

Комитет по образованию Правительства Санкт-Петербурга
**САНКТ-ПЕТЕРБУРГСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Отчет по практической работе
МДК 01.02 «Разработка мобильных приложений»
Разработка интерактивного графического приложения
с 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.

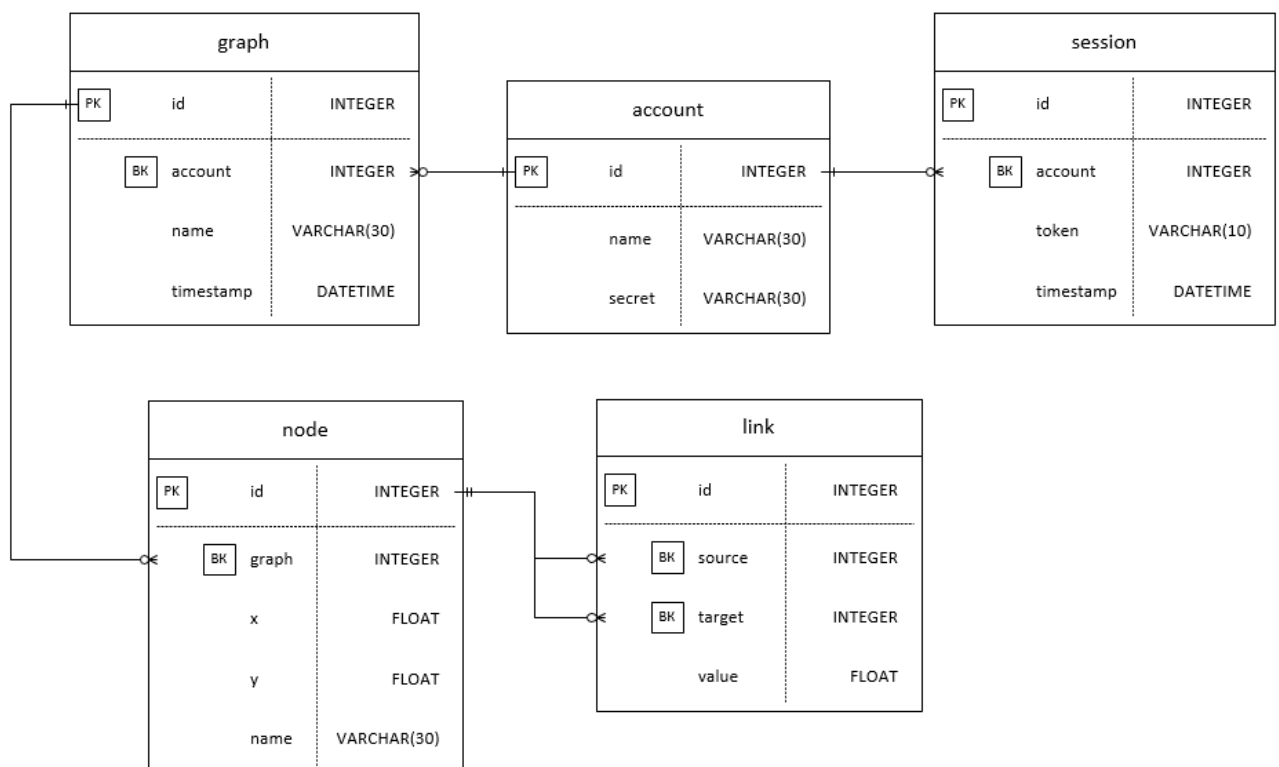


Рисунок 1 – ER диаграмма базы данных

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.

account		user accounts						
#	name	type	size	default	primary	foreign	unique	description
1	id	integer	-		yes	-	yes	unique identifier
2	name	varchar	30		no	-	yes	user login
3	secret	varchar	30		no	-	no	user password

Рисунок 2 – Описание столбцов таблицы account

Пример данных представлен на рисунке 3.

id	name	secret
1	Matveev	123
2		
3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233

Рисунок 3 – 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.

<i>graph</i>		<i>user graphs</i>						
#	name	type	size	default	primary	foreign	unique	description
1	id	integer	-		yes	-	yes	unique identifier
2	account	integer	-		no	account.id	no	associated account
3	name	varchar	30		no	-	no	graph name
4	timestamp	datetime	-	current_timestamp	no	-	no	creation date and time

Рисунок 4 – Описание столбцов таблицы graph

Пример данных представлен на рисунке 5.

id	account	name	timestamp
1	1	uu	2022-10-14 15:03:36
2	7	123	2022-10-15 19:53:05
3	1	P	2022-10-16 08:45:58
4	7	name	2022-10-16 11:55:55

Рисунок 5 – 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.

<i>node</i>		<i>graph nodes</i>						
#	name	type	size	default	primary	foreign	unique	description
1	id	integer	-		yes	-	yes	unique identifier
2	graph	integer	-		no	graph.id	no	associated graph
3	x	float	-		no	-	no	x coordinate value
4	y	float	-		no	-	no	y coordinate value
5	name	varchar	30		no	-	no	node name

Рисунок 6 – Описание столбцов таблицы node

Пример данных представлен на рисунке 7.

id	graph	x	y	name
3	3	328.05664	886.9531	
4	3	690.4297	179.14062	X
7	2	481.0	533.0	

Рисунок 7 – 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.

link		node links						
#	name	type	size	default	primary	foreign	unique	description
1	id	integer	-		yes	-	yes	unique identifier
2	source	integer	-		no	node.id	yes(target)	node from id
3	target	integer	-		no	node.id	yes(source)	node to id
4	value	float	-		no	-	no	link value

Рисунок 8 – Описание столбцов таблицы link

Пример данных представлен на рисунке 9.

id	source	target	value
2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493

Рисунок 9 – 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.

<i>session</i>		<i>account sessions</i>						
#	name	type	size	default	primary	foreign	unique	description
1	id	integer	-		yes	-	yes	unique identifier
2	account	integer	-		no	account.id	no	associated account
3	token	varchar	10		no	-	yes	unique session token
4	timestamp	datetime	-	current_timestamp	no	-	no	creation date and time

Рисунок 10 – Описание столбцов таблицы session

Пример данных представлен на рисунке 11.

id	account	token	timestamp
4	2	y83zqjc68s	2022-10-14 10:41:03
5	3	ojhxb0pfe9	2022-10-14 10:41:26
6	3	lh dh14agvb	2022-10-14 10:41:35
7	2	ay8bn4wti2	2022-10-14 10:45:53
8	2	s08nnchuob	2022-10-14 10:46:21
12	4	vusp6fsy90	2022-10-14 12:45:23

Рисунок 11 – 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.

The mockup shows a rectangular form with a light gray border. At the top, there is a header area with the text "lab19_Lukyanov". Below this, the form is divided into two main sections. The first section is titled "Authentication" and contains a blue button labeled "SIGN UP". Below the "Authentication" title, the word "Login" is displayed. Under "Login", there is a large, empty, light gray rectangular input field. The second section is titled "Password" and contains another large, empty, light gray rectangular input field. At the bottom of the form, there is a checkbox labeled "Save" and a blue button labeled "SIGN IN".

Рисунок 12 – Макет формы Auth

На рисунке 13 показан внешний вид формы авторизации в приложении.

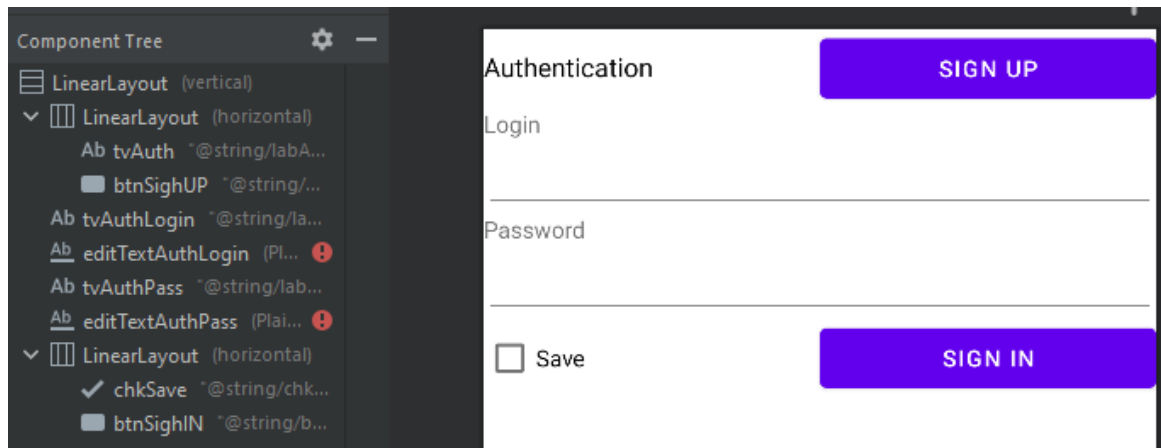


Рисунок 13 – Форма Auth в приложении

Форма Reg Form

На рисунке 14 показан макет внешнего вида формы регистрации.

The image is a mockup of a registration form. It is enclosed in a rectangular frame. At the top, there is a header bar with the text 'lab19_Lukyanov'. Below the header, the form is divided into sections. The first section is titled 'Registration' and contains a blue button labeled 'SIGN IN'. The second section is titled 'Login' and contains a light gray input field. The third section is titled 'Password' and contains another light gray input field. At the bottom of the form, there is a large blue button labeled 'SIGN UP'.

Рисунок 14 – Макет формы Reg

На рисунке 15 показан внешний вид формы регистрации в приложении.

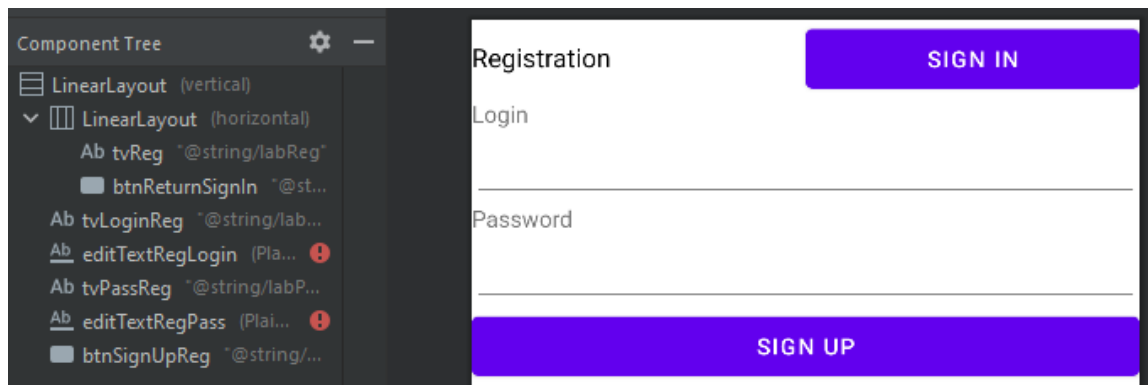


Рисунок 15 – Форма Reg в приложении

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.

