



ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ

API PHP Сокет чат сервер

Оглавление

| | |
|--------------------------------|---|
| Работа с API..... | 1 |
| Установка соединения..... | 2 |
| Создать пользователя..... | 2 |
| Авторизация пользователя | 3 |
| Редактирование сообщения | 4 |
| Удаление сообщения | 5 |
| Чтение списка сообщений | 6 |

Работа с API

Функционал API построен на обмене JSON сообщениями между клиентом и сервером с дописыванием в эти запросы (команды) результирующих значений. Сетевое взаимодействие использует веб-сокеты.

Запрос обрабатывается на стороне сервера или клиента, а результат выполнения при необходимости дописывается в исходный запрос (команду), который возвращается обратно в ответе в формате json.

Дописываемые поля:

- **status:**
 - 0 – не успешная операция
 - 1 – успешно выполненная операция
- **answer:**
 - текстовое описание выполнения операции (текст зависит от результата выполнения запроса и для команд он индивидуален)

В ответ могут дописываться другие поля:

- id – номер созданной записи в случае создания или авторизации
- name – логин пользователя
- email - пользователя
- user_key: ключ авторизации пользователя

- и др.

Отправлять сообщения в чат возможно только авторизованным пользователям.

Схема работы следующая

1. Сначала используя API создается пользователь.
2. Далее через API необходимо авторизоваться пользователя по его почте и паролю. В ответ на успешную авторизацию, сервер вернет ключ авторизации в поле `user_key`.
3. И теперь получив ключ авторизации `user_key` появилась возможность совершать действия от имени этого пользователя:
 - A. Добавление новых сообщений
 - B. Изменение сообщений
 - C. Удаление сообщений

Установка соединения

В момент установки соединения, сервер отправляет команду содержащую идентификатор соединения:

```
{"command": "connection_open",  
  "id": "Resource id #13"}
```

Создать пользователя

Пример JSON запроса:

```
{"command": "user_create",  
  "name": "Тестовый пользователь",  
  "email": "test@test1.ru",  
  "password": "test"}
```

Пример JSON ответа:

```
{"command": "user_create",  
  "name": "Тестовый пользователь",  
  "email": "test@test1.ru",  
  "password": "test",  
  "status": 1,  
  "answer": "Пользователь создан",  
  "id": 1}
```

Варианты answer:

- Если отправлены пустые поля формы
 - Не задано имя
 - Не задана почта

- Не задан пароль
- В случае успеха
 - Пользователь создан
- В случае ошибок
 - Пользователь с таким именем есть
 - Пользователь с такой почтой есть

Авторизация пользователя

Пример JSON запроса:

```
{ "command": "user_login",
  "email": "test@test1.ru",
  "password": "test" }
```

Пример JSON ответа:

```
{ "command": "user_login",
  "email": "test@test1.ru",
  "password": "test",
  "status": 1,
  "answer": "Пользователь авторизован",
  "user_key": "5fe0af1b32c2a" }
```

Варианты answer:

- Если отправлены пустые поля формы
 - Не задана почта
 - Не задан пароль
- В случае успеха
 - Пользователь авторизован
- В случае ошибок
 - Пользователь с такой комбинацией почты и пароля не найден

Выход пользователя:

Пример JSON запроса:

```
{ "command": "user_exit",
  "user_key": "as6a876aa876s" }
```

Пример JSON ответа:

```
{ "command": "user_exit",
  "user_key": "5fe0af1b32c2a",
  "answer": "Пользователь вышел",
  "status": 1 }
```

Варианты answer:

- В случае успеха
 - Пользователь вышел

Создание нового сообщения:

Пример JSON запроса:

```
{ "command": "message_create",  
  "message": "Привет",  
  "time": 1608561426405,  
  "user_key": "5fe0b1eccac31" }
```

Пример JSON ответа:

```
{ "command": "message_create",  
  "message": "Привет",  
  "time": 1608561426405,  
  "user_key": "5fe0b1eccac31",  
  "name": "Тестовый пользователь",  
  "status": 1,  
  "answer": "Сообщение создано",  
  "email": "test@test1.ru" }
```

Варианты answer:

- Если отправлены пустые поля формы
 - Пустое сообщение
 - Не задан ключ авторизации пользователя
- В случае успеха
 - Сообщение создано
- В случае ошибок
 - Сообщение не создано. Ключ авторизации уже не действителен.

Редактирование сообщения

Пример JSON запроса:

```
{ "command": "message_edit",  
  "id": 20,  
  "message": "Текст сообщения",  
  "user_key": "as6a876aa876s" }
```

Пример JSON ответа:

```
{ "command": "message_edit",
```

```
"id":20
"message":"Текст сообщения",
"user_key":"as6a876aa876s",
"status":1,
"answer":"Сообщение отредактировано"}
```

Варианты answer:

- Если отправлены пустые поля формы
 - Пустое сообщение
 - Не задан ключ авторизации пользователя
 - Не указан ИД сообщения
- В случае успеха
 - Сообщение отредактировано
- В случае ошибок
 - Сообщение не отредактировано. Ключ авторизации уже не действителен.
 - Сообщение не отредактировано. Такого сообщения нет
 - Сообщение не отредактировано. Редактирование чужого сообщения.

Удаление сообщения

Пример JSON запроса:

```
{"command":"message_del",
"id":20
"user_key":"as6a876aa876s"}
```

Пример JSON ответа:

```
{"command":" message_del ",
"id":20
"user_key":"as6a876aa876s",
"status":1,
"answer":"Сообщение удалено"}
```

Варианты answer:

- Если отправлены пустые поля формы
 - Не задан ключ авторизации пользователя
 - Не указан ИД сообщения
- В случае успеха
 - Сообщение удалено
- В случае ошибок
 - Сообщение не удалено. Ключ авторизации уже не действителен.
 - Сообщение не удалено. Такого сообщения нет
 - Сообщение не удалено. Удаление чужого сообщения.

Чтение списка сообщений

Пример JSON запроса:

```
{"command": "messages_read",  
  "start": "0",  
  "end": "100"}
```

Пример JSON ответа:

```
{"command": "messages_read",  
  "start": "0",  
  "end": "100",  
  "status": 1,  
  "answer": "Список прочитан",  
  "messages": [Массив сообщений]}
```

Варианты answer:

- В случае успеха
 - Список прочитан