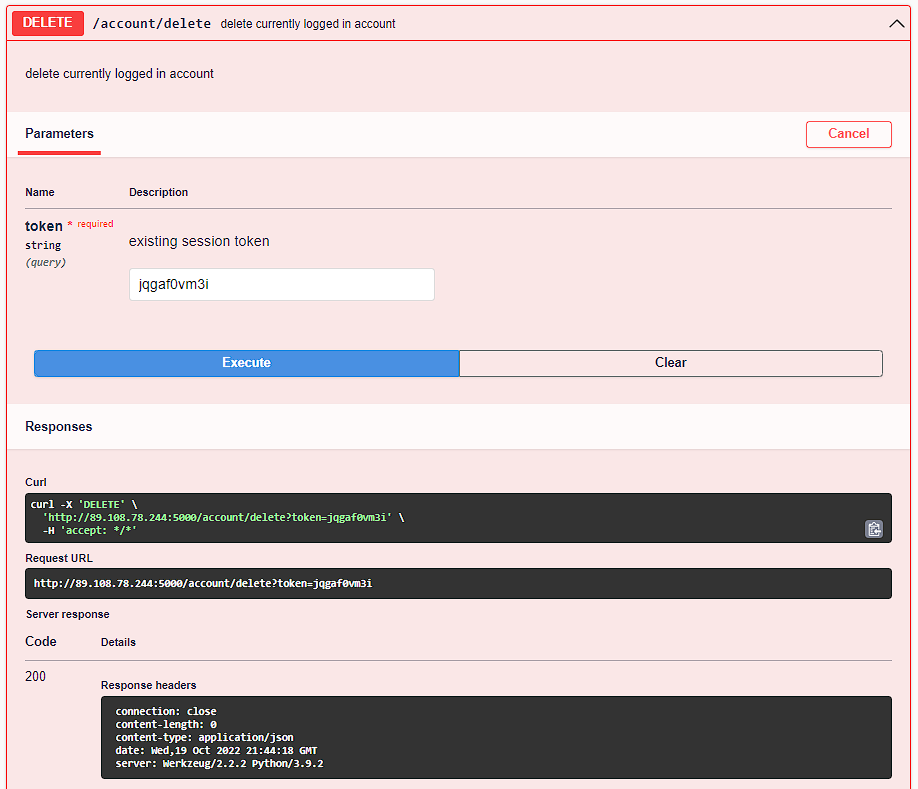


Рисунок 29 – Вызов функции /account/delete

* 1. **Функция /account/update**

Обеспечивает возможность изменения пароля пользователя с использованием ключа сессии открытой этим пользователем. Далее происходит изменения поля secret в таблице account.

Входные параметры:

* token – ключ сессии.
* secret – новый пароль учетной записи (текстовая строчка).

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 30.

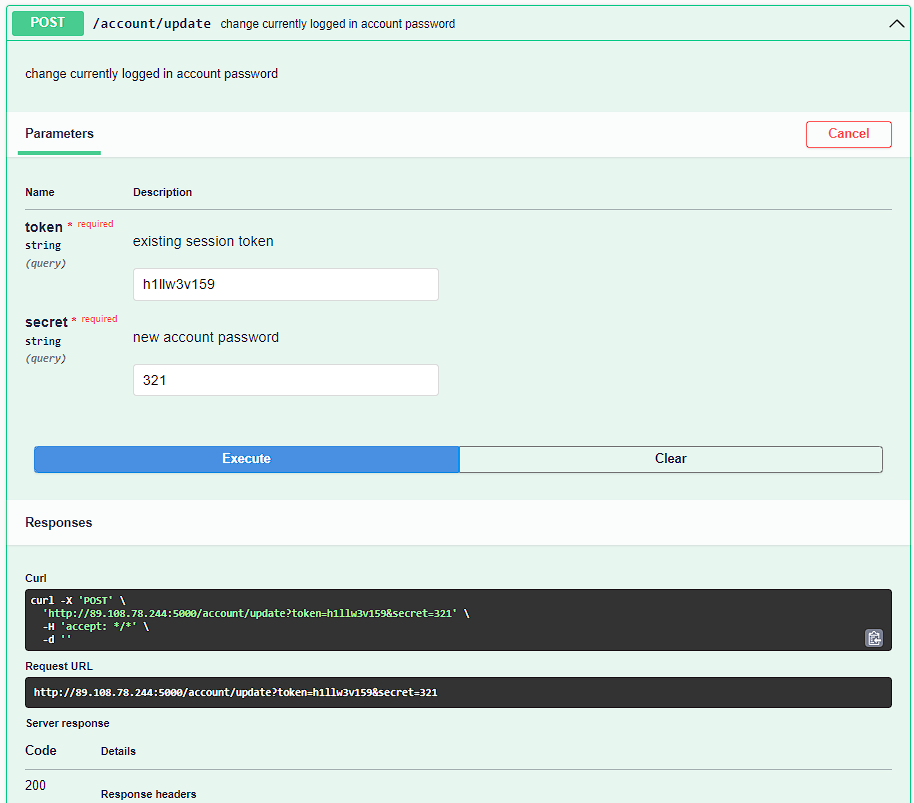


Рисунок 30 – Вызов функции /account/update

1. **Управление сессиями пользователей**

Данный набор функций позволяет управлять сессиями пользователей.

* 1. **Функция /session/close**

Обеспечивает возможность закрытие сессии пользователя с использованием ключа сессии. Далее происходит удаление его из таблицы session.

Входные параметры:

* token – ключ сессии.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 31.

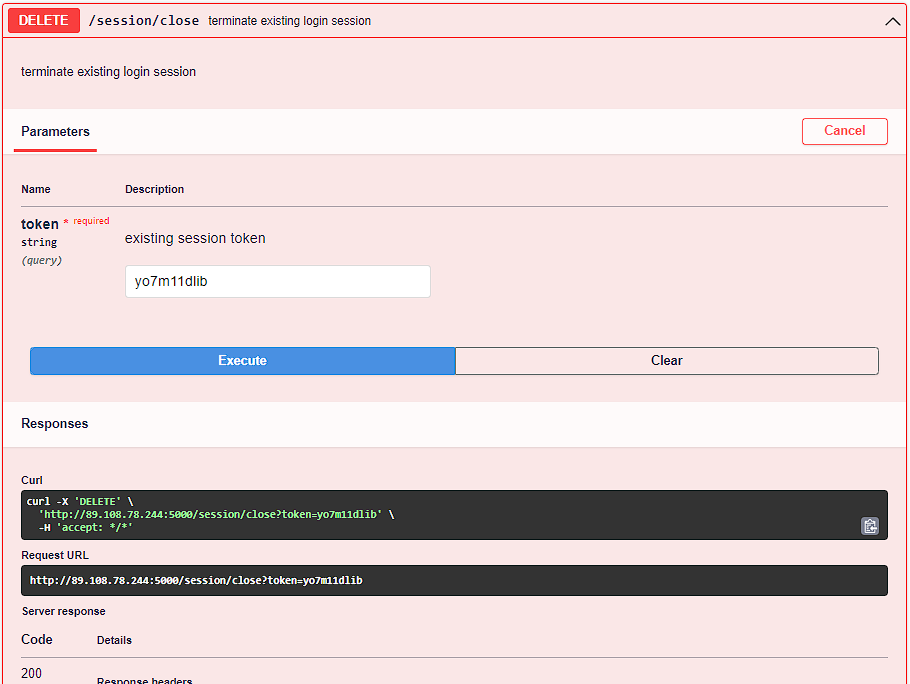


Рисунок 31 – Вызов функции /session/close

* 1. **Функция /session/list**

Обеспечивает возможность вывода списка сессий пользователя с использованием ключа сессии открытой этим пользователем.

Входные параметры:

* token – ключ сессии.

Выходные параметры:

* массив сессий с их идентификатором, ключом и временем создания.

Пример вызова функции показан на рисунке 32.

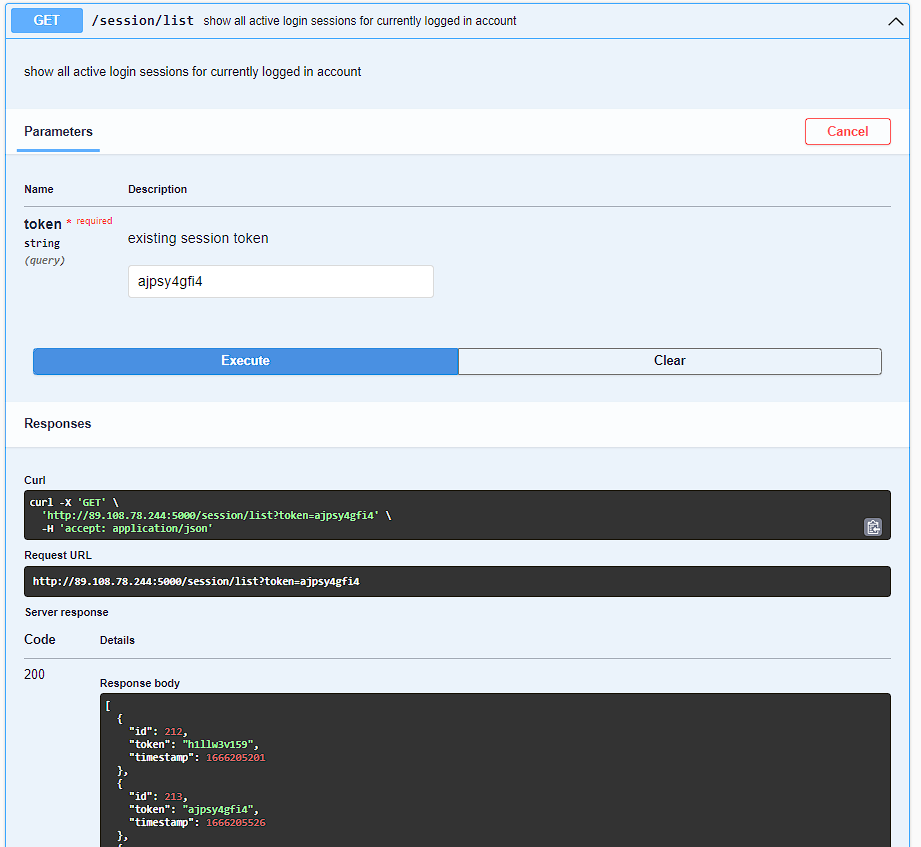


Рисунок 32 – Вызов функции /session/list

* 1. **Функция /session/open**

Обеспечивает возможность открытия сессии пользователя с использованием логина и пароля. Далее происходит генерация случайного ключа сессии и сохранение его в таблицу session.

Входные параметры:

* name – имя пользователя (текстовая строчка).
* secret – пароль учетной записи (текстовая строчка).

Выходные параметры:

* token: уникальный ключ сессии для данного пользователя.

Пример вызова функции показан на рисунке 33.

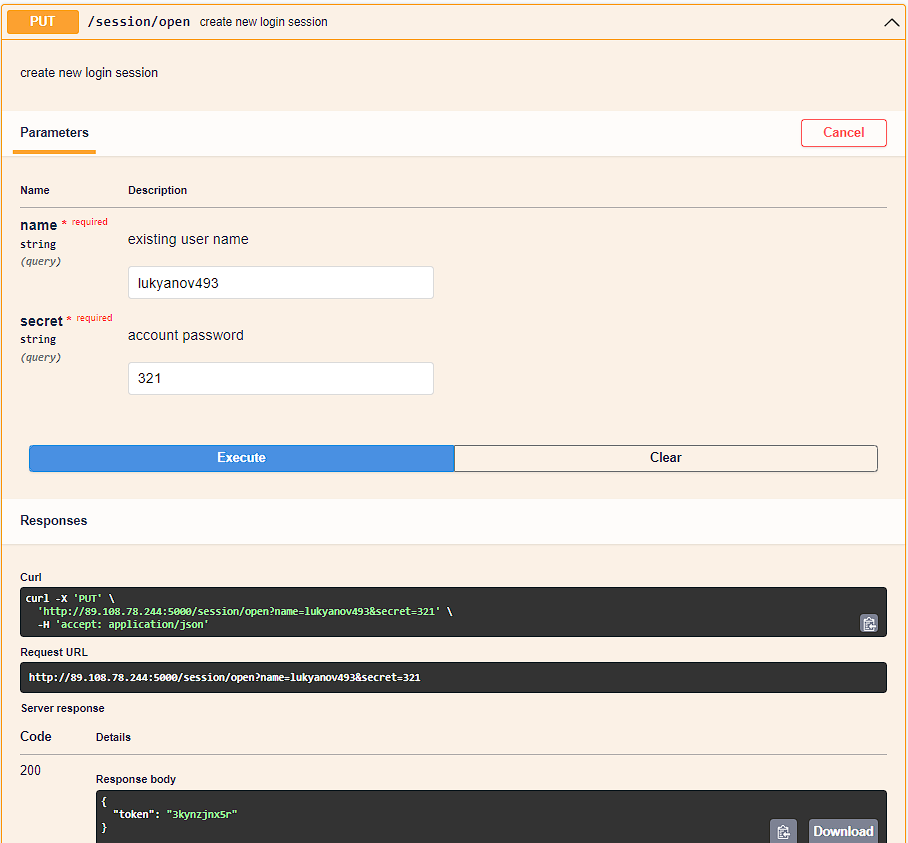


Рисунок 33 – Вызов функции /session/open

1. **Управление графами пользователей**

Данный набор функций позволяет управлять графами пользователей.

* 1. **Функция /graph/create**

Обеспечивает возможность создания графа пользователя с использованием ключа сессии и имени графа. Далее происходит создание его в таблице graph.

Входные параметры:

* token – ключ сессии.
* name – имя графа.

Выходные параметры:

* id – идентификатор созданного графа.

Пример вызова функции показан на рисунке 34.

