

API – документация

Работа с API происходит в следующем формате – на сервер отсылается запрос в формате JSON следующего формата:

```
{
    "type" : "api",
    "action" : "<метод>",
    <параметры>
}
```

где <метод> - название метода API, <параметры> - параметры метода в формате

```
"<параметр 1>" : "<значение 1>",
"<параметр 2>" : "<значение 2>",
...
"<параметр N>" : "<значение N>",
```

В ответ сервер возвращает ответ в формате строки (по умолчанию) или JSON (в таком случае будет указано в документации к методу).

Описание методов API см на следующих страницах

get_nick

Получает никнейм пользователя по его ID

Параметры:

- *user_id* – ID пользователя

Возвращаемое значение:

- Ник пользователя, если он существует
- *-1*, если пользователя с указанным ID не существует
- *-10*, если произошла ошибка сервера или БД

get_id

Получает ID пользователя по его никнейму или email

Параметры:

- *login* – ник или email пользователя

Возвращаемое значение:

- ID пользователя, если он существует
- *-1*, если пользователя с указанным ID не существует
- *-10*, если произошла ошибка сервера или БД

auth

Пытается провести авторизацию по нику/email и паролю

Параметры:

- *login* – ник или email пользователя
- *pass* – пароль, зашифрованный алгоритмом MD5

Возвращаемое значение:

- 2 значения "*user_id*" и "*session_key*" через пробел, если такой пользователь существует
- *-1*, если пользователя с указанным ID не существует
- *-2*, если такой пользователь существует, но сессия была разорвана (например, предпринята попытка авторизации при активной сессии). Повторите процесс авторизации.
- *-10*, если произошла ошибка сервера или БД

reg

Регистрирует нового пользователя с заданными параметрами

Параметры:

- *email* – email пользователя
- *nick* – никнейм пользователя
- *pass* – пароль, зашифрованный алгоритмом MD5

Возвращаемое значение:

- ID нового пользователя, если он был зарегистрирован
- -2, если передан некорректный email
- -3, если передан некорректный никнейм (он должен содержать только латинские буквы и символ «_», длина 3-20 символов)
- -4, если пользователь с указанным email уже существует
- -5, если пользователь с указанным никнеймом уже существует
- -1, -10, если все требования выполнены, но регистрацию не удалось завершить (проблемы с сервером или БД)

change_nick

Меняет никнейм пользователя

Параметры:

- *oldnick* – текущий никнейм пользователя
- *newnick* – новый никнейм пользователя
- *pass* – пароль, зашифрованный алгоритмом MD5

Возвращаемое значение:

- 1, если смена никнейма произошла успешно
- -1, если пользователя с таким никнеймом нет
- -2, если неверный пароль пользователя
- -3, если новый никнейм уже занят
- -4, если новый никнейм некорректен (он должен содержать только латинские буквы и символ «_», длина 3-20 символов)
- 0, -10, если все требования выполнены, но смену не удалось завершить (ошибка сервера или БД)

send_req_reset_pass

Создает запрос для сброса пароля и отправляет код на email

Параметры:

- *email* – email пользователя

Возвращаемое значение:

- *1*, если код для смены пароля успешно отправлен
- *-1*, если пользователя с таким email нет
- *-10*, если возникла внутренняя ошибка сервера или БД

reset_pass

Сбрасывает пароль

Параметры:

- *email* – email пользователя
- *newpass* – новый пароль, зашифрованный MD5
- *key* – проверочный код, высланный на почту

Возвращаемое значение:

- *1*, если пароль успешно изменен
- *-1*, если пользователя с таким email нет
- *-10*, если возникла внутренняя ошибка сервера или БД

send_to_friend

Отправляет заявку в друзья другому пользователю

Параметры:

- *session* – токен сессии
- *user1* – ID пользователя, от кого отправляется заявка
- *user2* – ID пользователя, кому отправляется заявка

Возвращаемое значение:

- *1*, если заявка отправлена
- *-1*, если сессия недействительна (требуется повторная авторизация)
- *-10*, если возникла внутренняя ошибка сервера или БД

get_to_friend

Получает список нерассмотренных заявок в друзья

Параметры:

- *session* – токен сессии
- *user_id* – ID пользователя, кому отправляется заявка

Возвращаемое значение:

- ID пользователей через пробел, от которых пришли заявки, если они есть
- *0*, если новых заявок нет
- *-1*, если сессия недействительна (требуется повторная авторизация)
- *-10*, если возникла внутренняя ошибка сервера или БД

Чат – документация

Общий чат

Отправка сообщения в общий чат происходит в следующем формате – на сервер отсылается запрос в формате JSON следующего формата:

```
{
    "type" : "general_chat",
    "action" : "write"
    "user_id" : "<ID>",
    "session_id" : "<session>",
    "text" : "<text>"
}
```

где <ID> - ID пользователя, <text> - текст сообщения, <session> - ID сессии.

Для получения списка сообщений на сервер отсылается запрос в формате JSON следующего формата:

```
{
    "type" : "general_chat",
    "action" : "get"
    "user_id" : "<ID>",
    "session_id" : "<session>",
    "start" : "<message_id>"
}
```

где <ID> - ID пользователя, <message_id> - ID последнего сообщения, которое было получено клиентом, <session> - ID сессии. В ответ возвращается список сообщений в формате JSON, где каждое сообщение хранится в формате JSON с полями <message_id>, <text>, <nickname>. Если новых сообщений нет, возвращается пустая строка.

Для получения ID последнего сообщения на сервер отсылается запрос в формате JSON следующего формата:

```
{
    "type" : "general_chat",
    "action" : "get_id"
    "user_id" : "<ID>",
    "session_id" : "<session>",
}
```

где `<ID>` - ID пользователя, `<session>` - ID сессии.

Возвращаемые значения ошибок методов чата: -1 (исходные данные неверны), -10 (ошибка БД или сервера), пустая строка (нет данных).