**Требуется:**

Реализовать клиент-серверное приложение "Чат".

***Клиент:***

Консольное приложение, с возможностью отправки сообщений в чаты на сервере.

При запуске приложения указывается имя пользователя и пароль, которые отправляются на сервер.

* POST /users – регистрация. В теле указываем имя, логин, пароль. В ответ приходит токен. Если логин не уникальный выслать 403 – запрещенно.
* POST /login, в хеадерах передаем логин и пароль.

В ответ на сообщение приходит 200 – если пользователь авторизован и 401 – если такого польователя нет. Вместе с 200 ответом в хеадере передаем токен. В теле передаем имя пользвателя и его id.

Далее, есть три возможности:

* Создать чат: POST /chats, в теле передаем название чата. Указываем в заголовке токен. В ответ приходит – id чата и 201.
* Зайти в существующий чат: сначала надо получить список всех чатов GET /chats, приходит массив ChatDto {id-чата, name-название чата}. Далее, делаем POST /chats/{chat-id}/members, тело запроса пустое. В ответ приходит 200 – если все хорошо.
* Получить сообщения из чата: LONG Pooling GET /chats/{chat-id}/messages
* Поулчить все сообщения GET /chats/{chat-id}/messages?get=all
* Написать сообщение в чат: POST /chats/{chat-id}/messages

MessageDto – на GET: {id-сообщения, text, name-пользователя}, в POST – только text

----------------------------------------------------------------------------

МЕНЮ:

1. Регистрация
2. Авторизация

-----

1. Список чатов

-----

1. Добавить чат
2. Войти в чат

Архитектурно:

ChatClientService – runService() – запустить сервис чата

Menu – show() – показать меню

MessagesHandler – обработчик сообщений с сервера – посылает запросы

-----------------

**Сервер:**

Dto – слой JSON-данных

Converters – model to dto

Controllers – все запросы

Models – модели

Service – основная логика.

DAO

UsersDao

List<User> findAll()

User find(Integer id)

Integer save(User user)

void delete(int id)

void update(User user)

void saveUserToChat(Integer userId, Integer chatId)

ChatsDao

MessagesDao

List<Messages> findAllByChatId(Integer chatId)