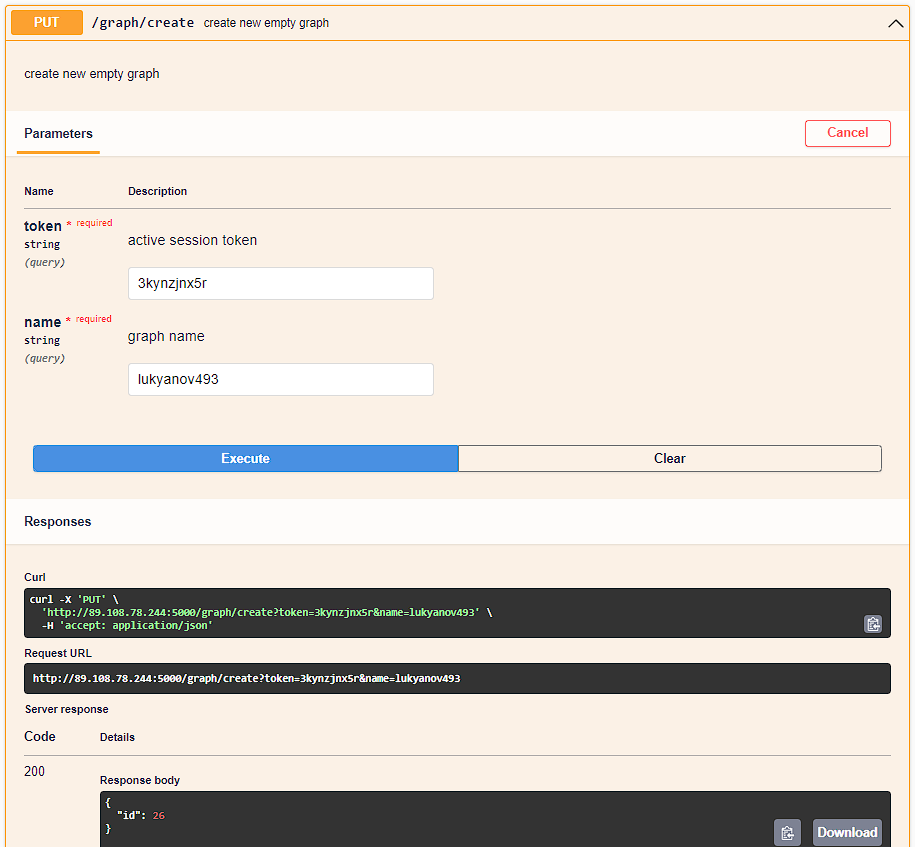


Рисунок 34 – Вызов функции /graph/create

* 1. **Функция /graph/list**

Обеспечивает возможность вывода списка графов пользователя с использованием ключа сессии открытой этим пользователем.

Входные параметры:

* token – ключ сессии.

Выходные параметры:

* массив графов с их идентификатором, временем создания и количеством узлов.

Пример вызова функции показан на рисунке 35.

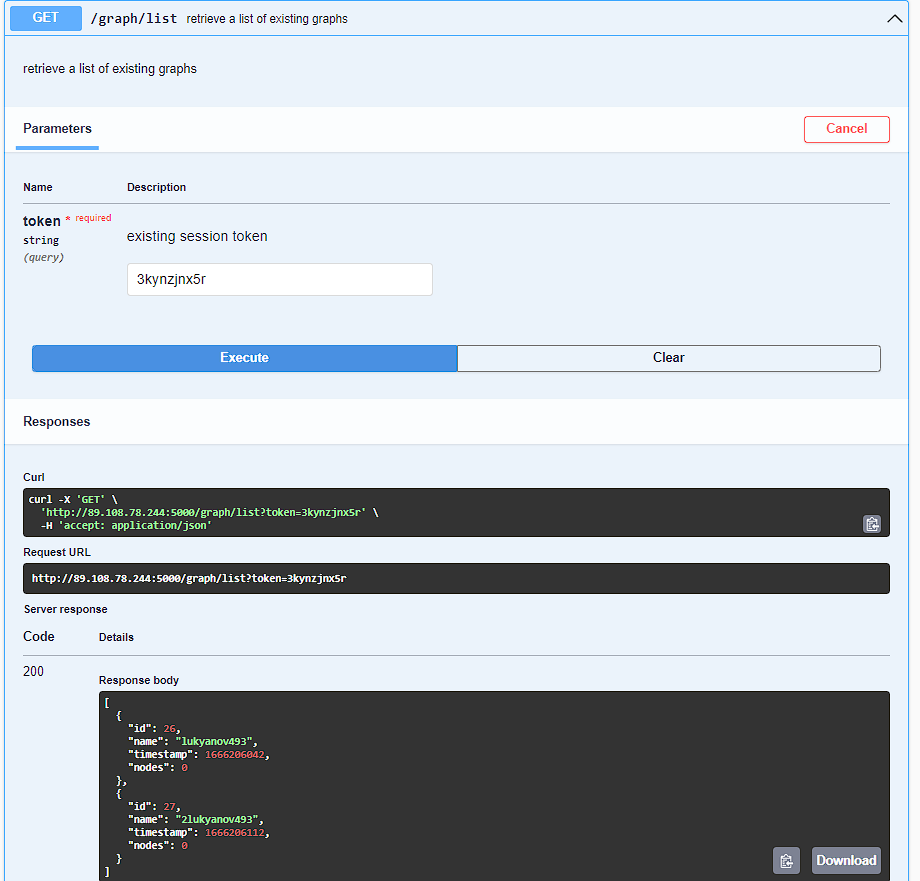


Рисунок 35 – Вызов функции /graph/list

* 1. **Функция /graph/delete**

Обеспечивает возможность удаление графа пользователя с использованием ключа сессии и идентификатора графа. Далее происходит удаление графа из таблицы graph.

Входные параметры:

* token – ключ сессии.
* id – идентификатор графа.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 36.

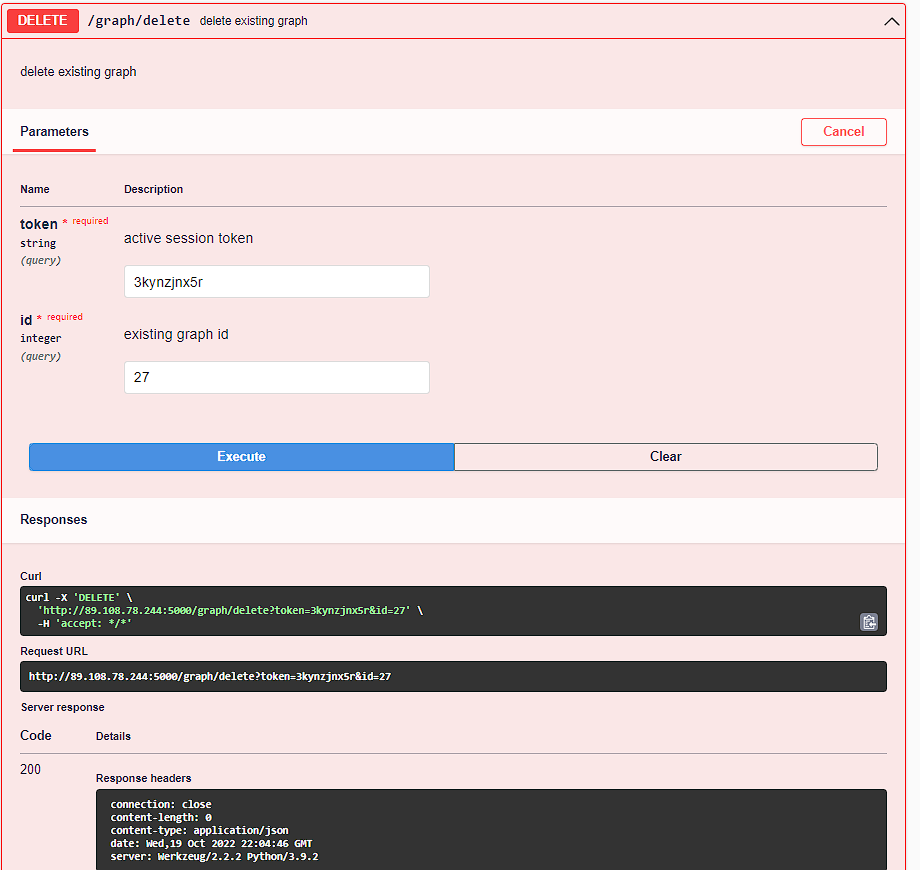


Рисунок 36 – Вызов функции /graph/delete

* 1. **Функция /graph/update**

Обеспечивает возможность изменения имени графа пользователя с использованием ключа сессии, идентификатора графа и нового имени. Далее происходит обновление поля name графа из таблицы graph.

Входные параметры:

* token – ключ сессии.
* id – идентификатор графа.
* name – новое имя графа.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 37.

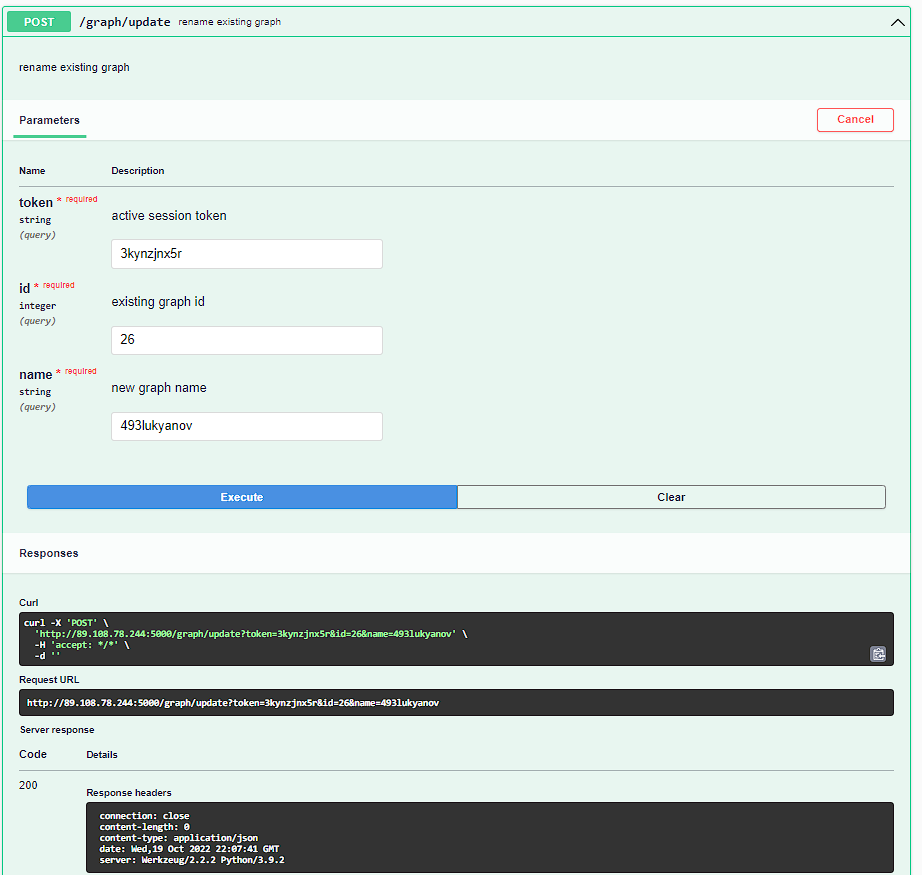


Рисунок 37 – Вызов функции /graph/update

1. **Управление узлами графов**

Данный набор функций позволяет управлять узлами графов пользователей.

* 1. **Функция /node/create**

Обеспечивает возможность создания узла графа пользователя с использованием ключа сессии, идентификатора графа и параметров узла. Далее происходит создание его в таблице node.

Входные параметры:

* token – ключ сессии.
* id – идентификатор графа.
* x – координата узла по горизонтали.
* y – координата узла по вертикали.
* name – имя узла

Выходные параметры:

* id – идентификатор созданного узла.

Пример вызова функции показан на рисунке 38.

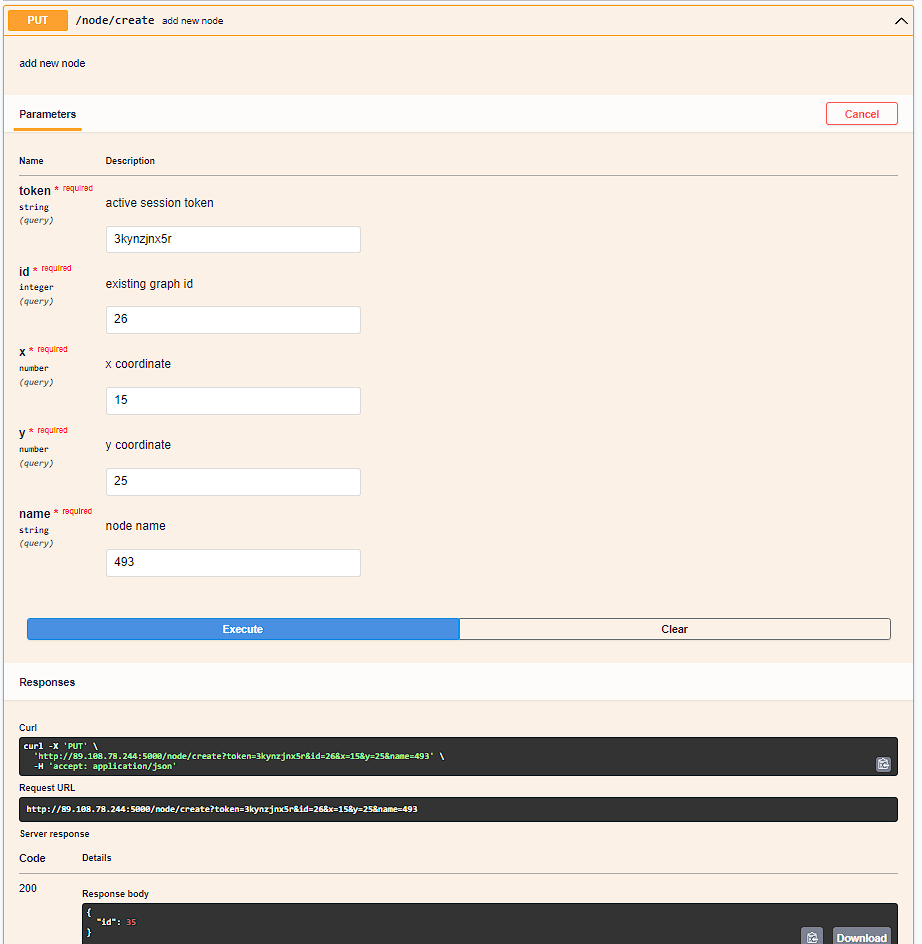


Рисунок 38 – Вызов функции /node/create

* 1. **Функция /node/list**

Обеспечивает возможность вывода списка узлов графа пользователя с использованием ключа сессии открытой этим пользователем и идентификатора графа.