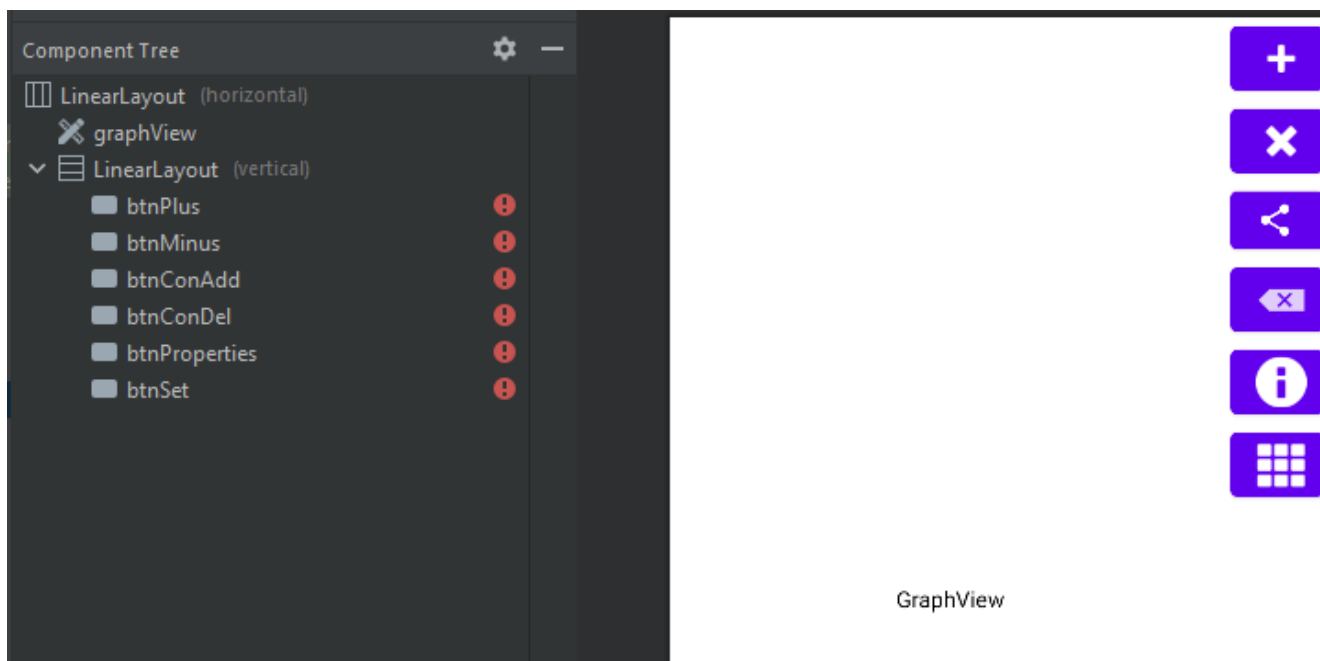


Рисунок 16 – Форма Main Menu в приложении

Форма Graph

На рисунке 19 показан макет внешнего вида формы графов.

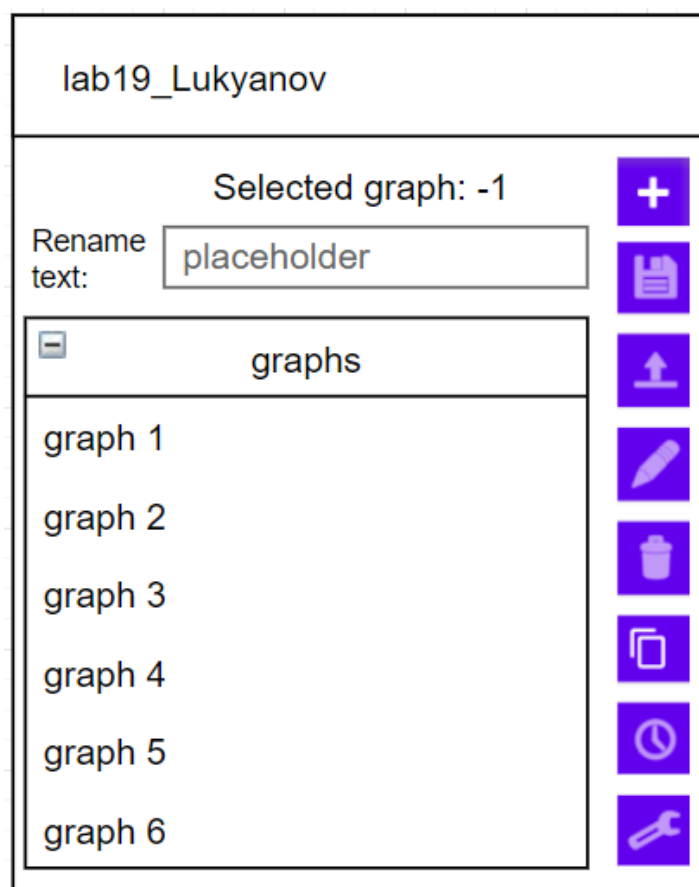


Рисунок 19 – Макет формы Graph

На рисунке 20 показан внешний вид формы графов в приложении.

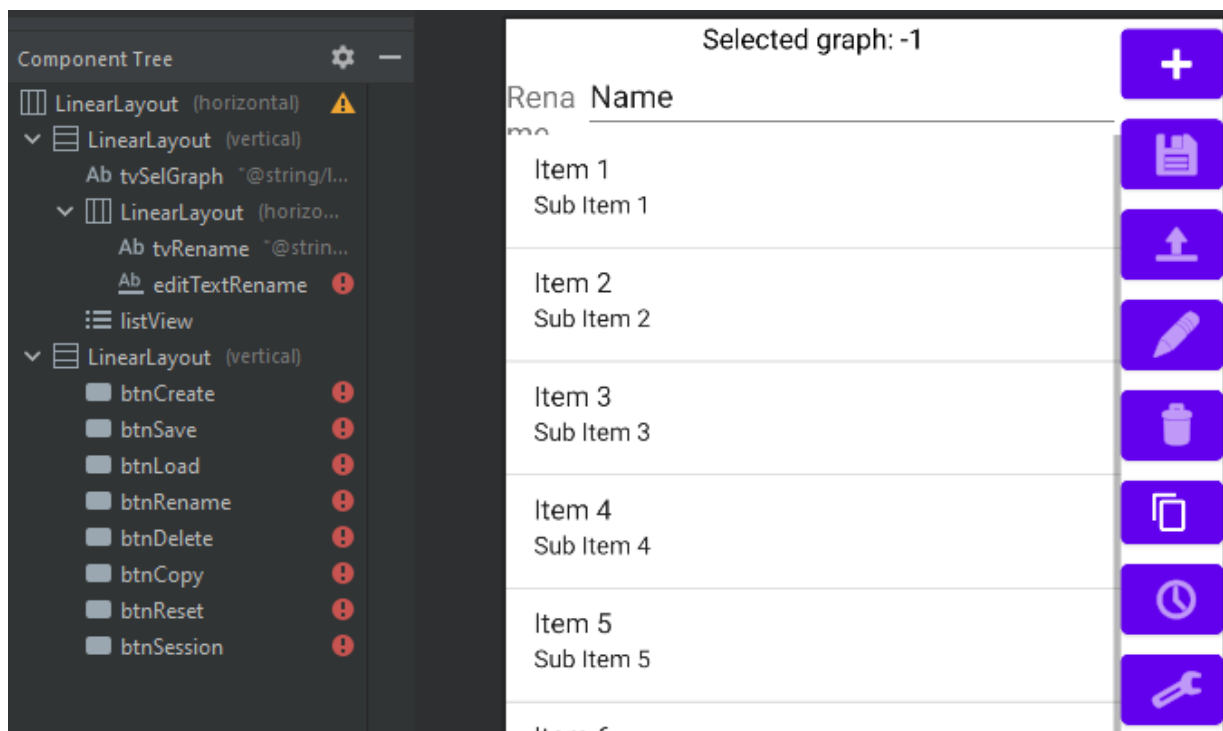


Рисунок 20 – Форма Graph в приложении

Форма Session

На рисунке 21 показан макет внешнего вида формы сессий.

The mockup shows a mobile application interface for session management. At the top, the text 'lab19_Lukyanov' is displayed. Below it is a blue button labeled 'RETURN TO GRAPHLIST'. There are two input sections: 'New pass:' with a text field containing 'new password' and a blue 'SAVE' button, and 'API endpoint:' with a dropdown menu showing 'endpoints' and a blue 'SAVE' button. Below these are two more blue buttons: 'DELETE CURRENT ACCOUNT' and 'CLOSE CURRENT SESSION'. At the bottom, there is a 'Session list' section with a minus icon on the left. The list contains two entries: 'session 1' and 'session 2'.

Рисунок 21 – Макет формы Session

На рисунке 22 показан внешний вид формы сессий в приложении.

