



FHQ 2014 Manual's

Doc-Version: 0

Doc-Date-Build: 3 декабря 2014 г.

Pages: 5

Dev.: Evgenii Sopov

Dev.: Dmitrii Mukovkin

Tomsk

Russian Federation

Web-site: <http://fhq.keva.su>

E-mails: mrseakg@gmail.com

With support by



Оглавление

1	Configurations	2
1.1	FIREWALL (CentOS 7 + firewalld)	2
2	Список методов и их описание	3
2.1	Общее описание	3
2.2	Пользователи	3
2.2.1	user.get	3
2.2.2	user.get - modify	4
2.2.3	user.list	4
2.2.4	user.list - modify	4
2.2.5	user.delete	5
2.2.6	user.search	5
2.2.7	user.search - modify	5

Глава 1

Configurations

1.1 FIREWALL (CentOS 7 + firewalld)

Some settings.

CentOS and MariaDB and Firewall.

CentOS 7 + firewalld:

```
# firewall-cmd --zone=public --add-port=3306/tcp --permanent
# firewall-cmd --reload
```

Глава 2

Список методов и их описание

В этой главе будут рассмотрены основные положения при работе с api.

2.1 Общее описание

Для работы с API необходимо осуществлять **GET** или **POST** запрос по протоколу **HTTP** на указанный по примеру URL.

```
https://api.example.com/method/"METHOD_NAME"? "PARAMETERS"&access_token="ACCESS_TOKEN"
```

- **METHOD_NAME** - имя метода из списка, приведенного ниже,
- **PARAMETERS** - дополнительные параметры, указанные в описании метода
- **ACCESS_TOKEN** - ключ доступа, полученный в результате успешной авторизации приложения.

Примечание:

При осуществлении запросов с объемом данных больше 2 кб, необходимо использовать метод POST.

2.2 Пользователи

Методы для работы с пользователями

2.2.1 user.get

Возвращает краткую информацию о пользователе

Метод не требует access_token

Входные параметры:

- **id** - Идентификатор пользователя (Целое положительное число)

Example

```
https://api.example.com/method/user.get?id=11&access_token={token}
```

Примечание:

При осуществлении запросов с объемом данных больше 2 кб, необходимо использовать метод POST.

2.2.2 user.get - modify

Думаю что метод должен требовать access_token

2.2.3 user.list

Возвращает список пользователей с краткой информацией о них

Используется пагинация

Список зарегистрированных пользователей.

Входные параметры:

- **нет**

Example 1

```
https://api.example.com/method/user.list?access_token={token}
```

Example 2

```
https://api.example.com/method/user.list?page=0
```

2.2.4 user.list - modify

Хотелось бы также сортировку по:

- **nick** - Никнейм пользователя
- **last_activity** - Дата последней активности

Хотелось бы также мочь фильтровать по:

- **nick** - Никнейм пользователя
- **last_activity_begin** - Дата последней активности начиная с такой то даты
- **last_activity_end** - Дата последней активности до такой то даты
- **country** - страна (собранный инфа по ip адресам)
- **city** - город (собранный инфа по ip адресам)
- **team** - по текущей команде
- **activation** - По статусу активации

2.2.5 user.delete

Удаление пользователя.

Для выполнения метода необходимо иметь права “admin”

Входные параметры:

- **id** - Идентификатор пользователя (Целое положительное число)

Example

```
https://api.example.com/method/user.delete?id=11&access_token={token}
```

2.2.6 user.search

Поиск зарегистрированных пользователей по никнейму.

Используется пагинация.

Входные параметры:

- **nick** - Никнейм пользователя (Строка)

Example

```
https://api.example.com/method/user.search?nick=2ks&access_token={token}
```

2.2.7 user.search - modify

Хотелось бы также сортировку по:

- **nick** - Никнейм пользователя
- **last_activity** - Дата последней активности

Хотелось бы также мочь фильтровать по:

- **nick** - Никнейм пользователя
- **last_activity_begin** - Дата последней активности начиная с такой то даты
- **last_activity_end** - Дата последней активности до такой то даты
- **country** - страна (собранный инфа по ip адресам)
- **city** - город (собранный инфа по ip адресам)
- **team** - по текущей команде
- **activation** - По статусу активации