Входные параметры:

* token – ключ сессии.
* id – идентификатор графа.

Выходные параметры:

* массив узлов с их идентификатором и всеми параметрами.

Пример вызова функции показан на рисунке 39.

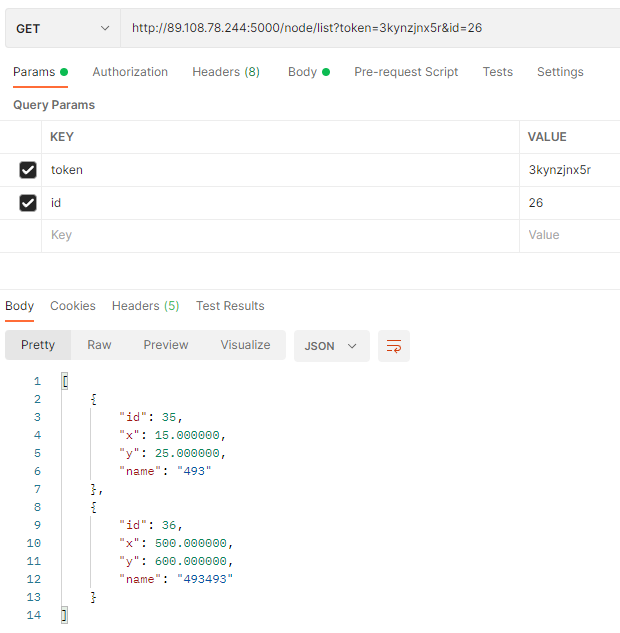


Рисунок 39 – Вызов функции /node/list

* 1. **Функция /node/delete**

Обеспечивает возможность удаления узла графа пользователя с использованием ключа сессии и идентификатора узла. Далее происходит удаление узла из таблицы node.

Входные параметры:

* token – ключ сессии.
* id – идентификатор узла.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 40.

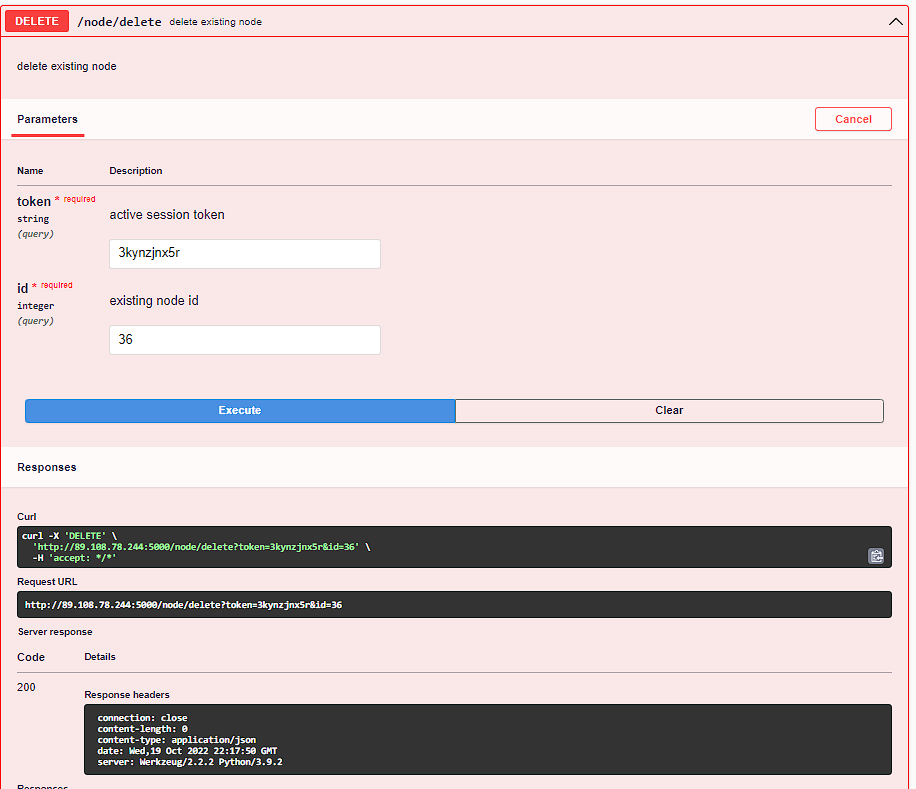


Рисунок 40 – Вызов функции /node/delete

* 1. **Функция /node/update**

Обеспечивает возможность изменения параметров узла графа с использованием ключа сессии, идентификатора узла и новых параметров. Далее происходит обновление всех полей кроме идентификатора в таблице node.

Входные параметры:

* token – ключ сессии.
* id – идентификатор узла.
* x – новая координата по горизонтали.
* y – новая координата по вертикали.
* name – новое имя узла.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 41.

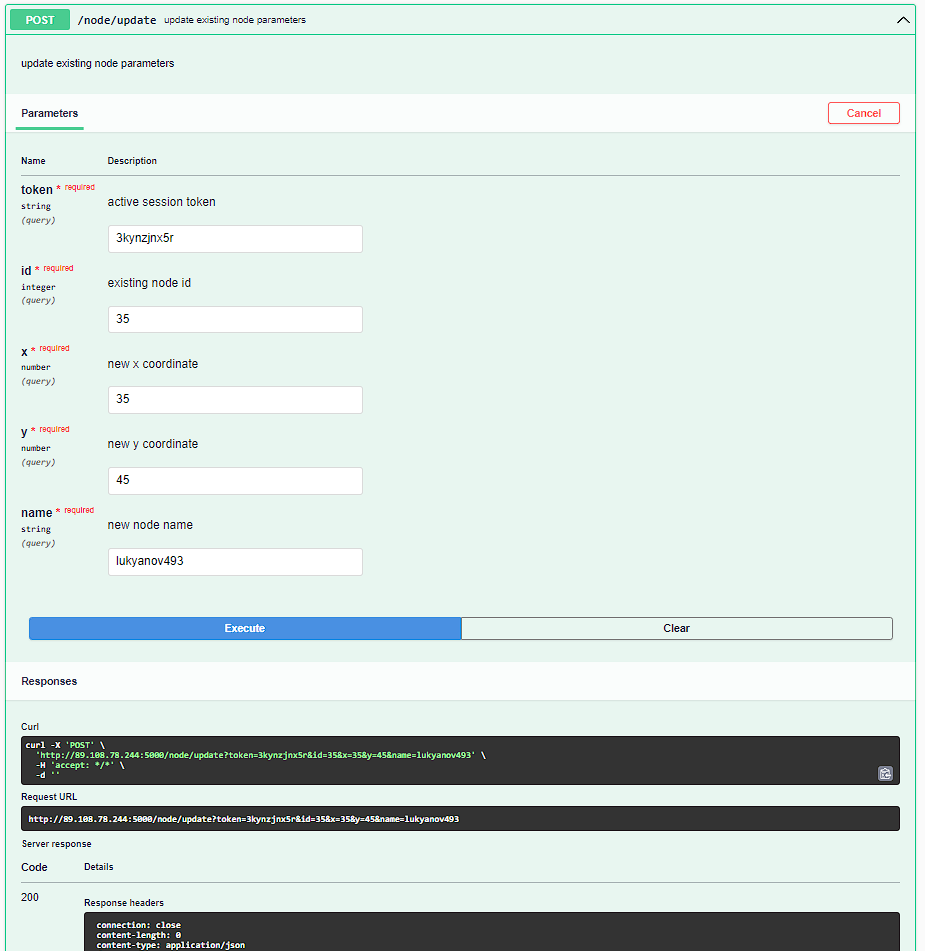


Рисунок 41 – Вызов функции /node/update

1. **Управление связями узлов графов**

Данный набор функций позволяет управлять связями узлов графов пользователей.

* 1. **Функция /link/create**

Обеспечивает возможность создания связи между двумя узлами графа пользователя с использованием ключа сессии, идентификаторов узлов и значения связи. Далее, если связи с такими же идентификаторами не существует, то происходит создание её в таблице link.

Входные параметры:

* token – ключ сессии.
* source – идентификатор узла, от которого пойдет связь.
* target – идентификатор узла, к которому пойдет связь.
* value – значение связи.

Выходные параметры:

* id – идентификатор созданной связи.

Пример вызова функции показан на рисунке 42.

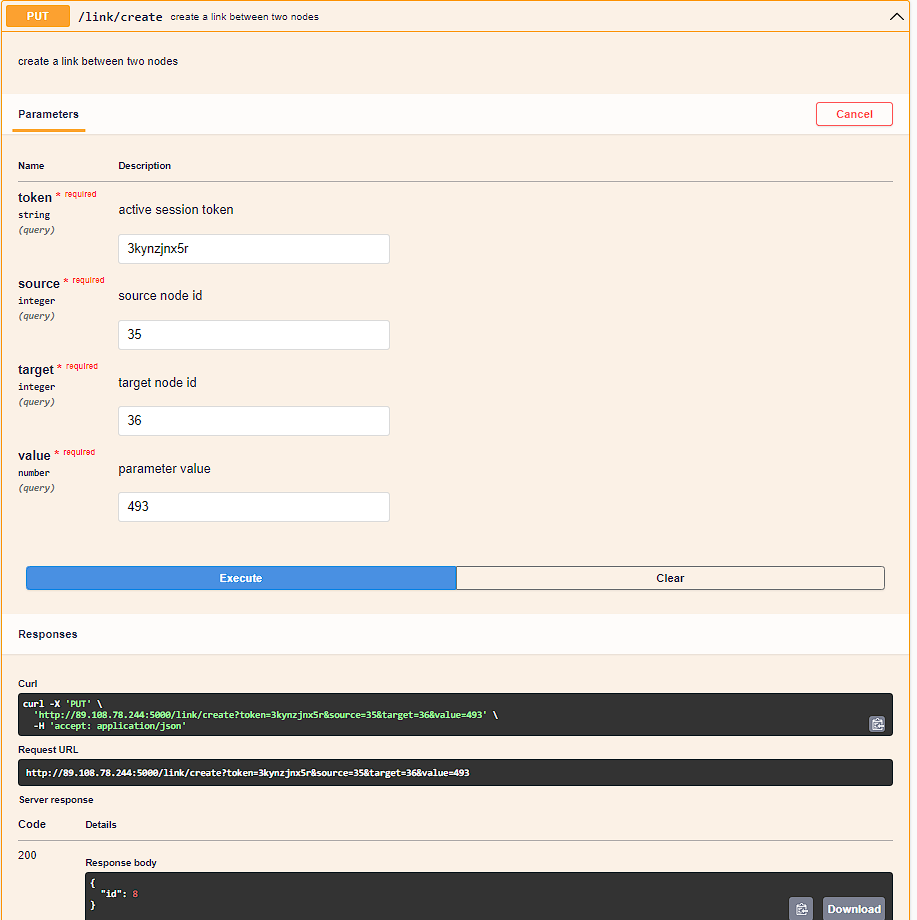


Рисунок 42 – Вызов функции /link/create

* 1. **Функция /link/list**

Обеспечивает возможность вывода списка связей графа пользователя с использованием ключа сессии открытой этим пользователем и идентификатора графа.

Входные параметры:

* token – ключ сессии.
* id – идентификатор графа.

Выходные параметры:

* массив связей с их идентификатором и всеми параметрами.

Пример вызова функции показан на рисунке 43.

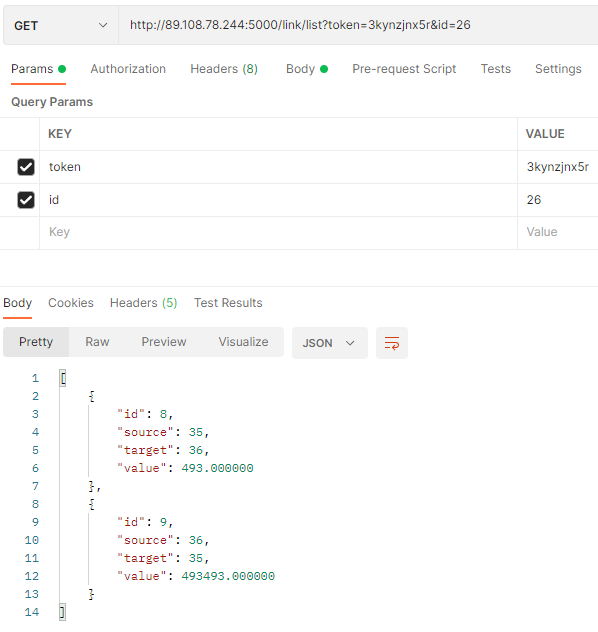


Рисунок 43 – Вызов функции /link/list

* 1. **Функция /link/delete**

Обеспечивает возможность удаления связи графа пользователя с использованием ключа сессии и идентификатора связи. Далее происходит удаление связи из таблицы link.

Входные параметры:

* token – ключ сессии.
* id – идентификатор связи.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 44.

