

## Логика работы приложения

1. Пользователь регистрируется в системе, после чего сразу попадает в главное окно приложения со списком чатов (на этапе загрузки основного окна получить список чатов и заполнить соответствующий блок).
2. Список чатов необходимо обновлять периодически (например, раз в 100 мсек.). Так можно получить новые чаты и информацию о новых сообщениях в чате (отслеживая поле *chat\_last\_message*).
3. При клике на чат в окне сообщений отображаются сообщения данного чата. При клике на другой чат – окно сообщений очищается и в него загружаются сообщения из другого чата.
4. Список сообщения при открытом окне чатов необходимо обновлять периодически (например, раз в 100 мсек.).
5. При отправке сообщения в чат не нужно добавлять его в окно сообщений (оно должно загрузиться стандартным способом – при обновлении окна сообщений.)
6. Сообщения передаются с сервера упорядоченно по времени – первое сообщение в массиве самое новое.
7. При выходе из системы (предусмотреть кнопку в окне приложения) пользователь перенаправляется на окно авторизации.

**При отправке любого запроса в случае получения сообщения ответа 401 – Login Failed пользователь должен быть переадресован на окно авторизации.**

Вам необходимо реализовать **REST API** заданной структуры.

В примерах будет использоваться переменная `{{host}}` которая обозначает адрес: <http://api-messenger.web-srv.local>

Идентификацию пользователя реализуется посредством **Bearer Token**.

При попытке доступа гостя к защищенным авторизацией функциям системы во всех запросах система возвращает ответ следующего вида:

```
Status: 401  
Content-Type: application/json  
Body:  
{  
  "message": "Login Failed"  
}
```

---

При попытке доступа авторизованным пользователем к функциям недоступным для него во всех запросах система возвращает ответ следующего вида:

```
Status: 403  
Content-Type: application/json  
Body:  
{  
  "message": "Forbidden for you"  
}
```

---

При попытке получить несуществующий ресурс система возвращает ответ следующего вида:

```
Status: 404  
Content-Type: application/json  
Body:  
{  
  "message": "Not Found"  
}
```

---

В случае ошибок связанных с валидацией данных во всех запросах система возвращает ответ следующего вида:

```
Status: 422  
Content-Type: application/json  
Body:  
{  
  "message": "<содержание ошибки>"  
}
```

---

## ОПИСАНИЕ ФУНКЦИОНАЛА ГОСТЯ

### Регистрация

При успешной регистрации возвращается сгенерированный токен добавленного пользователя (пользователь автоматически авторизован в системе).

Функция должна принимать следующие параметры:

- **email** - email пользователя обязательный, валидный e-mail адрес, уникальный
- **pass** - пароль пользователя обязательный, состоит минимум из 3 символов, из которых как минимум одна строчная и одна цифра
- **fam** - имя, обязательное
- **name** – фамилия обязательное
- **otch** - отчество, обязательное

Запрос	Ответ (успех)
<b>Метод: POST</b> <b>URL:</b> { {host} }/user/  <b>Body:</b> { «email»: «user@mail.ru», «pass»: «123qw123», «fam»: «Иванов», «name»: «Иван», «otch»: «Иванович» }	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "message": "Регистрация успешна", "Data": { "id": "14", "email": "user@usermail.ru", "pass": "123qw123", "fam": "Иванов", "name": "Иван", "otch": "Иванович", "photo_link": "/files/photos/default_men.png", "token": "9ia65d7b423j8e0cfg1h", "last_request": "2025-05-10 02:51:28" } }

## Авторизация

При успешной аутентификации возвращается сгенерированный токен пользователя.

Функция должна принимать следующие параметры:

- **email** - email пользователя
- **pass** - пароль пользователя

Запрос	Ответ (успех)
<b>Метод: POST</b> <b>URL:</b> { {host} }/auth/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "message": "Доступ разрешен", "Data": { "id": "14", "email": "user@usermail.ru", "pass": "123qw123", "fam": "Иванов", "name": "Иван", "otch": "Иванович", "photo_link": "/files/photos/default_men.png", "token": "4jfdg6c8i79e52hab301", "last_request": "2025-05-10 02:51:28" } }

## ОПИСАНИЕ ФУНКЦИОНАЛА АВТОРИЗИРОВАННОГО ПОЛЬЗОВАТЕЛЯ

### Выход из системы (сброс авторизации)

Запрос предназначен для очистки значение токена пользователя.  
Функция не принимает параметры.

Запрос	Ответ (успех)
<b>Метод: DELETE</b> <b>URL:</b> { {host} }/auth/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "success": true, "message": "Logout" }

### Получение данных о пользователе

Запрос предназначен для получения данных о каком-либо пользователе системы.

Функция должна принимать следующие параметры:

- **email** - email пользователя системы

Запрос	Ответ (успех)
<b>Метод: GET</b> <b>URL:</b> { {host} }/user/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "id": "5", "email": "user@mail.ru", "fam": "Иванов", "name": "Иван", "otch": "Иванович", "photo_link": "/files/photos/default_men.png" }

Получение данных текущем о пользователе

Запрос предназначен для получения данных о каком-либо пользователе системы.

Функция не принимает параметры.