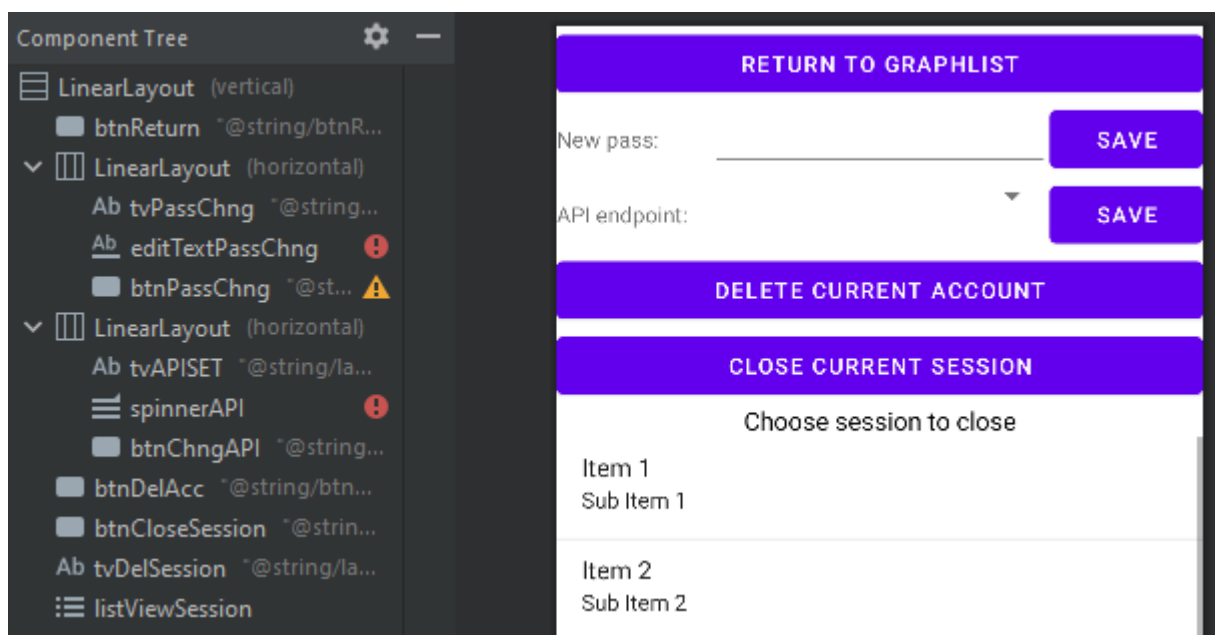


Рисунок 22 – Форма Session в приложении

Форма Node

На рисунке 23 показан макет внешнего вида формы узла.

lab18_Lukyanov	
Node text:	<input type="text" value="placeholder"/>
Node X:	<input type="text" value="placeholder"/>
Node Y:	<input type="text" value="placeholder"/>
<div>SAVE NODE</div>	

Рисунок 23 – Макет формы Node

На рисунке 24 показан внешний вид формы сообщения в приложении.

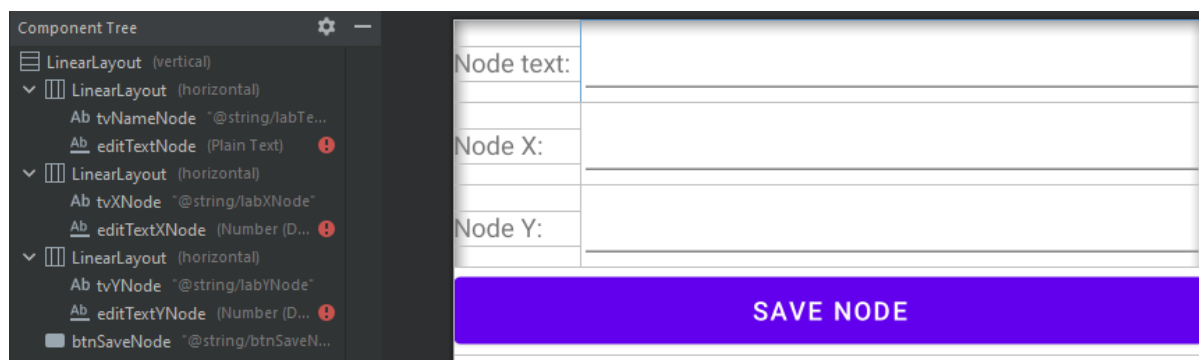
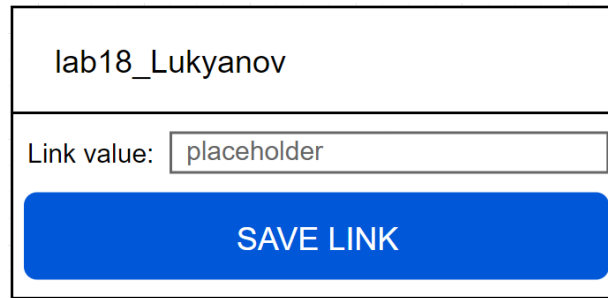


Рисунок 24 – Форма Node в приложении

Форма Link

На рисунке 25 показан макет внешнего вида формы связи.



The mockup shows a form with a white background and a black border. At the top, there is a text field containing the text "lab18_Lukyanov". Below this, there is a label "Link value:" followed by a text input field containing the placeholder text "placeholder". At the bottom of the form is a large blue button with the text "SAVE LINK" in white capital letters.

Рисунок 25 – Макет формы Link

На рисунке 26 показан внешний вид формы связи в приложении.

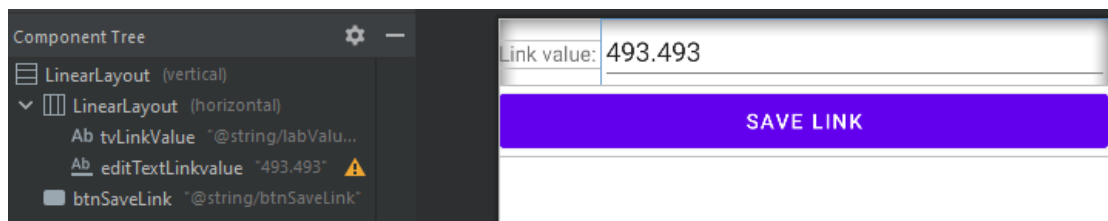


Рисунок 26 – Форма Link в приложении

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.

account account management ^	
PUT	/account/create create new account v
DELETE	/account/delete delete currently logged in account v
POST	/account/update change currently logged in account password v
session session control ^	
DELETE	/session/close terminate existing login session v
GET	/session/list show all active login sessions for currently logged in account v
PUT	/session/open create new login session v
graph graph management functions ^	
PUT	/graph/create create new empty graph v
DELETE	/graph/delete delete existing graph v
GET	/graph/list retrieve a list of existing graphs v
POST	/graph/update rename existing graph v
node node management functions ^	
PUT	/node/create add new node v
DELETE	/node/delete delete existing node v
GET	/node/list show all existing nodes on given graph v
POST	/node/update update existing node parameters v
link link management functions ^	
PUT	/link/create create a link between two nodes v
DELETE	/link/delete delete existing link between two nodes v
GET	/link/list retrieve a list of all links connecting nodes v
POST	/link/update change existing link parameter value v

Рисунок 27 – Список функций

1. Управление учетными записями

Данный набор функций позволяет управлять учетными записями пользователей.

1.1. 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 строка)
- secret – пароль учетной записи (текстовая строка)

Выходные параметры:

— код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 28.

The screenshot shows a web application interface for creating a new account. The top bar indicates a PUT request to /account/create. The main area is titled 'create new account' and contains a 'Parameters' section with two input fields: 'name' (required, string, query) with the value 'lukyanov493' and 'secret' (required, string, query) with the value '123'. Below the inputs are 'Execute' and 'Clear' buttons. The 'Responses' section shows the curl command, the request URL, and the server response. The server response is a 200 status code with headers: connection: close, content-length: 0, content-type: application/json, date: Wed, 19 Oct 2022 21:40:25 GMT, and server: Werkzeug/2.2.2 Python/3.9.2.

Name	Description
name * required string (query)	new unique user name
secret * required string (query)	account password

Execute Clear

Responses

Curl

```
curl -X 'PUT' \
'http://89.108.78.244:5000/account/create?name=lukyanov493&secret=123' \
-H 'accept: */*'
```

Request URL

```
http://89.108.78.244:5000/account/create?name=lukyanov493&secret=123
```

Server response

Code	Details
200	Response headers connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 21:40:25 GMT server: Werkzeug/2.2.2 Python/3.9.2

Рисунок 28 – Вызов функции /account/create

1.2. 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.

DELETE /account/delete delete currently logged in account

delete currently logged in account

Parameters Cancel

Name	Description
token * required	existing session token
string (query)	<input type="text" value="jqgaf0vm3i"/>

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://89.108.78.244:5000/account/delete?token=jqgaf0vm3i' \
-H 'accept: */*'
```

Request URL

```
http://89.108.78.244:5000/account/delete?token=jqgaf0vm3i
```

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 21:44:18 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

Рисунок 29 – Вызов функции /account/delete

1.3. 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 строка).

Выходные параметры:

- код успешного/неуспешного выполнения функции.

Пример вызова функции показан на рисунке 30.

POST /account/update change currently logged in account password

change currently logged in account password

Parameters Cancel

Name	Description
token * required string (query)	existing session token
<input type="text" value="h1llw3v159"/>	
secret * required string (query)	new account password
<input type="text" value="321"/>	

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://89.108.78.244:5000/account/update?token=h1llw3v159&secret=321' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://89.108.78.244:5000/account/update?token=h1llw3v159&secret=321
```

Server response

Code	Details
200	Response headers

Рисунок 30 – Вызов функции /account/update

2. Управление сессиями пользователей

Данный набор функций позволяет управлять сессиями пользователей.

2.1. 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.

DELETE /session/close terminate existing login session

terminate existing login session

Parameters

token * required

string

(query)