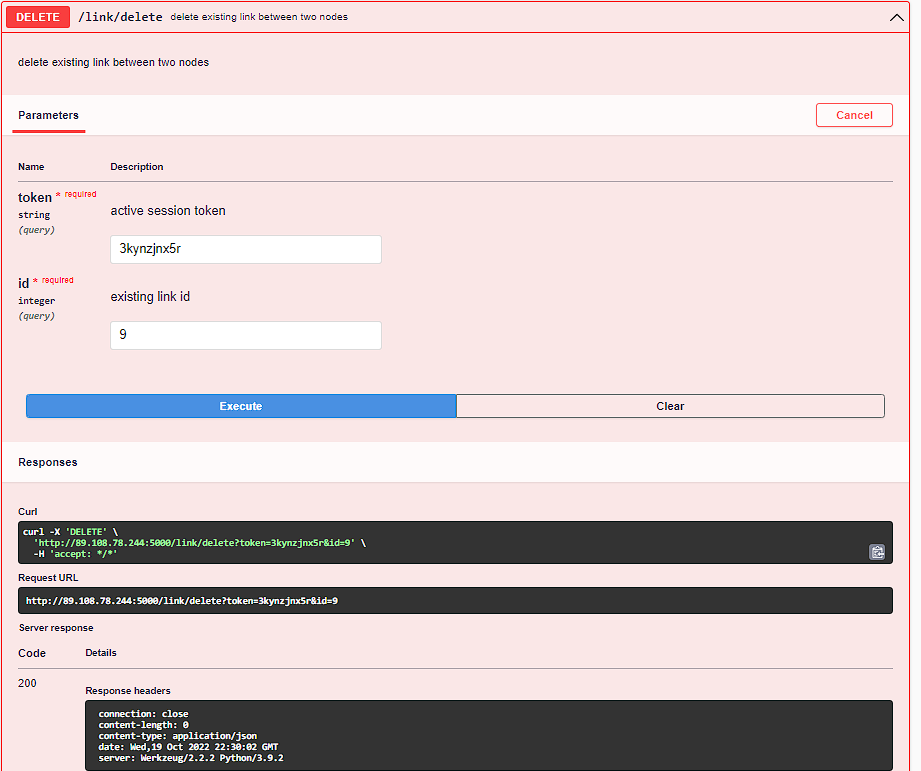


Рисунок 44 – Вызов функции /link/delete

* 1. **Функция /link/update**

Обеспечивает возможность изменения значения связи графа с использованием ключа сессии, идентификатора связи и нового значения. Далее происходит обновление поля value в таблице link.

Входные параметры:

* token – ключ сессии.
* id – идентификатор узла.
* value – новое значение связи.

Выходные параметры:

* код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 45.

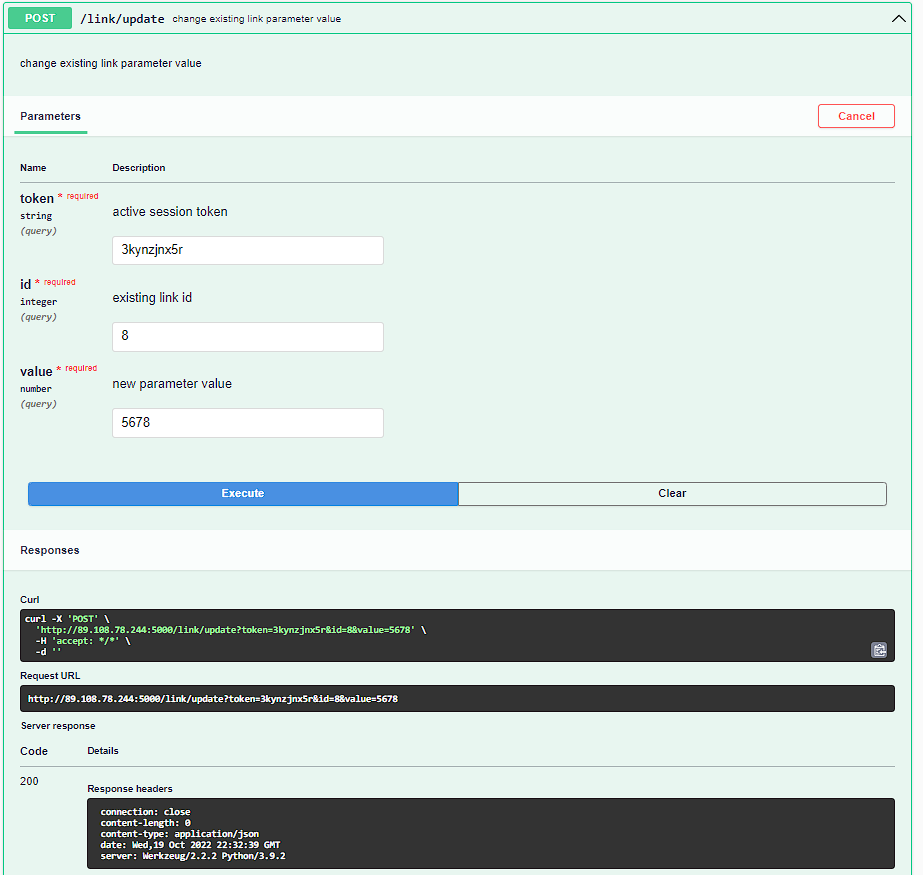


Рисунок 45 – Вызов функции /link/update

Демонстрация работы приложения

Авторизация (рис. 46 и 47). При установке флажка данные сохраняются. Также при запуске приложения производится проверка на то, открыта ли предыдущая сессия, и если открыта, то экран входа пропускается:



Рисунок 46 – До авторизации



Рисунок 47 – После авторизации

Регистрация с автоматическим входом (рис. 47 и 48):

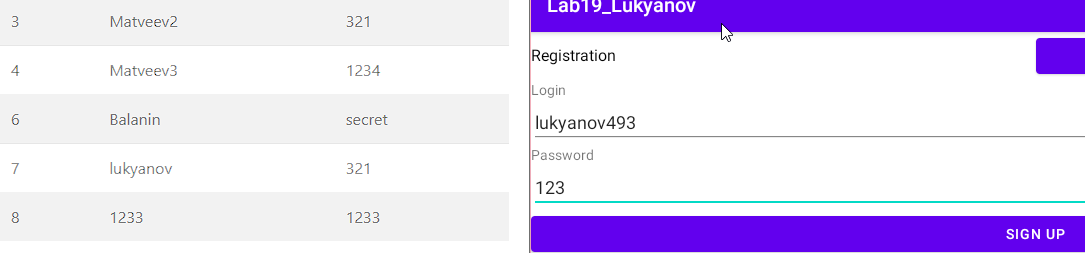


Рисунок 47 – До регистрации

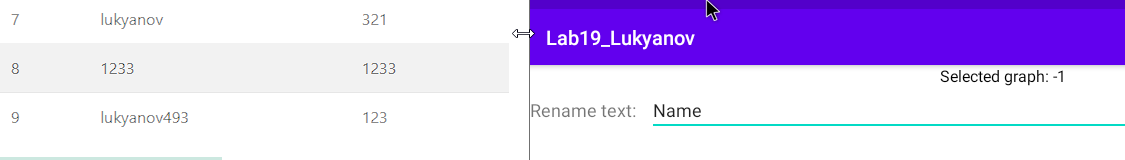


Рисунок 48 – После регистрации

Создание нового графа (рис. 49 и 50):

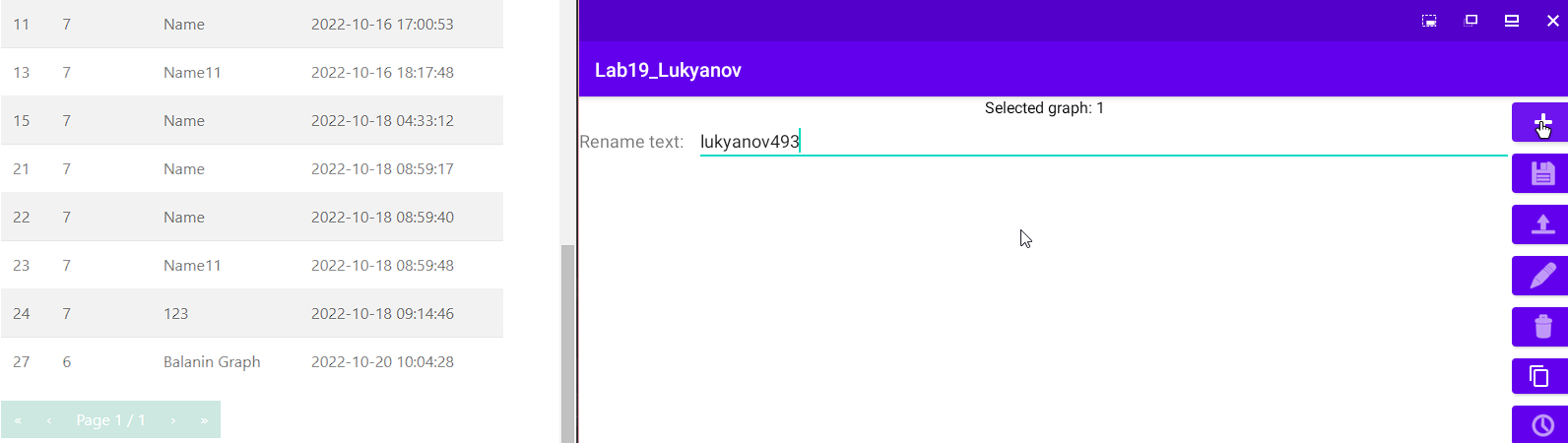


Рисунок 49 – До создания графа

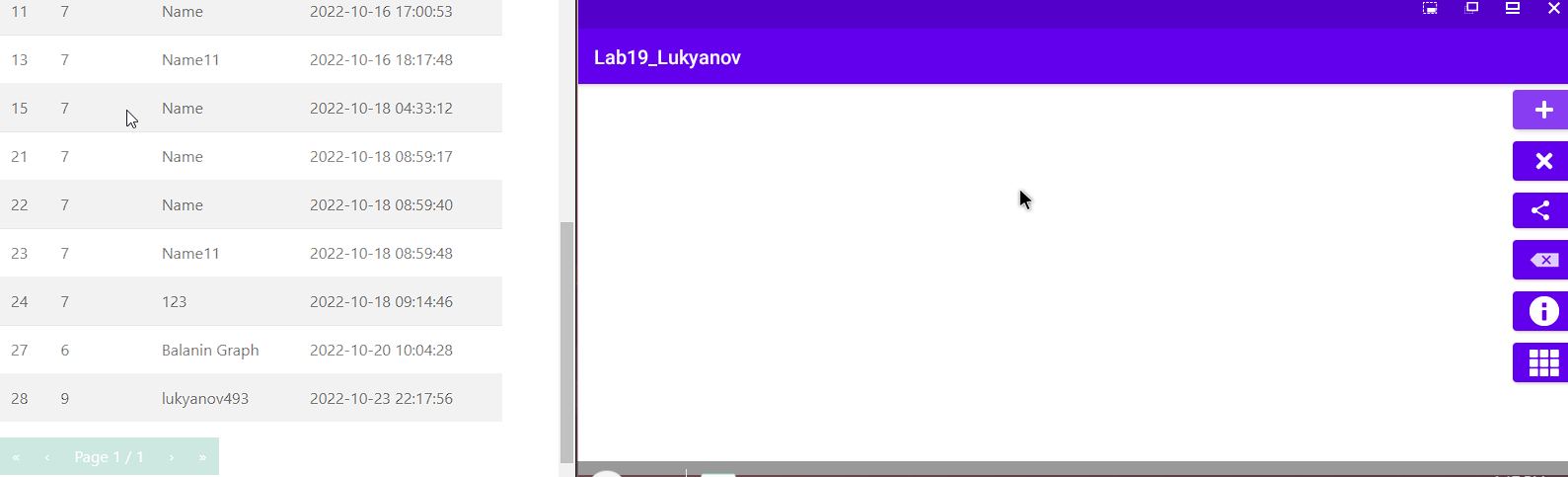


Рисунок 50 – После создания графа

Создание нового узла (рис. 51 и 52):

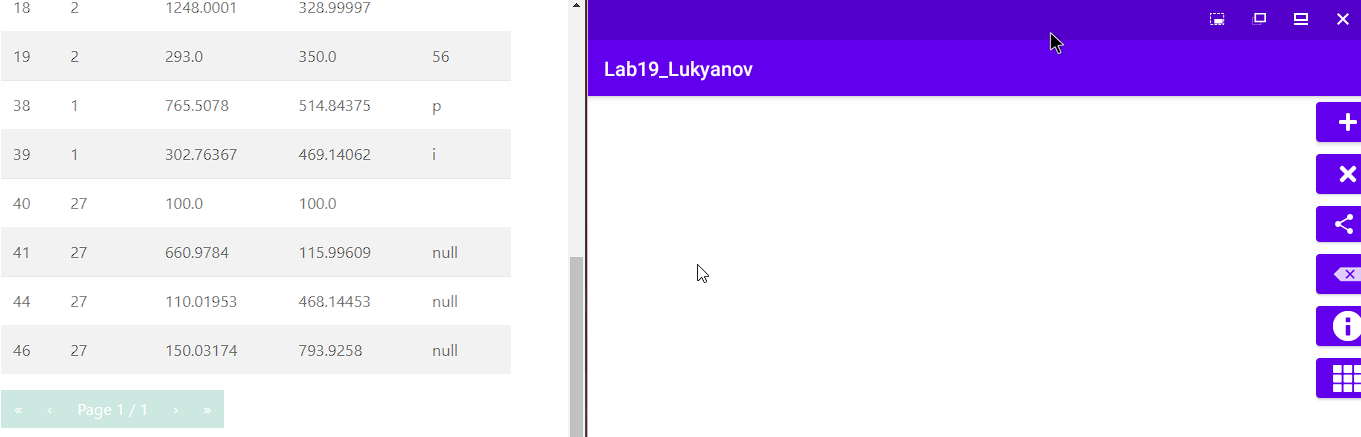


Рисунок 51 – До создания узла

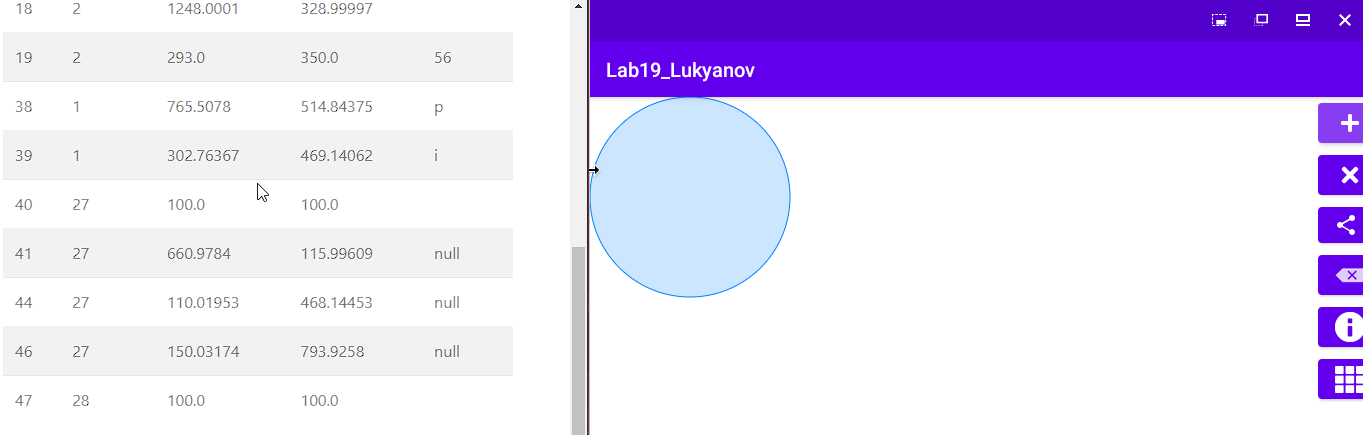


Рисунок 52 – После создания узла

Перемещение узла (рис. 53):