Запрос	Ответ (успех)
<b>Метод:</b> GET <b>URL:</b> { {host} }/user/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "id": "5", "email": "user@mail.ru", "pass": "123qw123", "fam": "Иванов", "name": "Иван", "otch": "Иванович", "photo_link": "/files/photos/default_men.png", "token": "a9c61gi3jf7284hd0eb5", "last_request": "2025-05-10 03:04:41" }

### Изменение данных текущем о пользователе

Запрос предназначен для получения данных о каком-либо пользователе системы.

Функция должна принимать следующие параметры (которые изменяют текущие):

- **pass** – пароль пользователя системы
- **fam** – фамилия пользователя системы
- **name** – имя пользователя системы
- **otch** – отчество пользователя системы
- **photo\_link** – фото пользователя системы

Запрос	Ответ (успех)
<b>Метод: PUT</b> <b>URL: { {host} }/user/</b>	<b>Status: 200</b> <b>Content-Type: application/json</b> <b>Body:</b> { "id": "5", "email": "user@mail.ru", "pass": "123qw123", "fam": "Иванов", "name": "Иван", "otch": "Иванович", "photo_link": "/files/photos/default_men.png", "token": "a9c61gi3jf7284hd0eb5", "last_request": "2025-05-10 03:04:41" }

### Удаление текущего пользователя

Запрос предназначен для удаления текущего пользователя из системы.

Функция не принимает параметры.

Запрос	Ответ (успех)
<b>Метод: DELETE</b> <b>URL: { {host} }/user/</b>	<b>Status: 200</b> <b>Content-Type: application/json</b> <b>Body:</b> { "message": "Пользователь успешно удален", "Data": null }

### Создание нового чата с пользователем

Запрос предназначен для создания нового чата с пользователем.

Функция должна принимать следующие параметры:

- **email** - email пользователя системы, с которым будет создан чат

Запрос	Ответ (успех)
<b>Метод: POST</b> <b>URL:</b> {{host}}/chats/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "message": "Чат создан", "Data": { "chat_id": 25, "chat_name": "Чат с Акулов А. В", "chat_users": [ { "email": "user@inbox.ru", "fam": "Петров", "name": "Петр", "otch": "Викторович", "photo_link": "/files/photos/default_men.png" }, { "email": "akkullov@inbox.ru", "fam": "Акулов", "name": "Андрей", "otch": "Викторович", "photo_link": "/files/photos/default_men.png" } ] } }



### Получить список чатов

Запрос предназначен для получения списка чатов пользователя.  
Также, по параметру **chat\_last\_message** необходимо отслеживать поступление новых сообщений в чат.  
Функция не принимает параметры.

Запрос	Ответ (успех)
<b>Метод:</b> GET <b>URL:</b> {{host}}/chats/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> [ { "chat_id": "25", "chat_name": "Топ чат с Петров П. П. 2", "chat_last_message": "2025-05-12 01:32:25", "companion_email": "akkulow@inbox.ru", "companion_fam": "Акулов", "companion_name": "Андрей", "companion_otch": "Викторович", "companion_photo_link": "/files/photos/default_men.png" }, { "chat_id": "29", "chat_name": "Чат с Ян М. А. ", "chat_last_message": "2025-05-12 03:11:08", "companion_email": "yan@mail.com", "companion_fam": "Ян", "companion_name": "Мармелад", "companion_otch": "Анатолевич", "companion_photo_link": "" } ]

## Изменение имени чата

Запрос предназначен изменения имени чата.

Функция должна принимать следующие параметры (которые изменяют текущие):

- **chat\_id** – ID чата
- **chat\_name** – новое имя чата

Запрос	Ответ (успех)
<b>Метод:</b> PUT <b>URL:</b> {{host}}/chats/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "message": "Чат изменен", "Data": { "chat_id": "25", "chat_name": "Топ чат с Петров П. П. 2", "chat_users": [ { "email": "user@inbox.ru", "fam": "Петров", "name": "Петр", "otch": "Викторович", "photo_link": "/files/photos/default_men.png" }, { "email": "akkullov@inbox.ru", "fam": "Акулов", "name": "Андрей", "otch": "Викторович", "photo_link": "/files/photos/default_men.png" } ] } }

### Получить сообщения для чата

Запрос предназначен для получения всех сообщений, которые были отправлены в чат. Ответ приходит в виде массива, отсортированного в порядке убывания по времени отправления сообщения.

Также, с каждым сообщением приходит полная информация о его отправителе.

Функция должна принимать следующие параметры (которые изменяют текущие):

- **chat\_id** – ID чата

Запрос	Ответ (успех)
<b>Метод:</b> GET <b>URL:</b> {{host}}/messages/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> [ { "id": "6", "chat_id": "29", "sender_id": "6", "datetime_create": "2025-05-12 03:11:08", "text": "Сообщение в чат 29", "sender": { "id": "6", "email": "user@inbox.ru", "fam": "Петров", "name": "Петр", "otch": "Викторович", "photo_link": "/files/photos/default_men.png" } } ]

### Отправить сообщение в чат

Запрос предназначен для отправки сообщения в чат.

Функция должна принимать следующие параметры (которые изменяют текущие):

- **chat\_id** – ID чата
- **text** – текст сообщения

Запрос	Ответ (успех)
<b>Метод:</b> POST <b>URL:</b> {{host}}/chats/	<b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b> { "message": "Сообщение отправлено", "Data": true }