existing session token

yo7m11dlib

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://89.108.78.244:5000/session/close?token=yo7m11dlib' \
-H 'accept: */*'
```

Request URL

http://89.108.78.244:5000/session/close?token=yo7m11dlib

Server response

Code

Details

200

Response headers

Рисунок 31 – Вызов функции /session/close

2.2. 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.

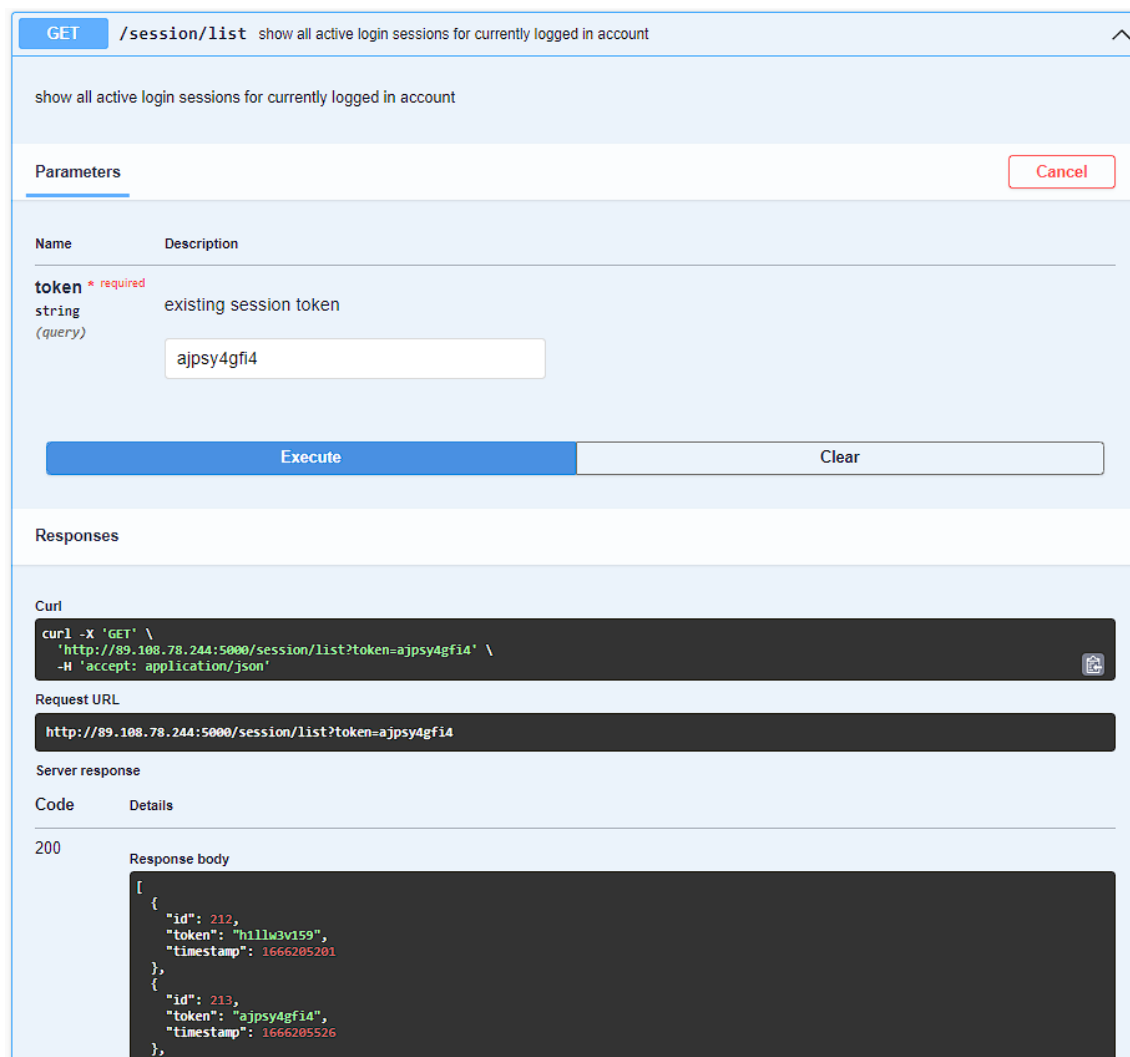


Рисунок 32 – Вызов функции /session/list

2.3. 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 строка).
- secret – пароль учетной записи (текстовая строка).

Выходные параметры:

- token: уникальный ключ сессии для данного пользователя.

Пример вызова функции показан на рисунке 33.

PUT /session/open create new login session

create new login session

Parameters Cancel

Name	Description
name * required string (query)	existing user name
secret * required string (query)	account password

Execute Clear

Responses

Curl

```
curl -X 'PUT' \
'http://89.108.78.244:5000/session/open?name=lukyanov493&secret=321' \
-H 'accept: application/json'
```

Request URL

```
http://89.108.78.244:5000/session/open?name=lukyanov493&secret=321
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "token": "3kynzjnxSr" }</pre> <p>Download</p>

Рисунок 33 – Вызов функции /session/open

3. Управление графами пользователей

Данный набор функций позволяет управлять графами пользователей.

3.1. 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.

PUT /graph/create create new empty graph

create new empty graph

Parameters Cancel

Name	Description
token * required string (query)	active session token
name * required string (query)	graph name

3kynzjnx5r

lukyanov493

Execute Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://89.108.78.244:5000/graph/create?token=3kynzjnx5r&name=lukyanov493' \
  -H 'accept: application/json'
```

Request URL

```
http://89.108.78.244:5000/graph/create?token=3kynzjnx5r&name=lukyanov493
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 26 }</pre> <p>Download</p>

Рисунок 34 – Вызов функции /graph/create

3.2. 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.

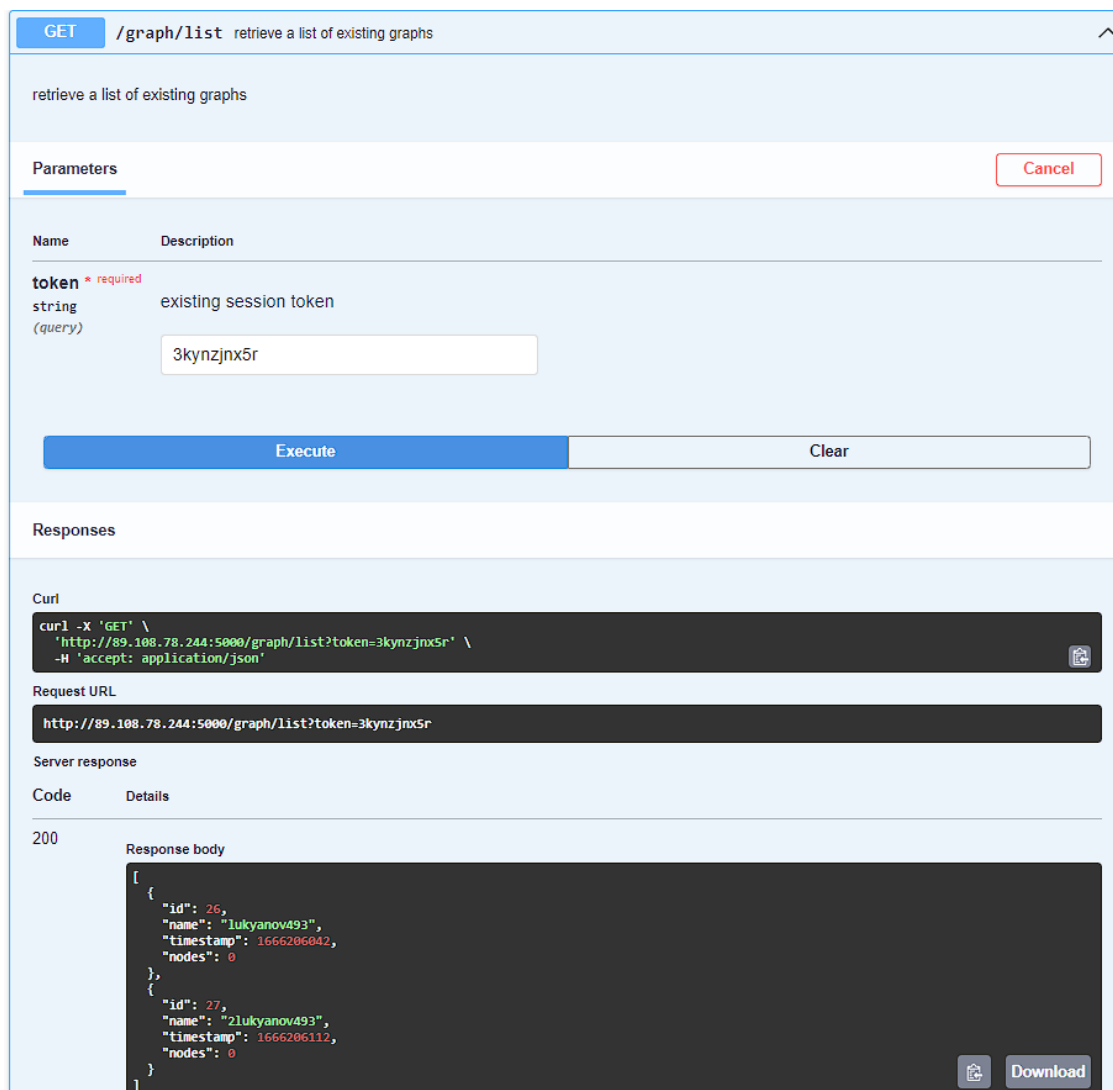


Рисунок 35 – Вызов функции /graph/list

3.3. 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.

DELETE /graph/delete delete existing graph

delete existing graph

Parameters Cancel

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing graph id

3kynzjnx5r

27

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://89.108.78.244:5000/graph/delete?token=3kynzjnx5r&id=27' \
  -H 'accept: */*'
```

Request URL

```
http://89.108.78.244:5000/graph/delete?token=3kynzjnx5r&id=27
```

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 22:04:46 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

Рисунок 36 – Вызов функции /graph/delete

3.4. 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.

POST /graph/update rename existing graph

rename existing graph

Parameters Cancel

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing graph id
name * required string (query)	new graph name

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://89.108.78.244:5000/graph/update?token=3kynzjnx5r&id=26&name=493lukyanov' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://89.108.78.244:5000/graph/update?token=3kynzjnx5r&id=26&name=493lukyanov
```

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 22:07:41 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

Рисунок 37 – Вызов функции /graph/update

4. Управление узлами графов

Данный набор функций позволяет управлять узлами графов пользователей.

4.1. 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.

The screenshot shows a web interface for calling the `/node/create` function. The interface is divided into two main sections: **Parameters** and **Responses**.

Parameters Section:

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing graph id
x * required number (query)	x coordinate
y * required number (query)	y coordinate
name * required string (query)	node name

Below the parameters table, there are input fields for each parameter:

- token: 3kynzjnx5r
- id: 26
- x: 15
- y: 25
- name: 493

At the bottom of the parameters section, there are two buttons: **Execute** and **Clear**.

Responses Section:

Curl

```
curl -X 'PUT' \
'http://89.188.78.244:5000/node/create?token=3kynzjnx5r&id=26&x=15&y=25&name=493' \
-H 'accept: application/json'
```

Request URL

```
http://89.188.78.244:5000/node/create?token=3kynzjnx5r&id=26&x=15&y=25&name=493
```

Server response

Code	Details
200	Response body

The response body is shown in a dark box with a **Download** button:

```
{
  "id": 35
}
```

Рисунок 38 – Вызов функции `/node/create`

4.2. 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.