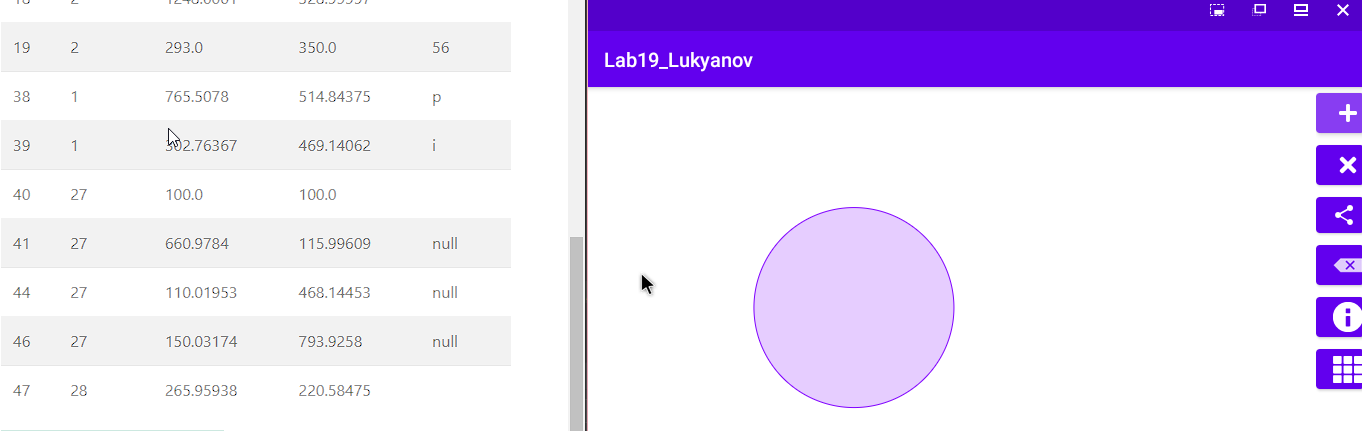


Рисунок 53 – Перемещение узла

Изменение имени и координат узла (рис. 54 и 55):

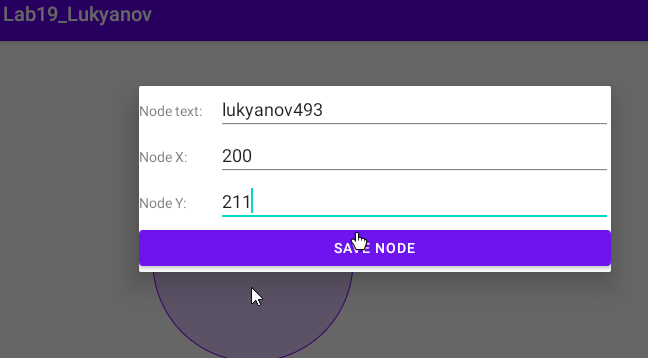


Рисунок 54 – До изменения узла

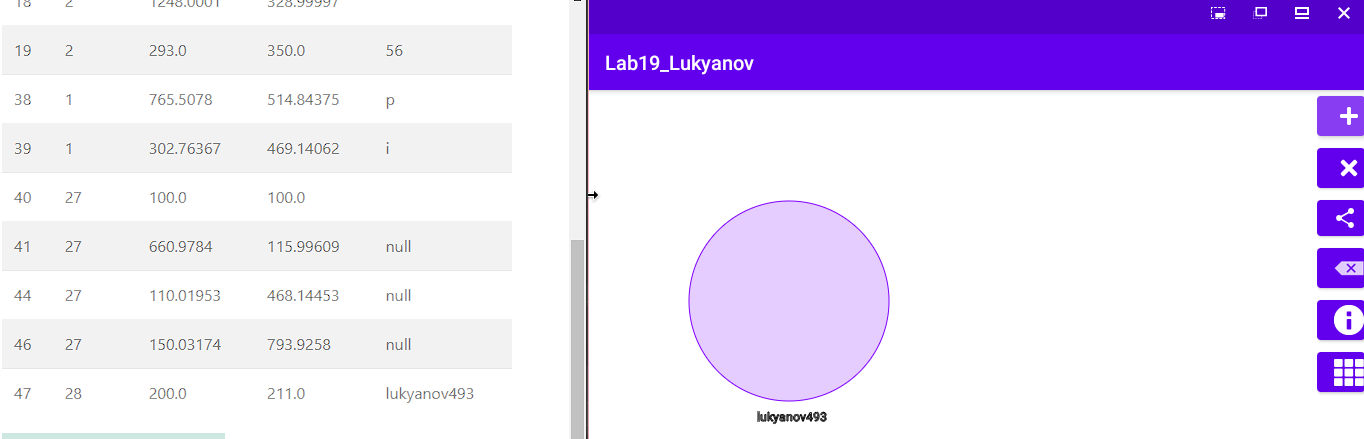


Рисунок 55 – После изменения узла

Создание связи между узлами (рис. 56 и 57):

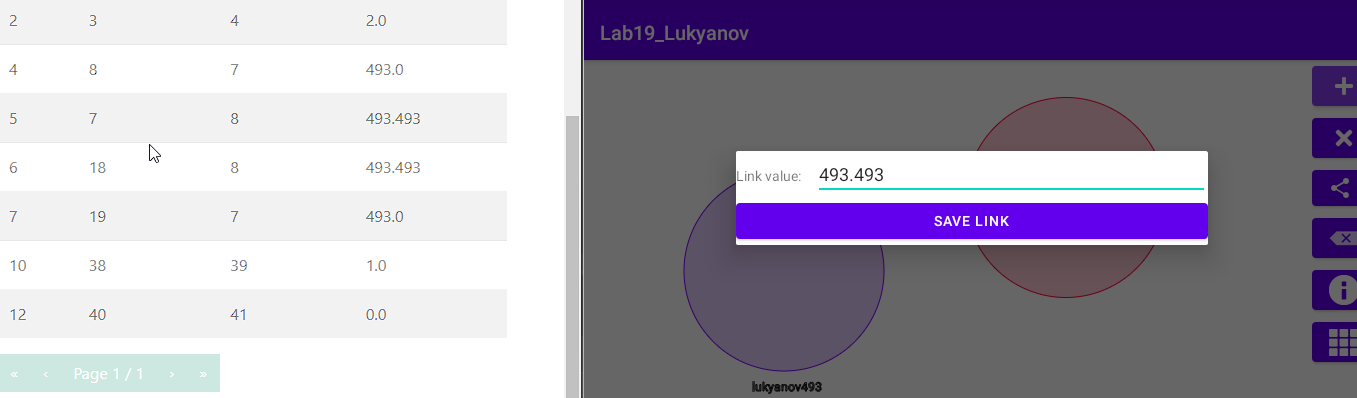


Рисунок 56 – До создания связи

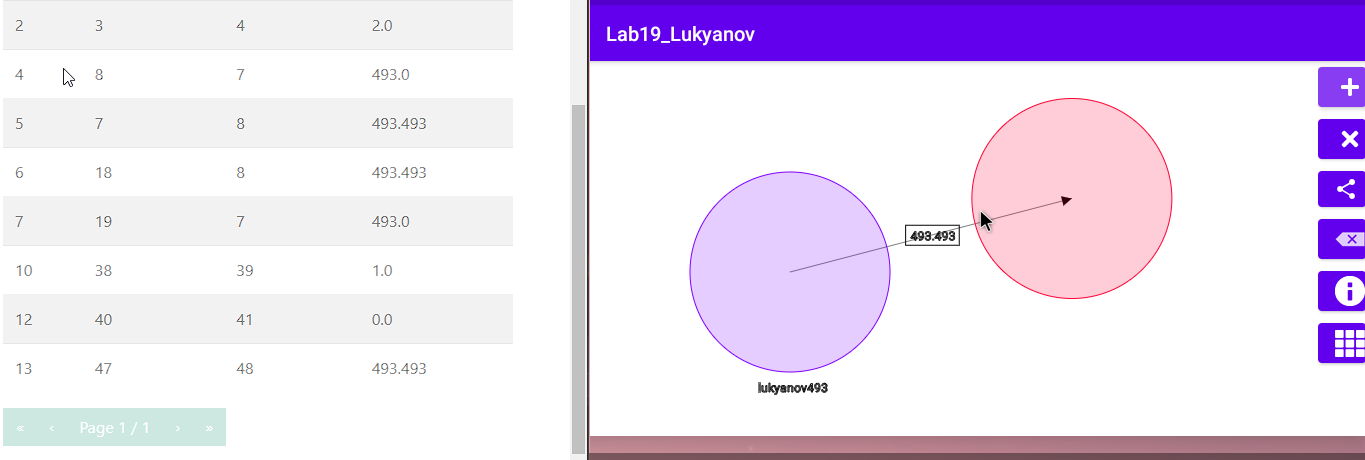


Рисунок 57 – После создания связи

Изменение значения связи (рис. 58 и 59):

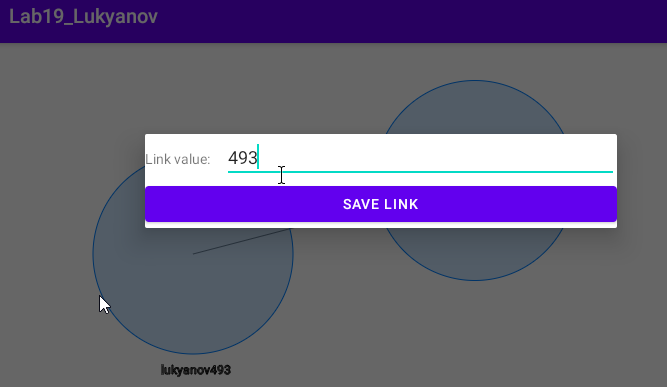


Рисунок 58 – До изменения связи

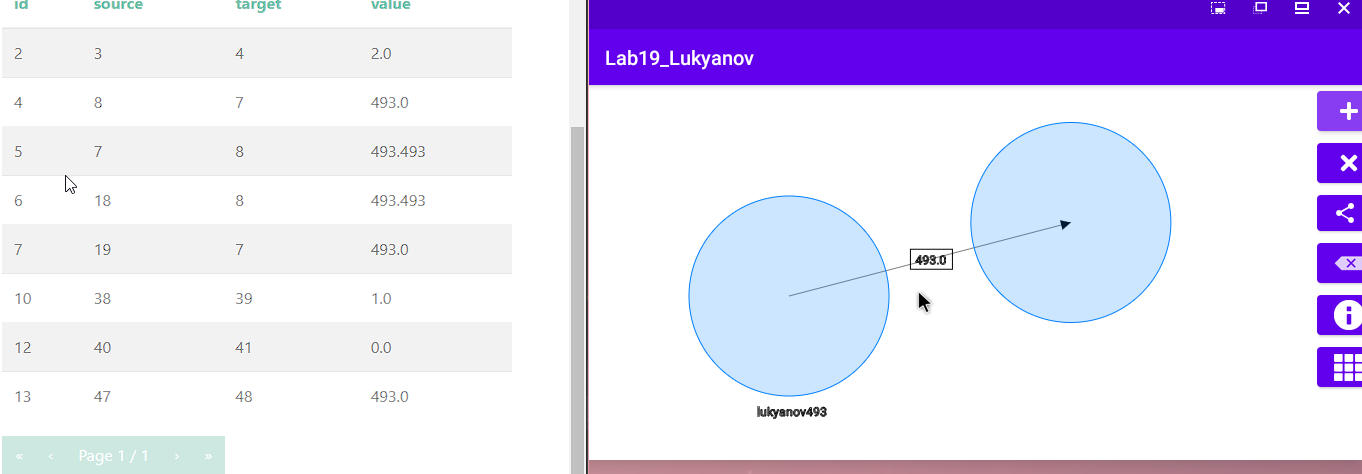


Рисунок 59 – После изменения связи

Удаление узла (рис. 60 и 61):

