**Логика работы приложения**

1. Пользователь регистрируется вы 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**](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: 403  
Content-Type: application/json  
Body:  
{  
"message": "Forbidden for you"  
}**

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) |
| **Метод: POST**  **URL:** {{host}}/user/  **Body:**  {  «email»: «user@mail.ru»,  «pass»: «123qw123»,  «fam»: «Иванов»,  «name»: «Иван»,  «otch»: «Иванович»  } | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: POST**  **URL:** {{host}}/auth/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: DELETE**  **URL:** {{host}}/auth/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: GET**  **URL:** {{host}}/user/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: GET**  **URL:** {{host}}/user/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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 изменят 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) |
| **Метод: PUT**  **URL:** {{host}}/user/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: DELETE**  **URL:** {{host}}/user/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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) |
| **Метод: POST**  **URL:** {{host}}/chats/ | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "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": "akkullow@inbox.ru",  "fam": "Акулов",  "name": "Андрей",  "otch": "Викторович",  "photo\_link": "/files/photos/default\_men.png"  }  ]  }  } |