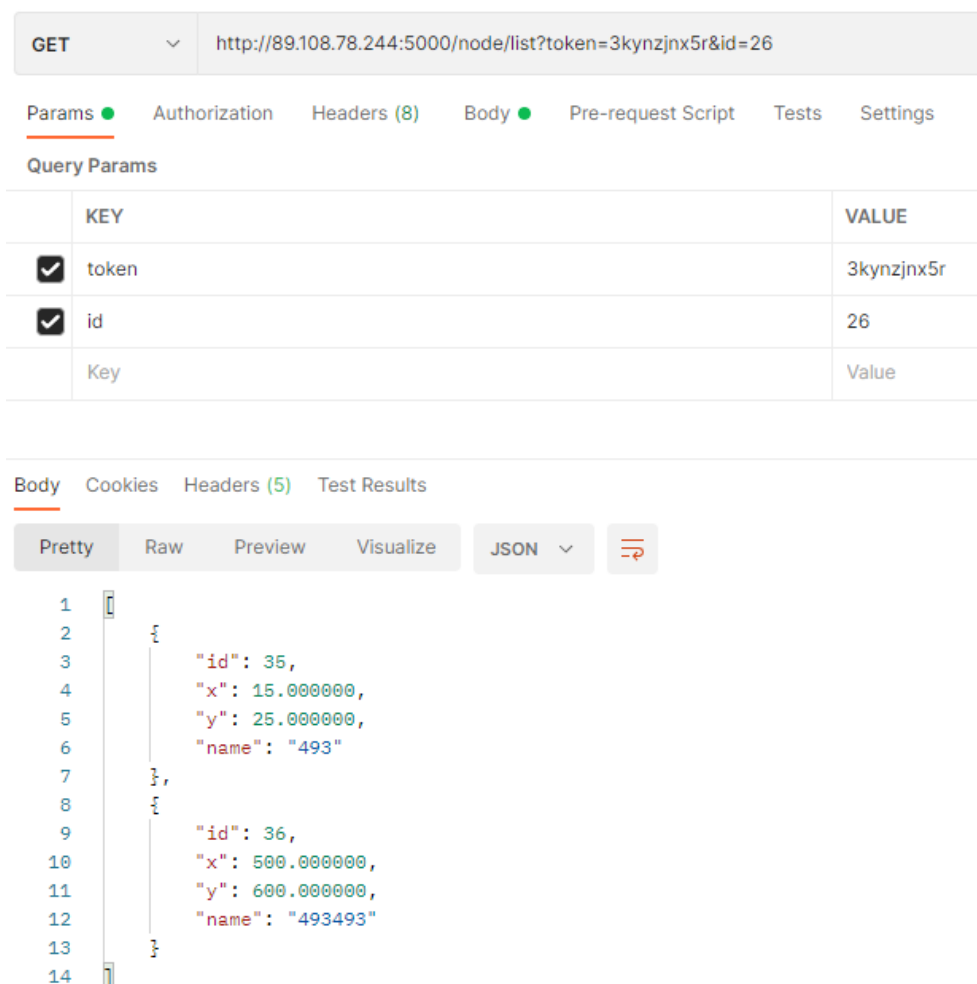


Рисунок 39 – Вызов функции /node/list

4.3. 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.

DELETE /node/delete delete existing node

delete existing node

Parameters Cancel

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing node id

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://89.108.78.244:5000/node/delete?token=3kynzjnx5r&id=36' \
-H 'accept: */*'
```

Request URL

```
http://89.108.78.244:5000/node/delete?token=3kynzjnx5r&id=36
```

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 22:17:50 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

Рисунок 40 – Вызов функции /node/delete

4.4. 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.

POST /node/update update existing node parameters

update existing node parameters

Parameters

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing node id
x * required number (query)	new x coordinate
y * required number (query)	new y coordinate
name * required string (query)	new node name

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://89.188.78.244:5000/node/update?token=3kynzjnx5r&id=35&x=35&y=45&name=lukyanov493' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://89.188.78.244:5000/node/update?token=3kynzjnx5r&id=35&x=35&y=45&name=lukyanov493
```

Server response

Code	Details
200	Response headers connection: close content-length: 0 content-type: application/json

Рисунок 41 – Вызов функции /node/update

5. Управление связями узлов графов

Данный набор функций позволяет управлять связями узлов графов пользователей.

5.1. 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.

PUT /link/create create a link between two nodes

create a link between two nodes

Parameters Cancel

Name	Description
token * required string (query)	active session token
<input type="text" value="3kynzjnx5r"/>	
source * required integer (query)	source node id
<input type="text" value="35"/>	
target * required integer (query)	target node id
<input type="text" value="36"/>	
value * required number (query)	parameter value
<input type="text" value="493"/>	

Execute Clear

Responses

Curl

```
curl -X 'PUT' \
'http://89.188.78.244:5000/link/create?token=3kynzjnx5r&source=35&target=36&value=493' \
-H 'accept: application/json'
```

Request URL

```
http://89.188.78.244:5000/link/create?token=3kynzjnx5r&source=35&target=36&value=493
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 8 }</pre> Download

Рисунок 42 – Вызов функции /link/create

5.2. 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.

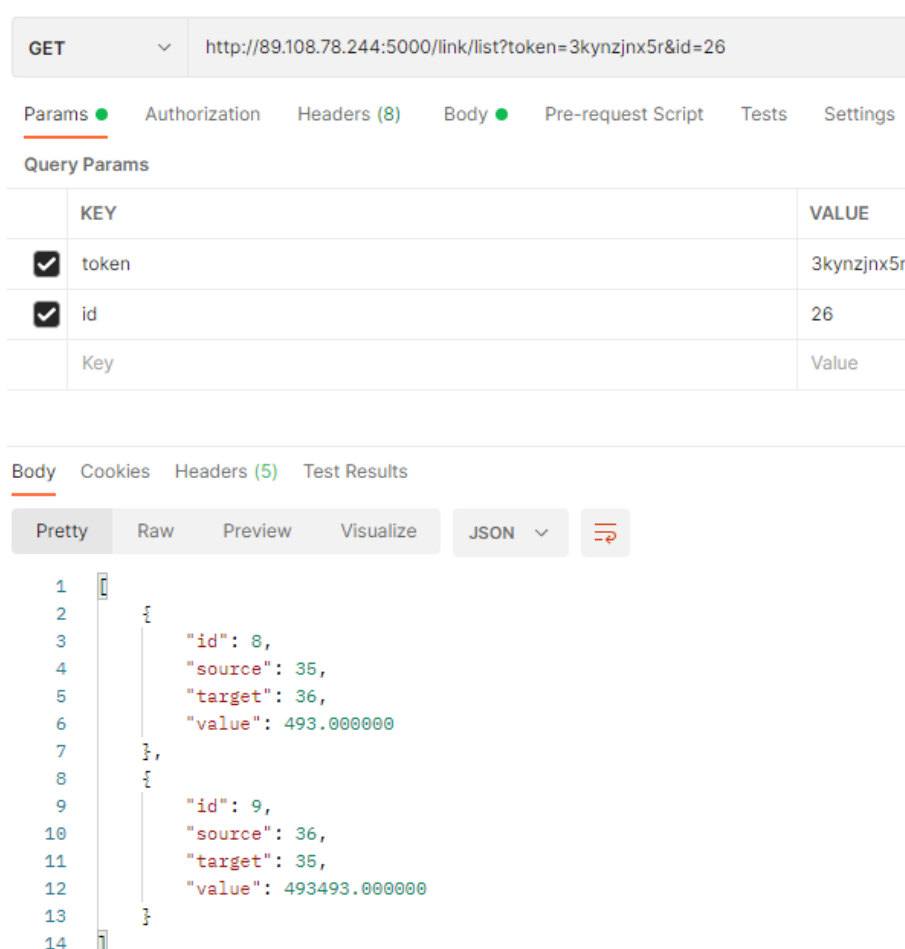


Рисунок 43 – Вызов функции /link/list

5.3. 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.

DELETE /link/delete delete existing link between two nodes

delete existing link between two nodes

Parameters Cancel

Name	Description
token * required string (query)	active session token
<input type="text" value="3kynzjnx5r"/>	
id * required integer (query)	existing link id
<input type="text" value="9"/>	

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \  
'http://89.108.78.244:5000/link/delete?token=3kynzjnx5r&id=9' \  
-H 'accept: */*'
```

Request URL

```
http://89.108.78.244:5000/link/delete?token=3kynzjnx5r&id=9
```

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 22:30:02 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

Рисунок 44 – Вызов функции /link/delete

5.4. 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.

POST /link/update change existing link parameter value

change existing link parameter value

Parameters Cancel

Name	Description
token * required string (query)	active session token
id * required integer (query)	existing link id
value * required number (query)	new parameter value

3kynzjnx5r

8

5678

Execute Clear

Responses

Curly

```
curl -X 'POST' \
  'http://89.108.78.244:5000/link/update?token=3kynzjnx5r&id=8&value=5678' \
  -H 'accept: */*' \
  -d ''
```

Request URL

http://89.108.78.244:5000/link/update?token=3kynzjnx5r&id=8&value=5678

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: close content-length: 0 content-type: application/json date: Wed, 19 Oct 2022 22:32:39 GMT server: Werkzeug/2.2.2 Python/3.9.2</pre>

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:

7	lukyanov	321
8	1233	1233
9	lukyanov493	321

« < Page 1 / 1 > »

Login

lukyanov

Password

321

☒ Save

SIGN IN

Рисунок 46 – До авторизации

Lab19_Lukyanov

Selected graph: -1

Rename text: Name

API | 2 | 123

API | 4 | name

API | 5 | Name

API | 6 | Name

API | 11 | Name

API | 13 | Name11

API | 15 | Name

+
📄
⬆️
✎️
🗑️
📋
🕒
🔧

Рисунок 47 – После авторизации

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

3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233

Lab19_Lukyanov

Registration

Login

lukyanov493

Password

123

SIGN UP

Рисунок 47 – До регистрации

7	lukyanov	321
8	1233	1233
9	lukyanov493	123

Lab19_Lukyanov

Selected graph: -1

Rename text: Name

Рисунок 48 – После регистрации

Создание нового графа (рис. 49 и 50):

11	7	Name	2022-10-16 17:00:53
13	7	Name11	2022-10-16 18:17:48
15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28

« < Page 1 / 1 > »

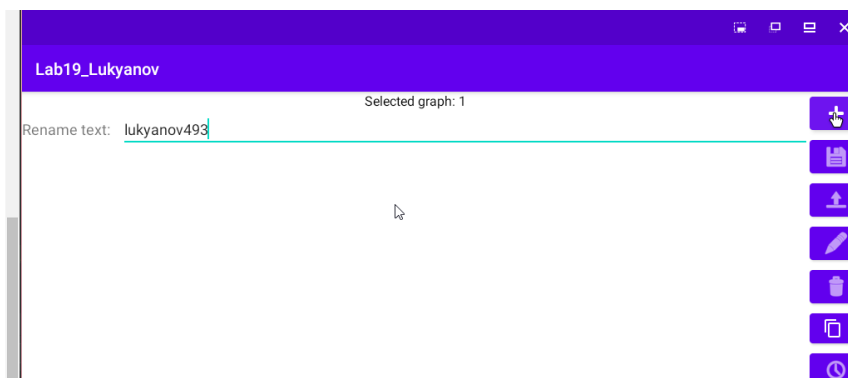


Рисунок 49 – До создания графа

11	7	Name	2022-10-16 17:00:53
13	7	Name11	2022-10-16 18:17:48
15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28
28	9	lukyanov493	2022-10-23 22:17:56