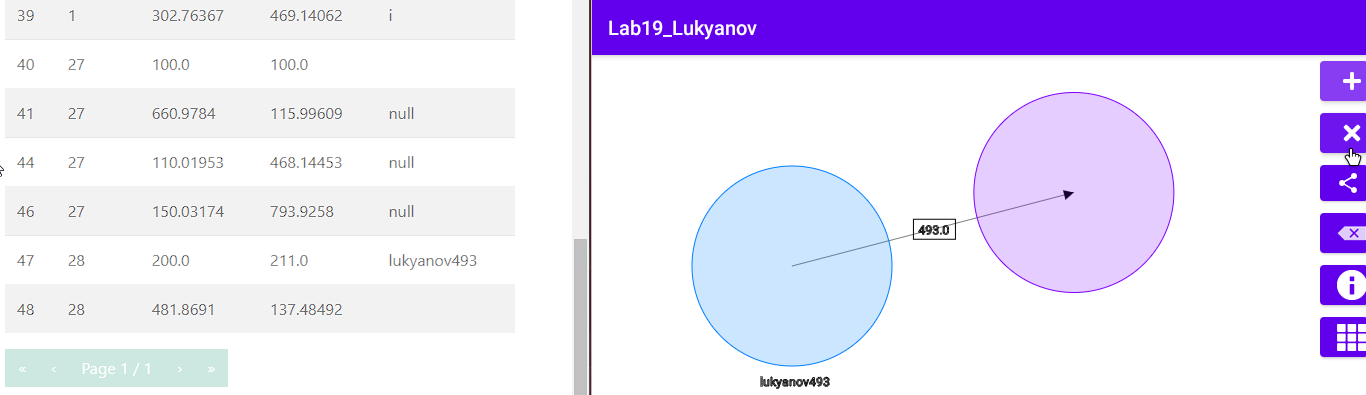


Рисунок 60 – До удаления узла

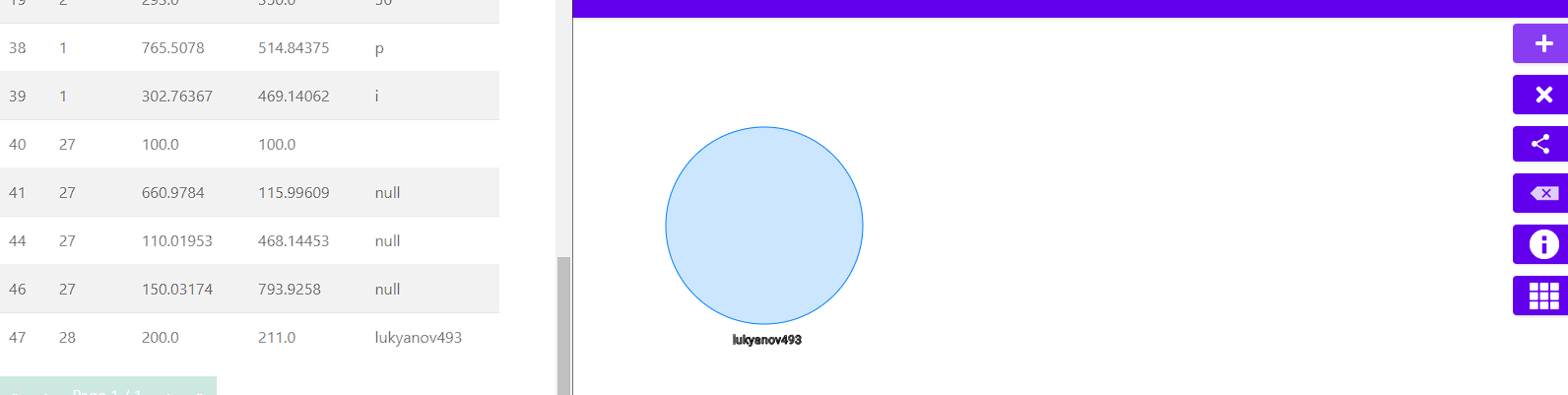


Рисунок 61 – После удаления узла

Удаление связи (рис. 62 и 63):

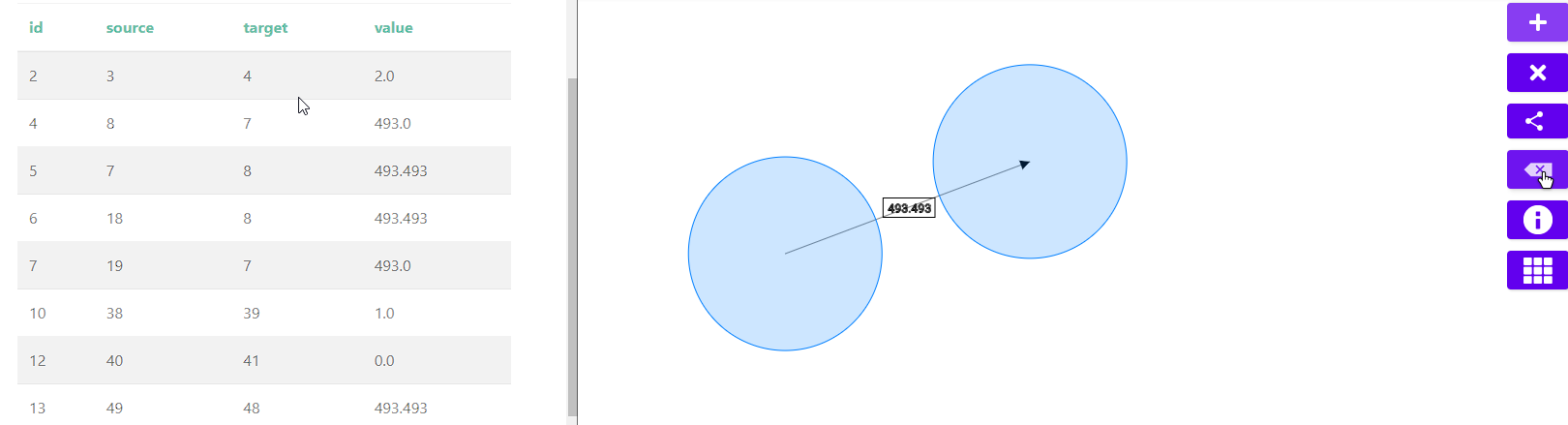


Рисунок 62 – До удаления связи

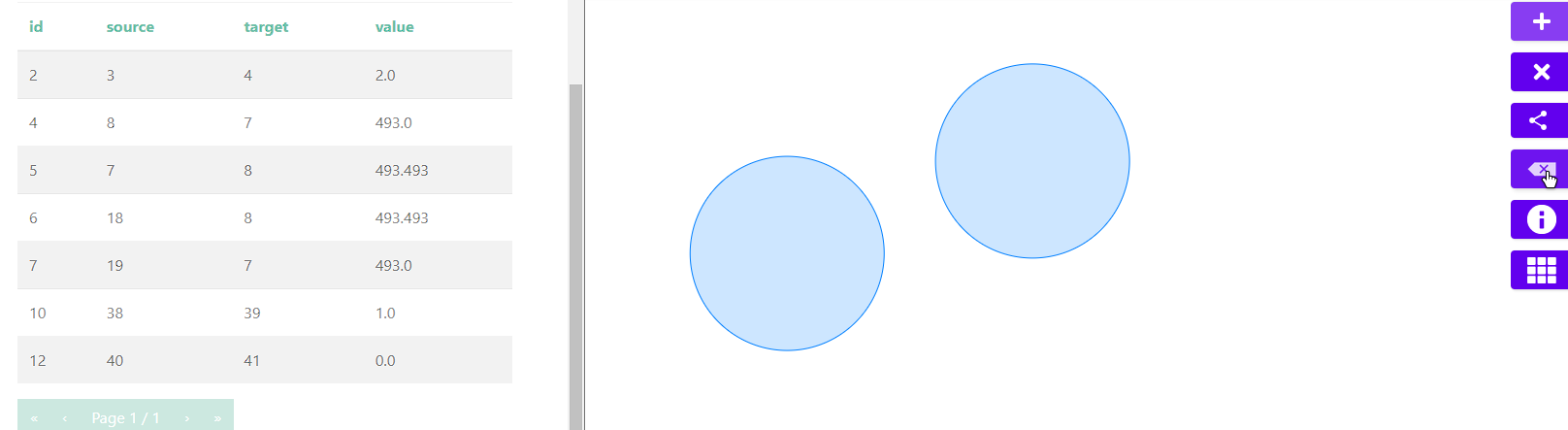


Рисунок 63 – После удаления связи

Граф для сохранения (рис. 64):

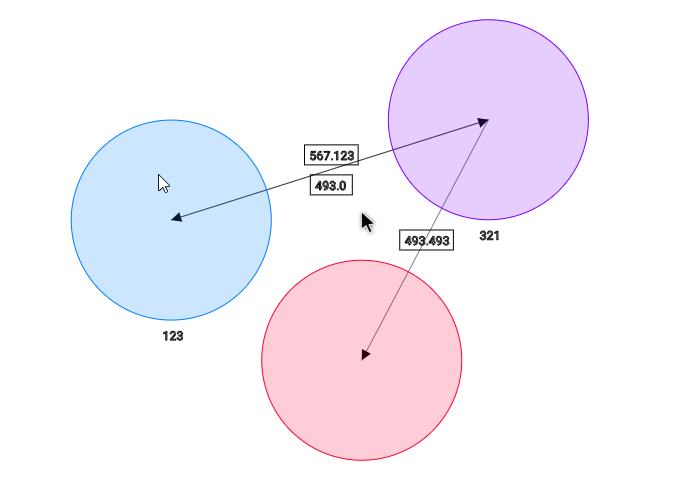


Рисунок 64 – Граф для сохранения

Загрузка графа из API, также можно загрузить из локальной базы данных, но тогда изменения не будут фиксироваться на сервере в реальном времени (рис. 65 и 66):

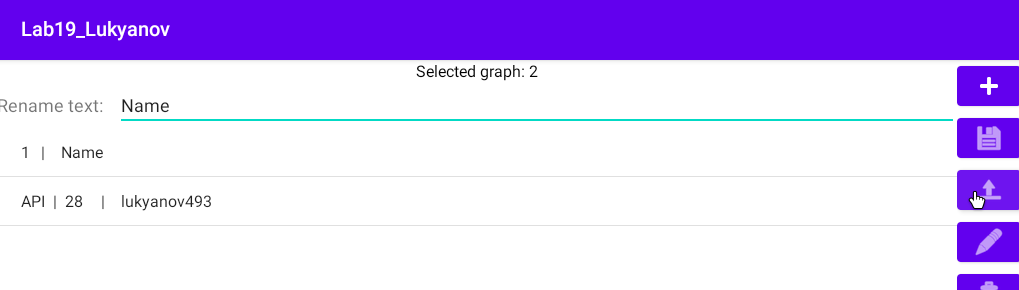


Рисунок 65 – До загрузки графа

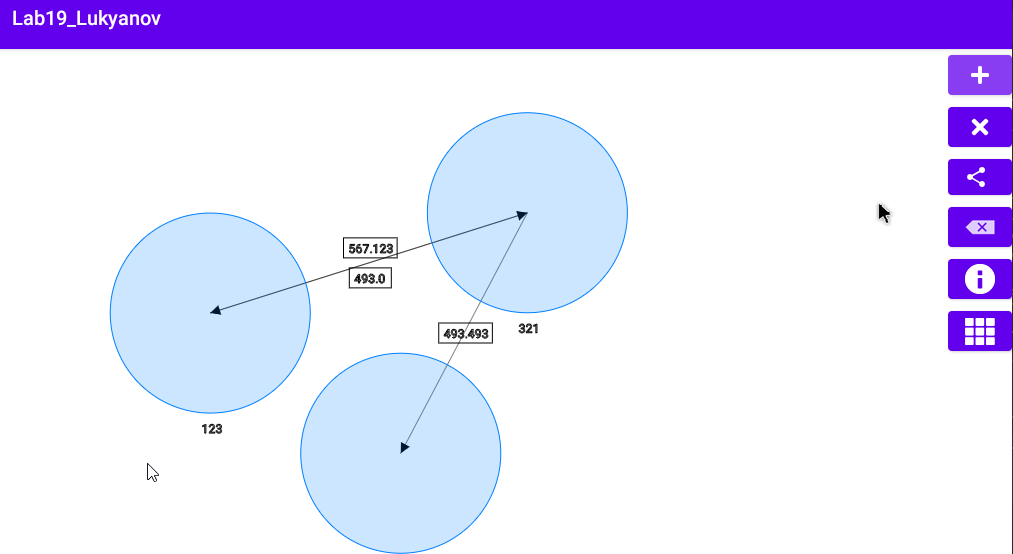


Рисунок 66 – После загрузки графа

Переименование графа (рис. 67 и 68):

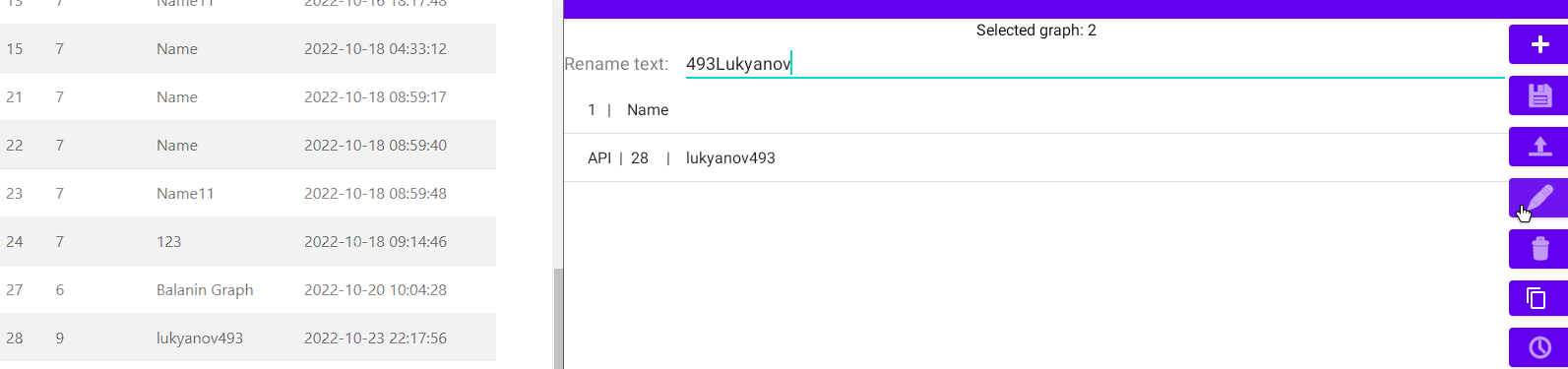


Рисунок 67 – До переименования графа

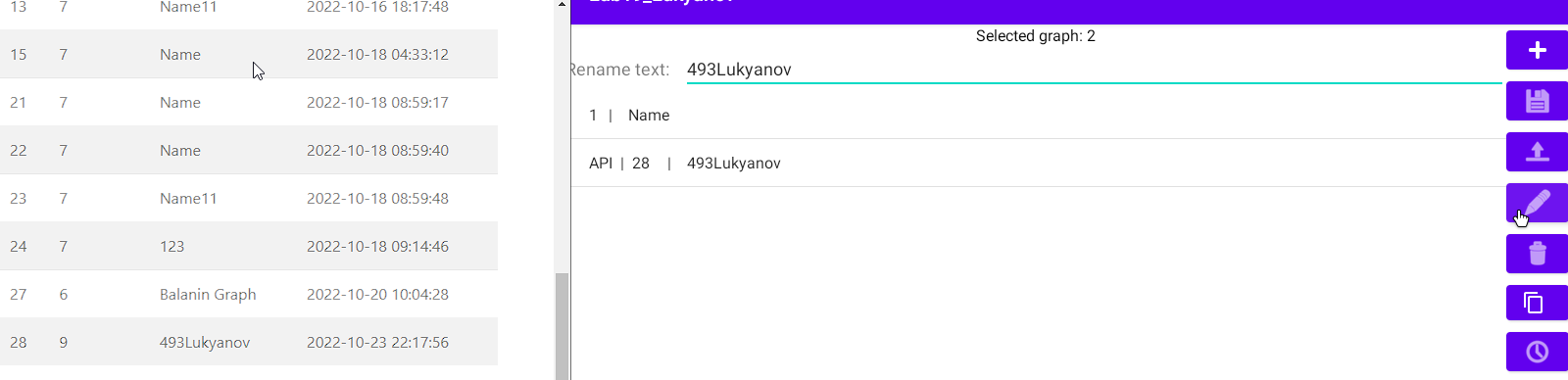


Рисунок 68 – После переименования графа

Удаление графа (рис. 69 и 70):

