

Документация

Емельянов Матвей

Github проекта: <https://github.com/Kekuch13/messenger/>

Краткое описание

Для сборки проекта необходимо наличие следующих библиотек и зависимостей:

- Qt6
- boost 1.81.0 или новее
- postgresSQL
- libpqxx
- Cmake версии 3.5 или новее

Для настройки проекта необходимо:

- Создать базу данных postgresql “messenger”
- Запустить для создания структуры БД файл MEGAADDER.sql (содержится в проекте Server)
- Указать свои данные, используемые для подключения к БД, в файле databaseManager.h
- Для изменения порта необходимо:
 - Изменить аргумент конструктора объекта Server в файле main.cpp проекта Server
 - Изменить аргумент конструктора объекта Connection в конструкторе Mainwidget в файле mainwidget.cpp проекта Client

Описание структуры проекта:

- Client содержит файлы:
 - main.cpp – главный файл
 - mainwidget.ui, mainwidget.h, mainwidget.cpp – файлы главного виджета
 - dialog.ui, dialog.h, dialog.cpp – файлы виджета диалога
 - connection.h, connection.cpp – класс Connection, обеспечивающий связь Клиента с Сервером
 - CMakeLists.txt – файл Cmake

- Server содержит файлы:
 - main.cpp – главный файл
 - databaseManager.h, databaseManager.cpp – класс DatabaseManager, отвечающий за связь Сервера с БД
 - server.h, server.cpp – класс Server, отвечающий за связь и обработку запросов от Клиента.
 - CMakeLists.txt – файл Cmake
 - MEGAADDER.sql – файл, создающий структуры БД

Описание взаимодействия Клиента и Сервера

Виды запросов, отправляемых клиентом, и ответов на них от сервера:

- authorization – запрос, отправляемый Клиентом, для авторизации в системе.

Содержит:

- login – введенный логин
- password – введенный пароль

```
{  
  "requestName": "authorization",  
  "login": "anna",  
  "password": "anna"  
}
```

Ответ на него от Сервера:

Содержит:

- status – наличие пары логин/пароль в БД

```
{  
  "responseName": "authorization",  
  "status": "success"  
}
```

- registration – запрос, отправляемый Клиентом, для попытки регистрации нового аккаунта.

Содержит:

- login – введенный логин
- password – введенный пароль

```
{  
  "requestName": "registration",  
  "login": "jacob",  
  "password": "jacob"  
}
```

Ответ на него от Сервера:

Содержит:

- status – удалось ли зарегистрировать нового пользователя.

```
{  
  "responseName": "registration",  
  "status": "success"  
}
```

- getDialogs – запрос, отправляемый Клиентом, для получения уже созданных диалогов.

Содержит:

- username – уникальное имя пользователя автора запроса

```
{  
  "username": "anna",  
  "requestName": "getDialogs"  
}
```

Ответ на него от Сервера:

Содержит:

- dialogs – список диалогов, каждый из которых характеризуется:
 - id – id диалога
 - username – уникальное имя другого пользователя, участвующего в диалоге.

```
{
  "responseName": "dialogs",
  "dialogs": [
    {
      "id": "8",
      "username": "donald"
    },
    {
      "id": "1",
      "username": "john"
    },
    {
      "id": "3",
      "username": "ivan"
    }
  ]
}
```

- `getNewDialogs` – запрос, отправляемый Клиентом, для получения списка пользователей с кем еще нет диалога.

Содержит:

- `username` – уникальное имя пользователя автора запроса

```
{
  "username": "anna",
  "requestName": "getNewDialogs"
}
```

Ответ на него от Сервера:

Содержит:

- `dialogs` – список уникальных имен пользователей с кем еще нет диалога

```
{
  "responseName": "NewDialogs",
  "dialogs": [
    "maria",
    "nick"
  ]
}
```

- `dialogMessages` – запрос, отправляемый Клиентом, для получения списка сообщений конкретного диалога.

Содержит:

- `username` – уникальное имя пользователя автора запроса

- dialog_id – id запрашиваемого диалога

```
{
  "username": "anna",
  "requestName": "dialogMessages",
  "dialog_id": "8"
}
```

Ответ на него от Сервера:

Содержит:

- dialog_id – id запрашиваемого диалога
- messages – список сообщений диалога, упорядоченных по времени отправления. Каждое сообщение характеризуется:
 - author – автор сообщения
 - text – текст сообщения

```
{
  "responseName": "dialogMessages",
  "dialog_id": "8",
  "messages": [
    {
      "author": "donald",
      "text": "hi"
    }
  ]
}
```

- sendNewMessage – запрос, отправляемый Клиентом, для отправки нового сообщения

Содержит:

- username – уникальное имя пользователя автора запроса
- dialog_id – id диалога, в который отправляется сообщение
- text – текст сообщения

```
{
  "username": "anna",
  "requestName": "sendNewMessage",
  "dialog_id": "8",
  "text": "hello"
}
```

Ответ на него от Сервера – заглушка, сообщающая, что всё прошло успешно.

```
{  
    "responseName": "success"  
}
```

- createDialog – запрос, отправляемый Клиентом, для создания нового диалога.

Содержит:

- username – уникальное имя пользователя автора запроса
- buddy – уникальное имя пользователя, с которым текущий пользователь хочет создать диалог

```
{  
    "username": "anna",  
    "requestName": "createDialog",  
    "buddy": "maria"  
}
```

Ответ на него от Сервера:

В случае, если диалог между данными двумя пользователями еще не был создан, то создается новая запись в БД и отправляется id только что созданного диалога.

Иначе, если запись о диалоге с данными двумя пользователями уже есть в БД, то отправляется id существующего диалога.

Содержит:

- dialog_id – id диалога

```
{  
    "responseName": "createdDialog",  
    "dialog_id": "9"  
}
```