« < Page 1 / 1 > »



Рисунок 50 – После создания графа

Создание нового узла (рис. 51 и 52):

18	2	1248.0001	328.99997	
19	2	293.0	350.0	56
38	1	765.5078	514.84375	p
39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null

« < Page 1 / 1 > »

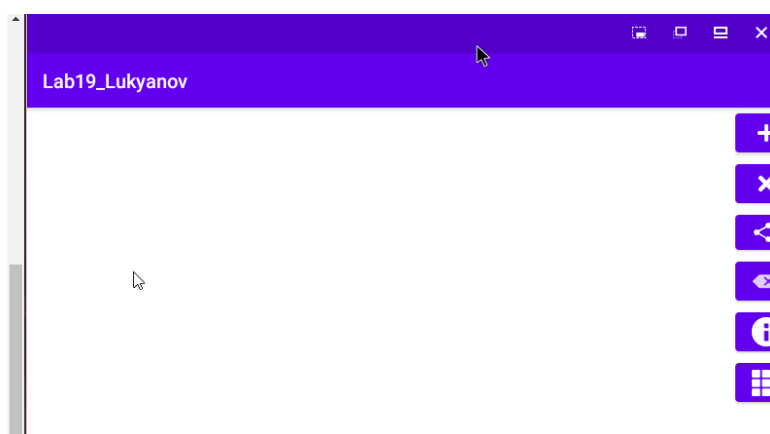


Рисунок 51 – До создания узла

18	2	1248.0001	328.99997	
19	2	293.0	350.0	56
38	1	765.5078	514.84375	p
39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null
47	28	100.0	100.0	

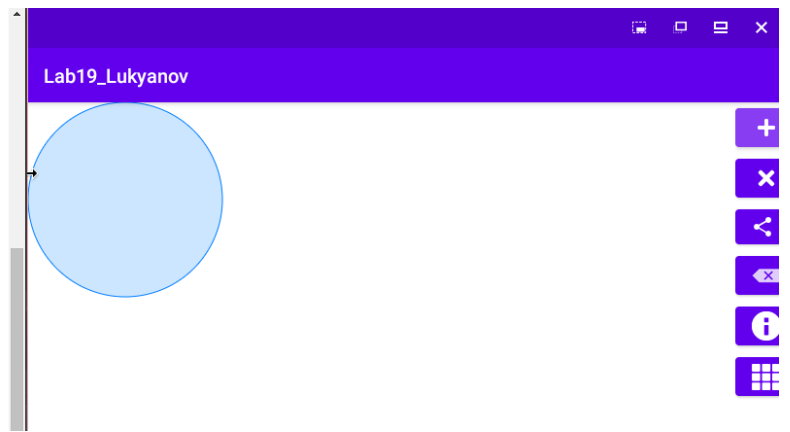


Рисунок 52 – После создания узла

Перемещение узла (рис. 53):

18	2	1248.0001	328.99997	
19	2	293.0	350.0	56
38	1	765.5078	514.84375	p
39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null
47	28	265.95938	220.58475	

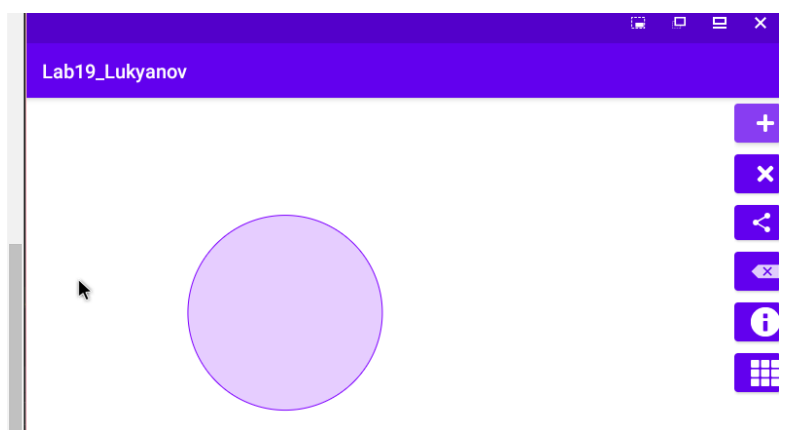


Рисунок 53 – Перемещение узла

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

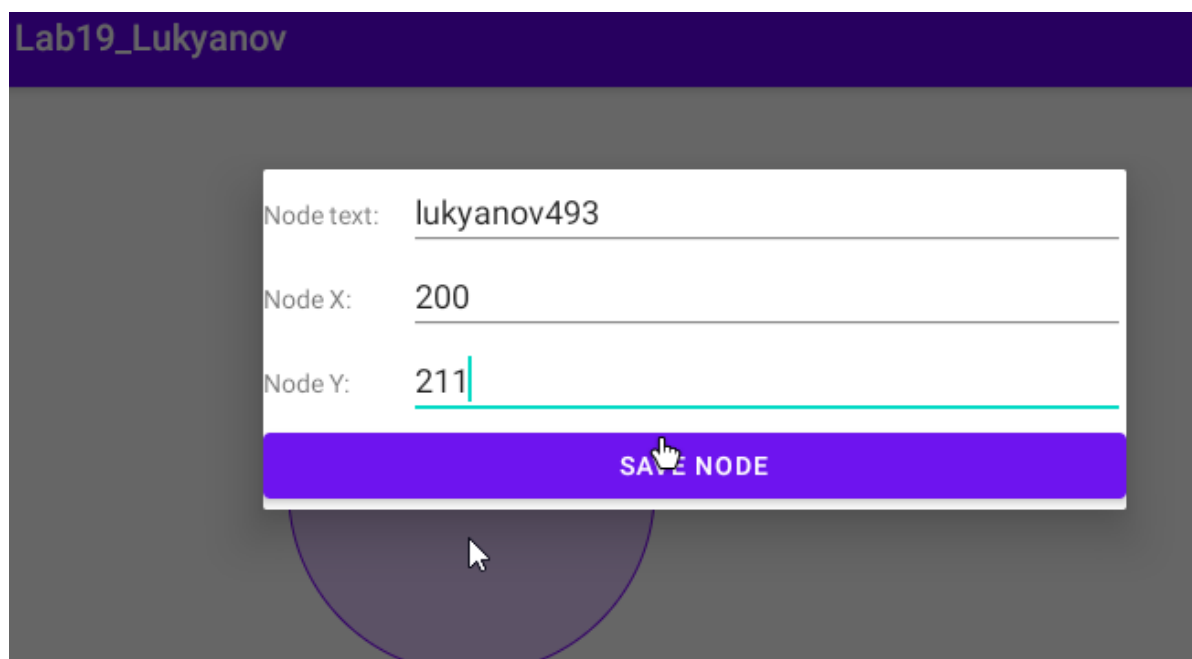


Рисунок 54 – До изменения узла

18	2	1248.0001	328.99997	
19	2	293.0	350.0	56
38	1	765.5078	514.84375	p
39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null
47	28	200.0	211.0	lukyanov493

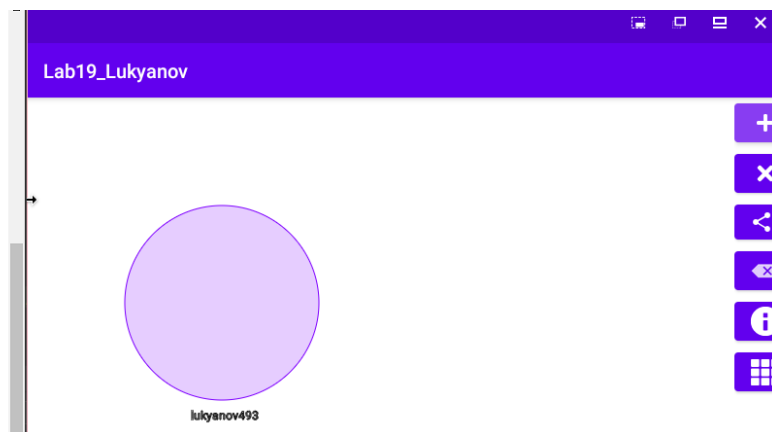


Рисунок 55 – После изменения узла

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

2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493
7	19	7	493.0
10	38	39	1.0
12	40	41	0.0

« < Page 1 / 1 > »

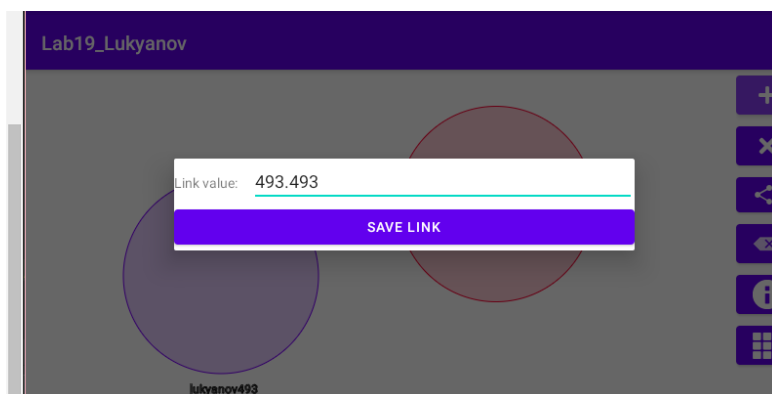


Рисунок 56 – До создания связи

2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493
7	19	7	493.0
10	38	39	1.0
12	40	41	0.0
13	47	48	493.493

« < Page 1 / 1 > »

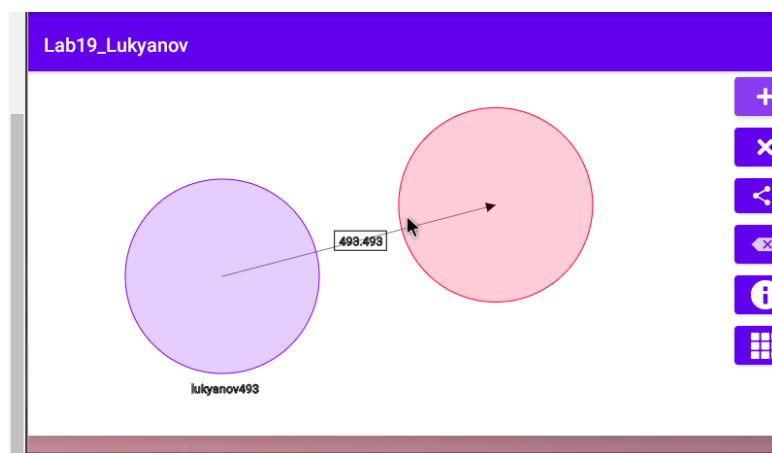


Рисунок 57 – После создания связи

Изменение значения связи (рис. 58 и 59):