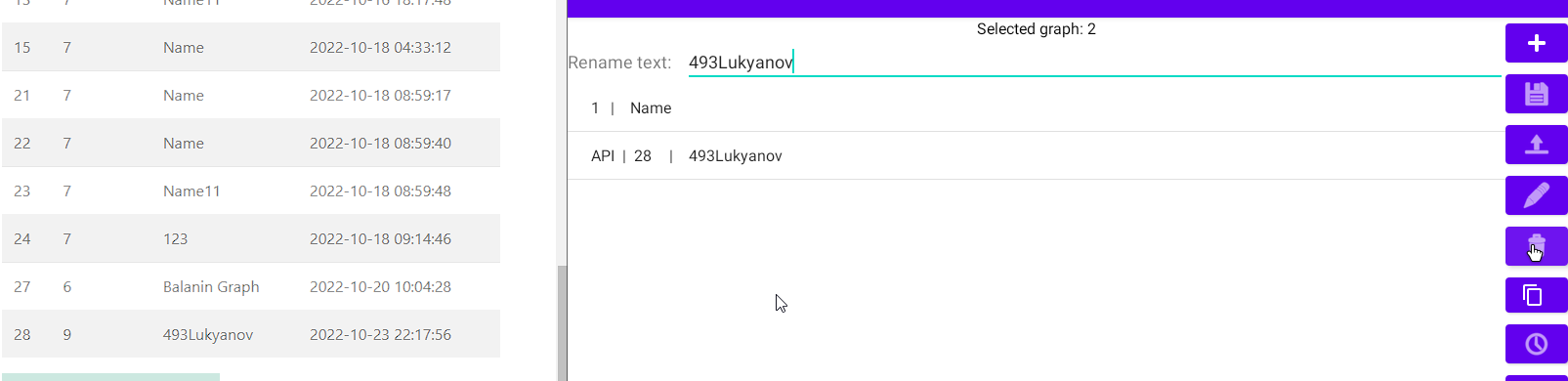


Рисунок 69 – До удаления графа

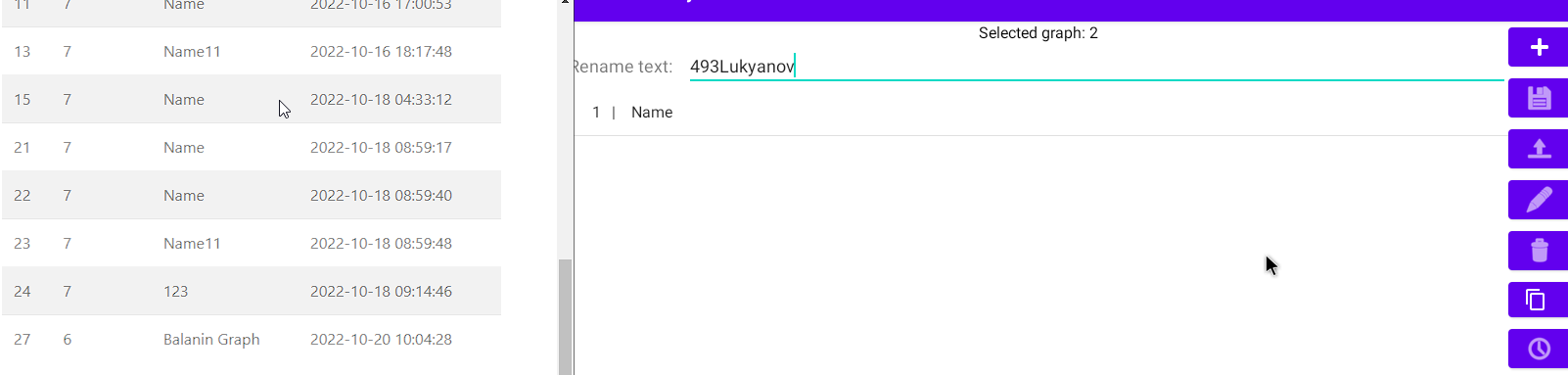


Рисунок 70 – После удаления графа

Изменение пароля (рис. 71 и 72):

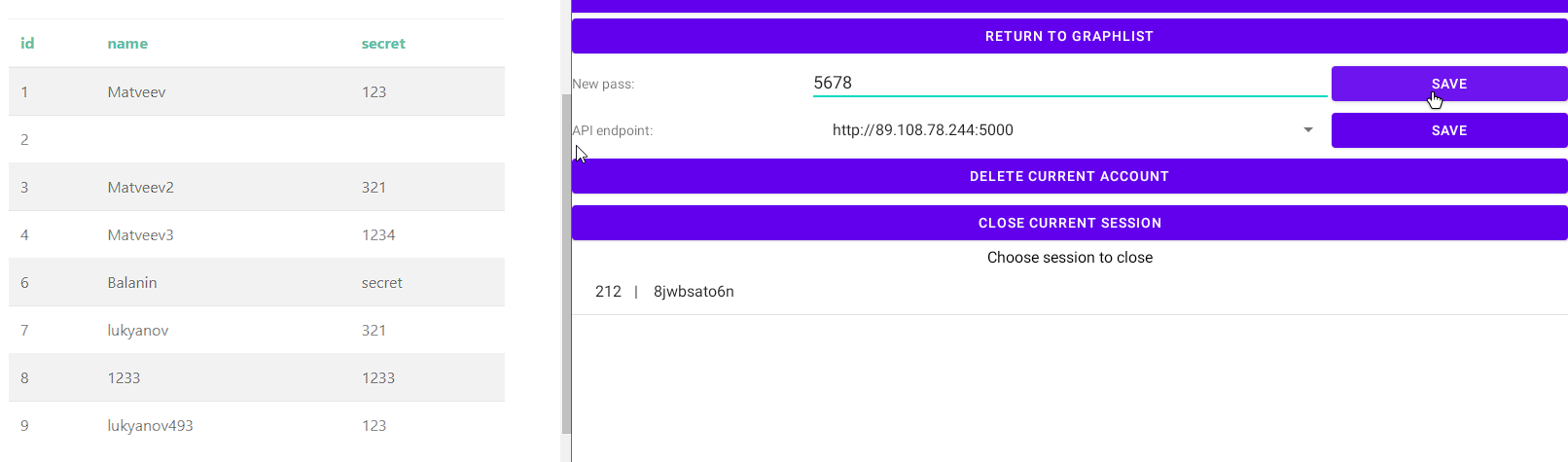


Рисунок 71 – До изменения пароля

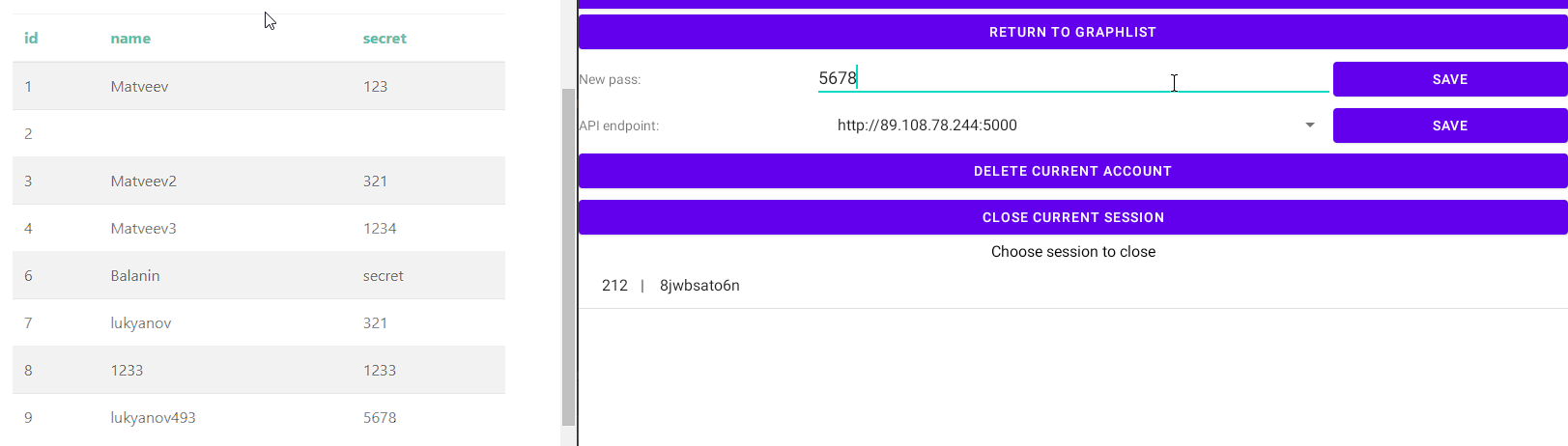


Рисунок 72 – После изменения пароля

Можно выбрать конечную точки API из списка, который был указан в документации на «Swagger» (рис. 73):

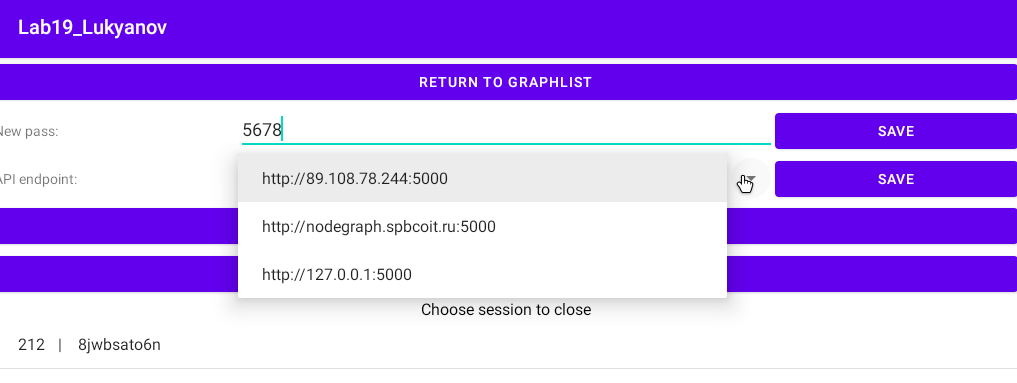


Рисунок 73 – Список с выбором конечной точки API

Отображается список сессий пользователя (рис. 74):

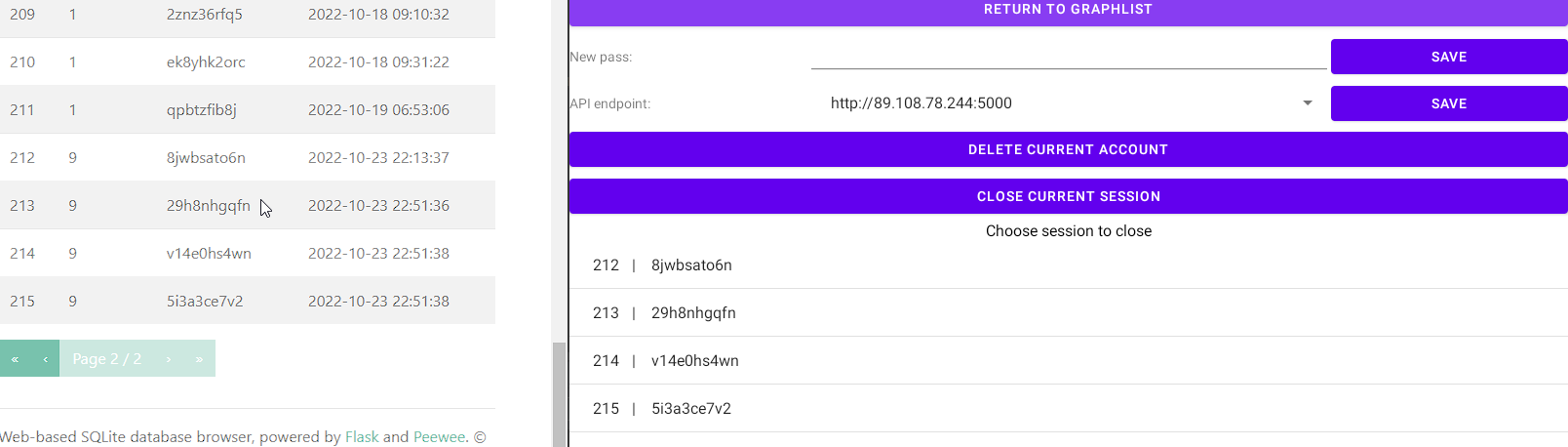


Рисунок 74 – Отображения списка сессий пользователя с сервера

При нажатии на сессию из списка она закрывается (рис. 75 и 76):

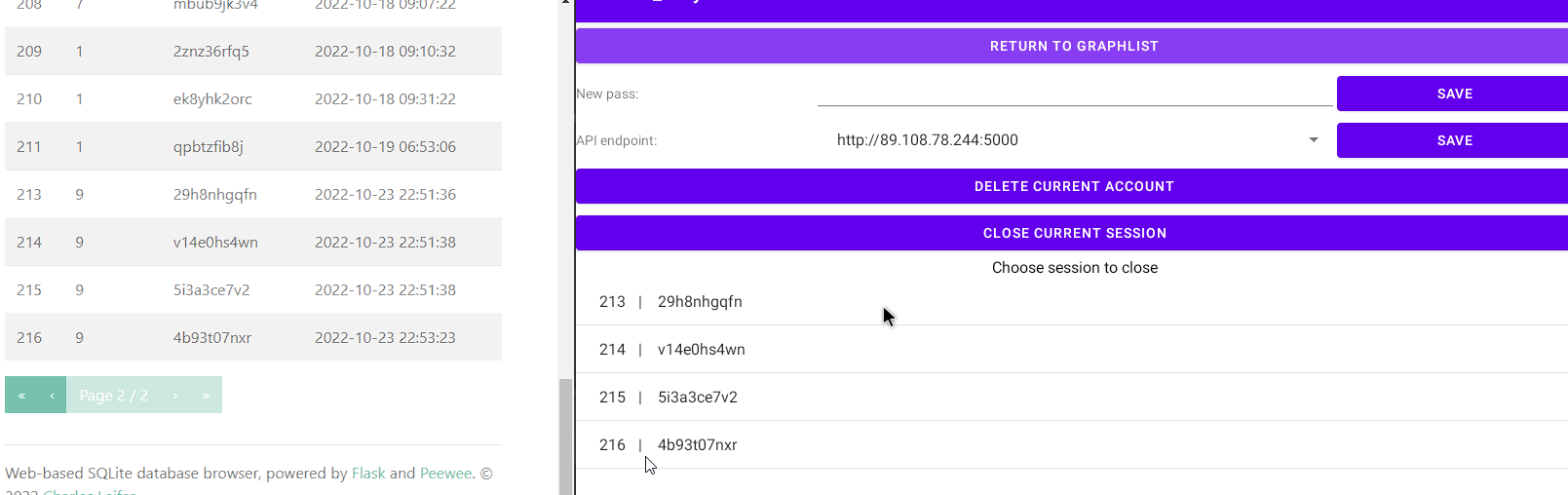


Рисунок 75 – До закрытия сессии

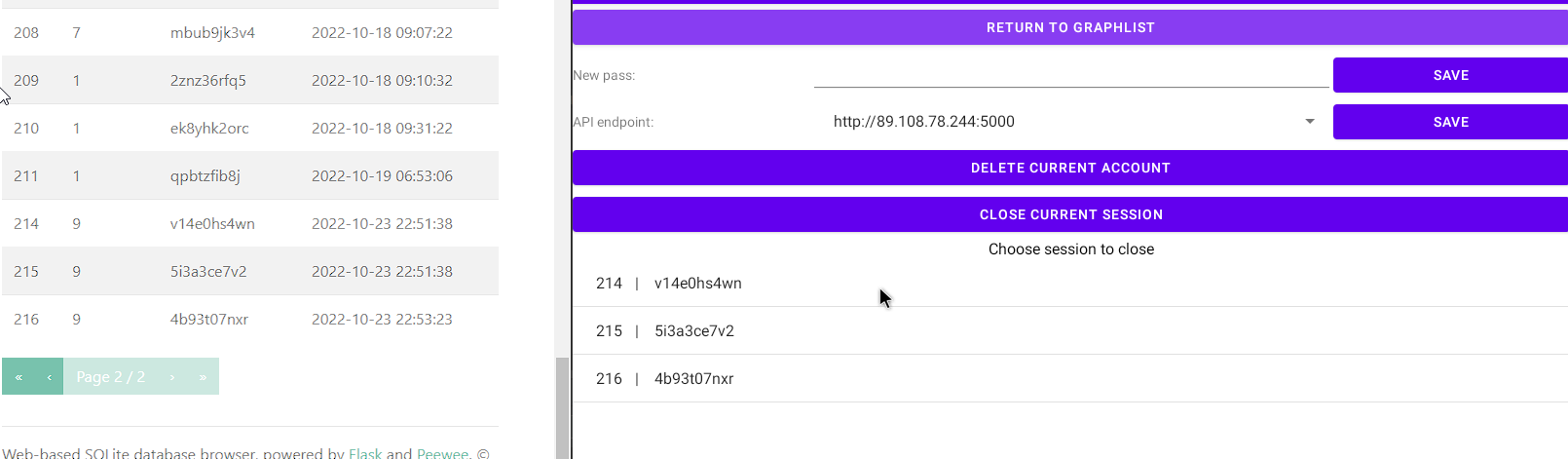


Рисунок 76 – После закрытия сессии

Можно закрыть текущую сессию, тогда произойдёт переход на форму авторизации (рис. 77 и 78):

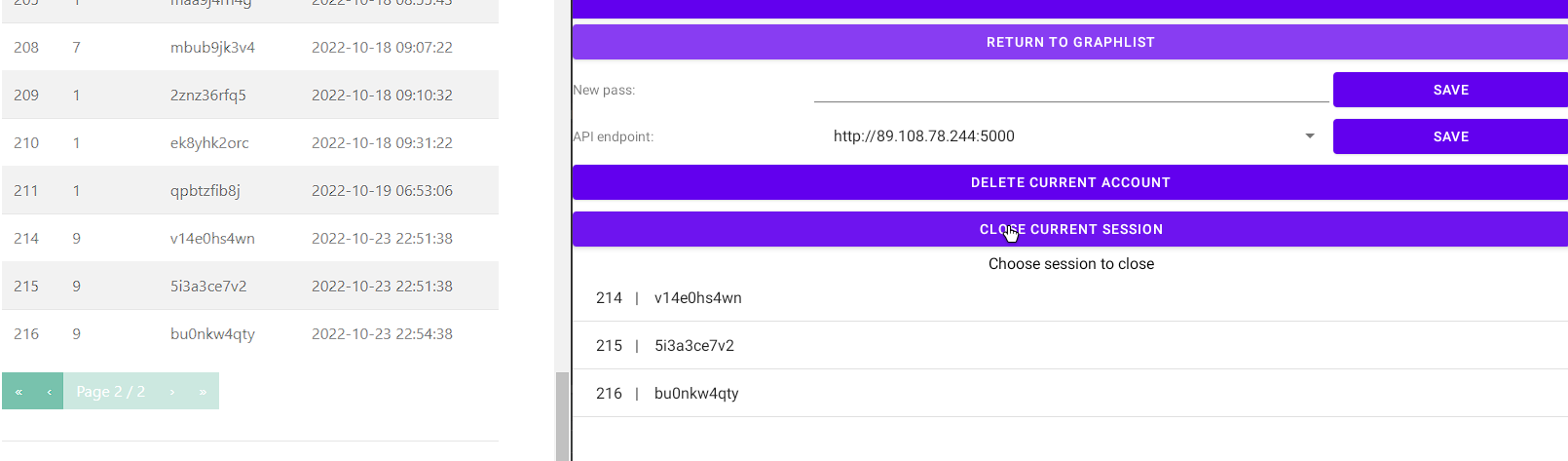


Рисунок 77 – До закрытия текущей сессии

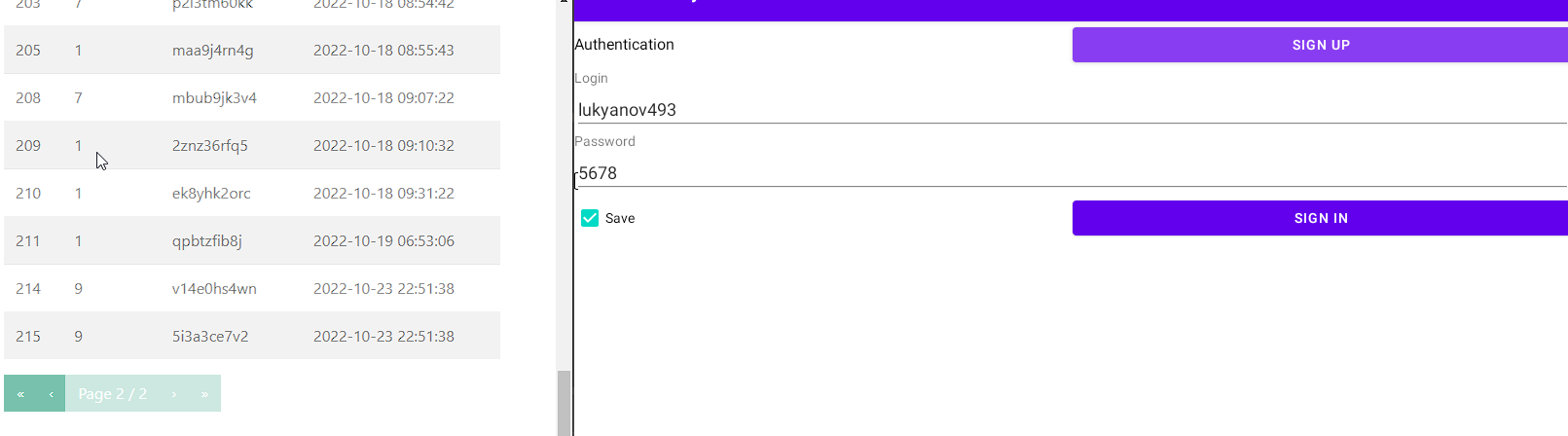


Рисунок 78 – После закрытия текущей сессии

Можно удалить текущий аккаунт (рис. 79 и 80):

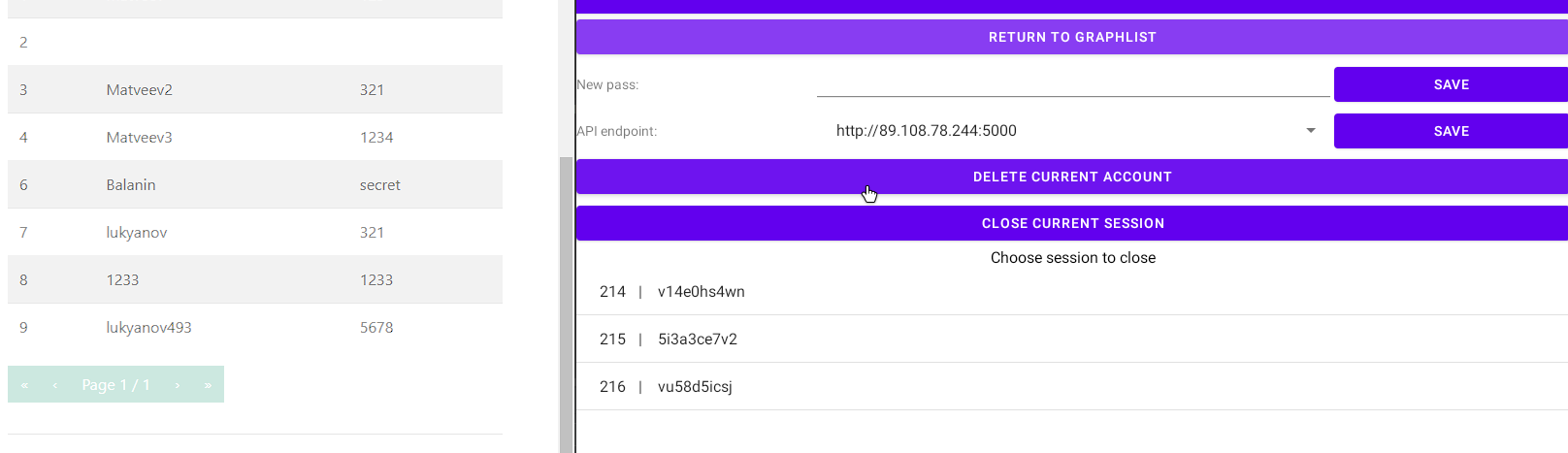


Рисунок 79 – До удаления текущего аккаунта

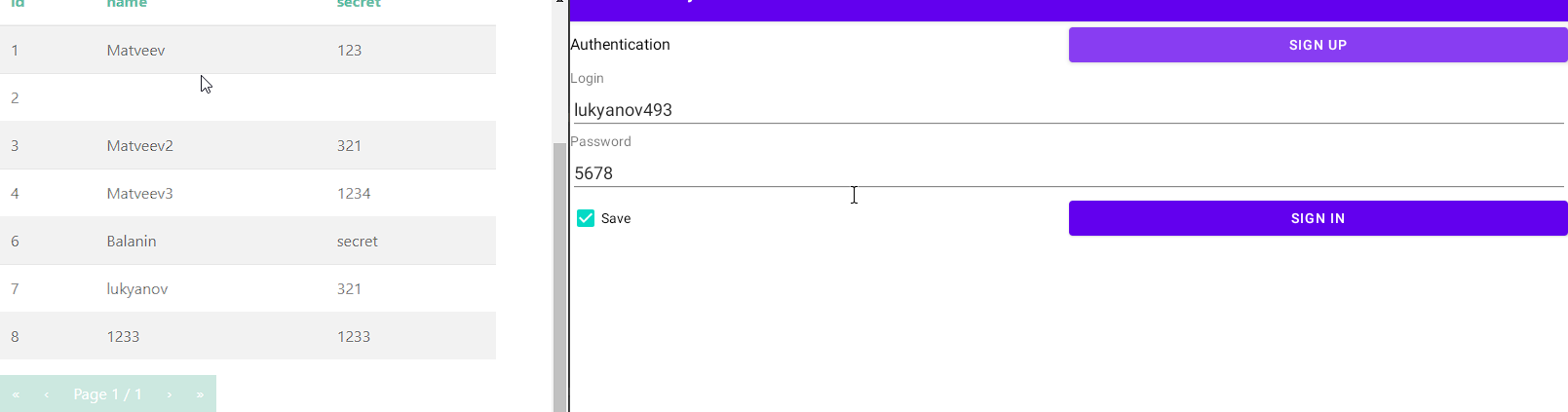


Рисунок 80 – После удаления текущего аккаунта

Приложение имеет собственную иконку (рис. 81):

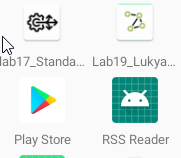


Рисунок 81 – Иконка приложения