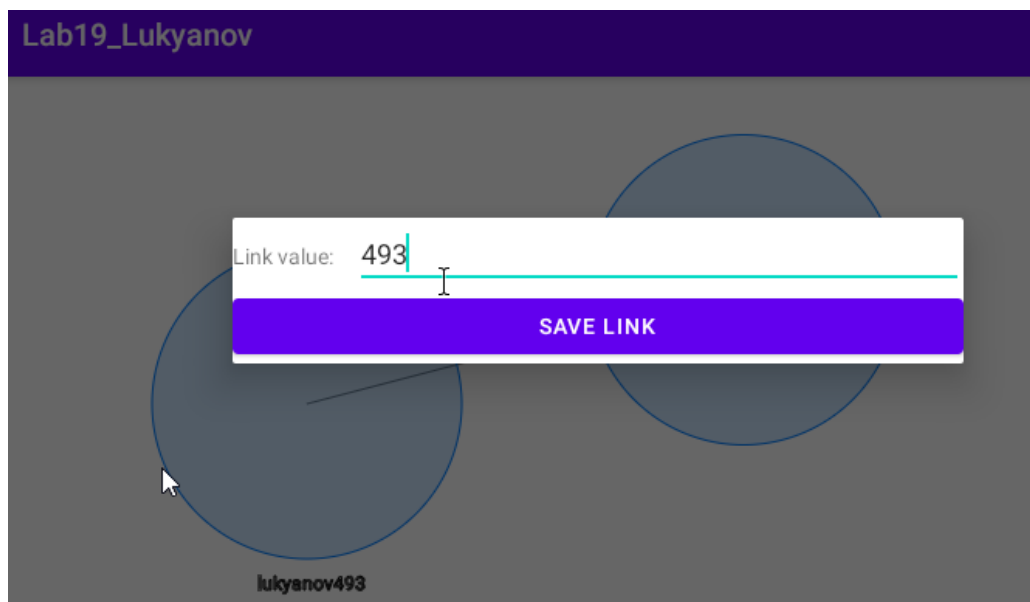


Рисунок 58 – До изменения связи

id	source	target	value
2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493
7	19	7	493.0
10	38	39	1.0
12	40	41	0.0
13	47	48	493.0

« < Page 1 / 1 > »

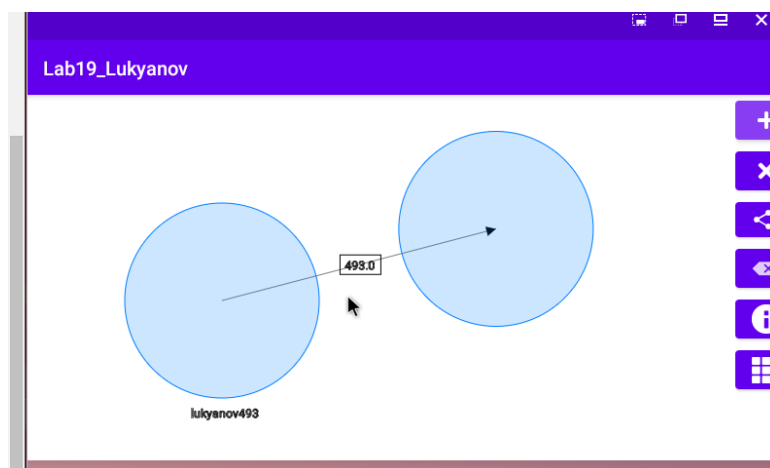


Рисунок 59 – После изменения связи

Удаление узла (рис. 60 и 61):

39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null
47	28	200.0	211.0	lukyanov493
48	28	481.8691	137.48492	

« < Page 1 / 1 > »

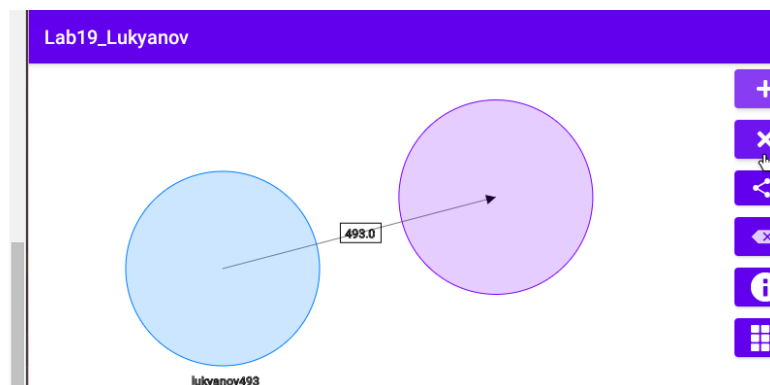


Рисунок 60 – До удаления узла

38	1	765.5078	514.84375	p
39	1	302.76367	469.14062	i
40	27	100.0	100.0	
41	27	660.9784	115.99609	null
44	27	110.01953	468.14453	null
46	27	150.03174	793.9258	null
47	28	200.0	211.0	lukyanov493

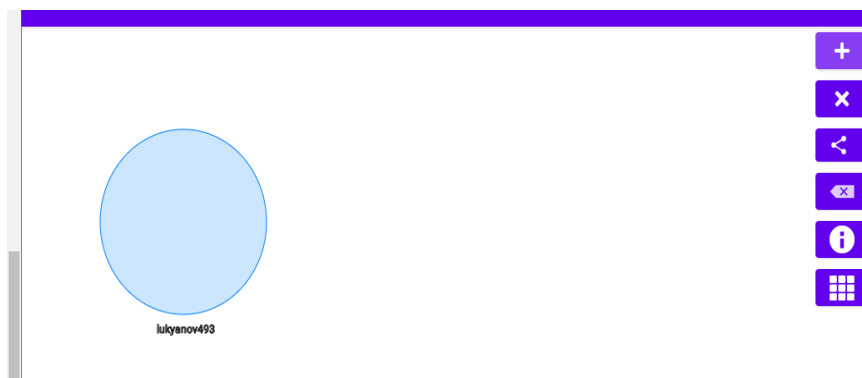


Рисунок 61 – После удаления узла

Удаление связи (рис. 62 и 63):

id	source	target	value
2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493
7	19	7	493.0
10	38	39	1.0
12	40	41	0.0
13	49	48	493.493

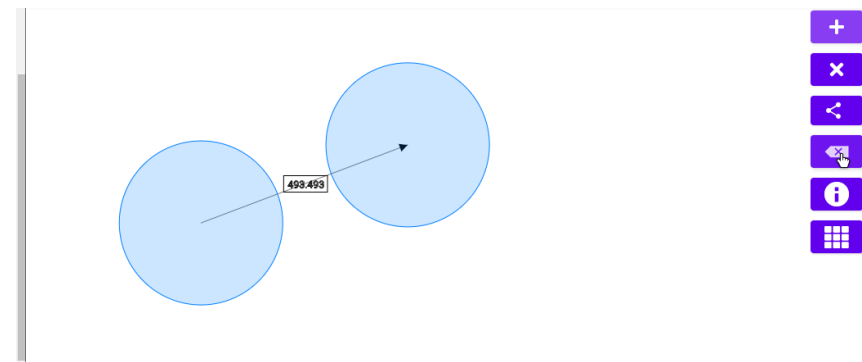


Рисунок 62 – До удаления связи

id	source	target	value
2	3	4	2.0
4	8	7	493.0
5	7	8	493.493
6	18	8	493.493
7	19	7	493.0
10	38	39	1.0
12	40	41	0.0

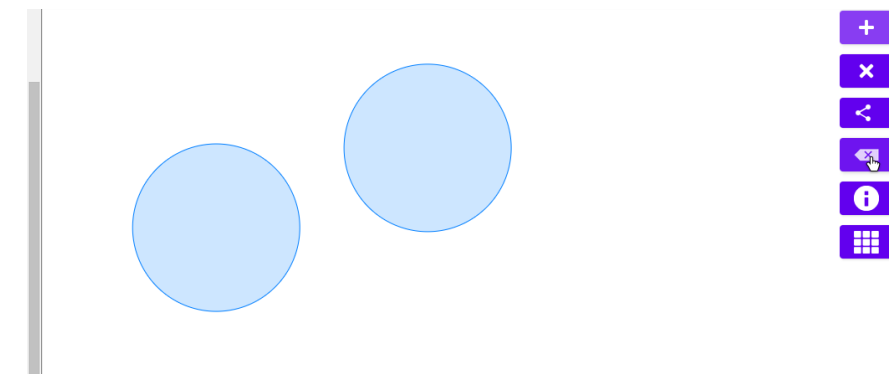


Рисунок 63 – После удаления связи

Граф для сохранения (рис. 64):

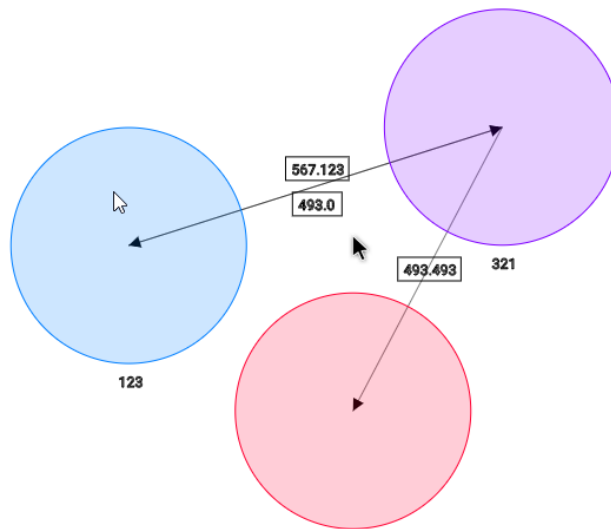


Рисунок 64 – Граф для сохранения

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):

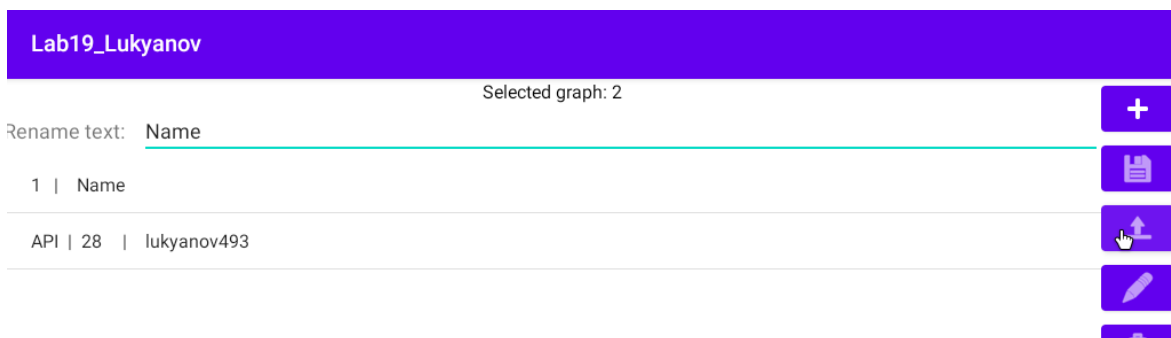


Рисунок 65 – До загрузки графа

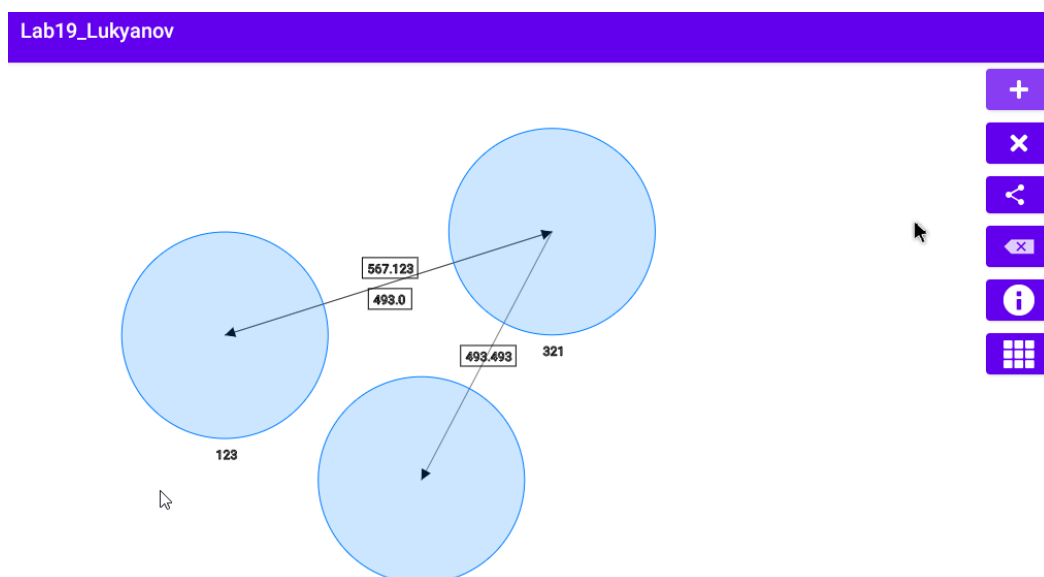


Рисунок 66 – После загрузки графа

Переименование графа (рис. 67 и 68):

ID	Г	Имя	Создано
15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28
28	9	lukyanov493	2022-10-23 22:17:56

Selected graph: 2

Rename text: 493Lukyanov

1 | Name

API | 28 | lukyanov493

Рисунок 67 – До переименования графа

13	7	Name11	2022-10-16 18:17:48
15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28
28	9	493Lukyanov	2022-10-23 22:17:56

Selected graph: 2

Rename text: 493Lukyanov

1 | Name

API | 28 | 493Lukyanov

Рисунок 68 – После переименования графа

Удаление графа (рис. 69 и 70):

15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28
28	9	493Lukyanov	2022-10-23 22:17:56

Selected graph: 2

Rename text: 493Lukyanov

1 | Name

API | 28 | 493Lukyanov

Рисунок 69 – До удаления графа

11	7	Name	2022-10-16 17:00:33
13	7	Name11	2022-10-16 18:17:48
15	7	Name	2022-10-18 04:33:12
21	7	Name	2022-10-18 08:59:17
22	7	Name	2022-10-18 08:59:40
23	7	Name11	2022-10-18 08:59:48
24	7	123	2022-10-18 09:14:46
27	6	Balanin Graph	2022-10-20 10:04:28

Selected graph: 2

Rename text: 493Lukyanov

1 | Name

Рисунок 70 – После удаления графа

Изменение пароля (рис. 71 и 72):

id	name	secret
1	Matveev	123
2		
3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233
9	lukyanov493	123

RETURN TO GRAPHLIST

New pass: SAVE

API endpoint: SAVE

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

212 | 8jwbsato6n

Рисунок 71 – До изменения пароля

id	name	secret
1	Matveev	123
2		
3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233
9	lukyanov493	5678

RETURN TO GRAPHLIST

New pass: SAVE

API endpoint: SAVE

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

212 | 8jwbsato6n

Рисунок 72 – После изменения пароля

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

Lab19_Lukyanov

RETURN TO GRAPHLIST

New pass: SAVE

API endpoint:

http://89.108.78.244:5000

http://nodegraph.spbcoit.ru:5000

http://127.0.0.1:5000

SAVE

Choose session to close

212 | 8jwbsato6n

Рисунок 73 – Список с выбором конечной точки API

Отображается список сессий пользователя (рис. 74):

209	1	2znz36rfq5	2022-10-18 09:10:32
210	1	ek8yhk2orc	2022-10-18 09:31:22
211	1	qpbztzfb8j	2022-10-19 06:53:06
212	9	8jwbsato6n	2022-10-23 22:13:37
213	9	29h8nhgqfn	2022-10-23 22:51:36
214	9	v14e0hs4wn	2022-10-23 22:51:38
215	9	5i3a3ce7v2	2022-10-23 22:51:38

« < Page 2 / 2 > »

Web-based SQLite database browser, powered by Flask and Peewee. ©

RETURN TO GRAPHLIST

New pass:

API endpoint:

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

212		8jwbsato6n
213		29h8nhgqfn
214		v14e0hs4wn
215		5i3a3ce7v2

Рисунок 74 – Отображения списка сессий пользователя с сервера

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

208	7	mbub9jk3v4	2022-10-18 09:07:22
209	1	2znz36rfq5	2022-10-18 09:10:32
210	1	ek8yhk2orc	2022-10-18 09:31:22
211	1	qpbztzfb8j	2022-10-19 06:53:06
213	9	29h8nhgqfn	2022-10-23 22:51:36
214	9	v14e0hs4wn	2022-10-23 22:51:38
215	9	5i3a3ce7v2	2022-10-23 22:51:38
216	9	4b93t07nrx	2022-10-23 22:53:23

« < Page 2 / 2 > »

Web-based SQLite database browser, powered by Flask and Peewee. ©

RETURN TO GRAPHLIST

New pass:

API endpoint:

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

213		29h8nhgqfn
214		v14e0hs4wn
215		5i3a3ce7v2
216		4b93t07nrx

Рисунок 75 – До закрытия сессии

208	7	mbub9jk3v4	2022-10-18 09:07:22
209	1	2znz36rfq5	2022-10-18 09:10:32
210	1	ek8yhk2orc	2022-10-18 09:31:22
211	1	qpbztzfb8j	2022-10-19 06:53:06
214	9	v14e0hs4wn	2022-10-23 22:51:38
215	9	5i3a3ce7v2	2022-10-23 22:51:38
216	9	4b93t07nrx	2022-10-23 22:53:23

« < Page 2 / 2 > »

Web-based SQLite database browser, powered by Flask and Peewee. ©

RETURN TO GRAPHLIST

New pass:

API endpoint:

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

214		v14e0hs4wn
215		5i3a3ce7v2
216		4b93t07nrx

Рисунок 76 – После закрытия сессии

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

ID	/	name	secret
208	7	mbub9jk3v4	2022-10-18 09:07:22
209	1	2znz36rfq5	2022-10-18 09:10:32
210	1	ek8yhk2orc	2022-10-18 09:31:22
211	1	qpbtfib8j	2022-10-19 06:53:06
214	9	v14e0hs4wn	2022-10-23 22:51:38
215	9	5i3a3ce7v2	2022-10-23 22:51:38
216	9	bu0nkw4qty	2022-10-23 22:54:38

Page 2 / 2

RETURN TO GRAPHLIST

New pass: SAVE

API endpoint: SAVE

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

214 | v14e0hs4wn

215 | 5i3a3ce7v2

216 | bu0nkw4qty

Рисунок 77 – До закрытия текущей сессии

ID	/	name	secret
205	1	maa9j4m4g	2022-10-18 08:55:43
208	7	mbub9jk3v4	2022-10-18 09:07:22
209	1	2znz36rfq5	2022-10-18 09:10:32
210	1	ek8yhk2orc	2022-10-18 09:31:22
211	1	qpbtfib8j	2022-10-19 06:53:06
214	9	v14e0hs4wn	2022-10-23 22:51:38
215	9	5i3a3ce7v2	2022-10-23 22:51:38

Page 2 / 2

Authentication SIGN UP

Login lukyanov493

Password 5678

☒ Save SIGN IN

Рисунок 78 – После закрытия текущей сессии

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

ID	name	secret
2		
3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233
9	lukyanov493	5678

Page 1 / 1

RETURN TO GRAPHLIST

New pass: SAVE

API endpoint: SAVE

DELETE CURRENT ACCOUNT

CLOSE CURRENT SESSION

Choose session to close

214 | v14e0hs4wn

215 | 5i3a3ce7v2

216 | vu58d5icsj

Рисунок 79 – До удаления текущего аккаунта

ID	name	secret
1	Matveev	123
2		
3	Matveev2	321
4	Matveev3	1234
6	Balanin	secret
7	lukyanov	321
8	1233	1233

Page 1 / 1

Authentication SIGN UP

Login lukyanov493

Password 5678

☒ Save SIGN IN

Рисунок 80 – После удаления текущего аккаунта

Приложение имеет собственную иконку (рис. 81):

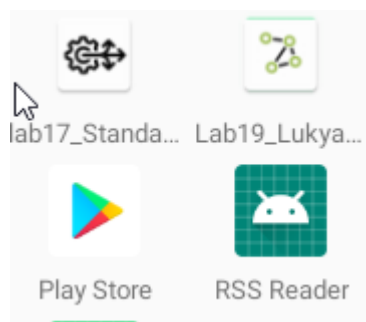


Рисунок 81 – Иконка приложения