



Техническая документация к FHQ 2014 г от Р.Х.

Версия документации: 19

Дата сборки док.: 14 декабря 2014 г.

Количество страниц: 15

Developer: Evgenii Sopov

Developer: Dmitriy Mukovkin

Designer: Taisiya Lebedeva

Developer: Alexander Menschikov

Tomsk

Russian Federation

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

E-mails: mrseakg@gmail.com

With support by



Оглавление

1	Общее описание	3
1.1	Получение access_token	3
1.2	Права доступа пользователей	3
1.3	Работа с датами	4
1.4	Пагинация	4
2	Список методов и их описание	5
2.1	Общее описание	5
2.2	Таймер	5
2.2.1	timer.get	5
2.3	Пользователи	6
2.3.1	user.get	6
2.3.2	user.get - modify	6
2.3.3	user.list	6
2.3.4	user.list - modify	6
2.3.5	user.delete	7
2.3.6	user.search	7
2.3.7	user.search - modify	7
2.4	Квесты	8
2.4.1	quest.create	8
2.4.2	quest.save	8
2.4.3	quest.delete	9
2.4.4	quest.get	9
2.4.5	quest.delete	9
2.4.6	quest.take	10
2.4.7	quest.take	10
2.4.8	quest.addSection	10
2.4.9	quest.getSection	10
2.4.10	quest.editSection	11
2.4.11	quest.deleteSection	11
2.4.12	quest.listSection	11
2.4.13	quest.addGames	12
2.4.14	quest.removeGames	12
2.5	Аккаунт	12
2.5.1	account.getProfileInfo	12
2.5.2	account.saveProfileInfo	13
2.5.3	account.saveProfileInfo - modify	13

2.5.4	account.changePassword	13
2.5.5	account.changeMail - modify	14
2.5.6	account.changeNick - modify	14
2.5.7	account.changeStyle	14
2.6	Авторизация	14
2.6.1	auth.signup	15

Глава 1

Общее описание

В этой главе будут рассмотрены основные принципы работы с API, а также приняты стандарты обмена вывода некоторых видов информации при работе с API.

1.1 Получение access_token

Для получения access_token необходимо передать на специальный адрес GET запрос вида.

```
https://api.example.com/token/?client_id=web&username=root&password=123456
```

Для получения логина и пароля необходимо зарегистрироваться, используя метод **auth.signup**.

В ответ сервер вернет ответ типа:

```
{
  "access_token": "62947380b6bacd6b6ca49c6b8e90dc3f79dae1f1dd6af695e046453d4550c12b",
  "user_id": "234",
  "expires_in": 1401646393
}
```

В случае неудачи система вернет сообщение с информацией об ошибке.

1.2 Права доступа пользователей

Права выставляются согласно битовой маске. Чтобы получить сразу несколько прав, необходимо сложить их. Ниже приведен список доступных прав

- **user (+1)** - Права, доступные простому пользователю
- **admin (+2)** - Администратор

1.3 Работа с датами

Все временные данные отдаются в UNIX формате.

Исключение составляют поля, где этот формат будет являться излишним (например вывод даты рождения)

1.4 Пагинация

В некоторых методах выводится массив с большим количеством информации. Для удобства вывода информации используется пагинация.

- **count** - Количество возвращаемых элементов (Целое положительное число);
- **offset** - Смещение относительно первого найденного элемента для выборки определенного подмножества (Целое положительное число);

Глава 2

Список методов и их описание

В этой главе будут рассмотрены методы, их описание и результат работы.

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 Таймер

Методы для работы с таймеров (время начала и окончания игры)

2.2.1 timer.get

Получение времени начала и окончания игры

Метод требует access_token

Входные параметры:

- нет

Example

```
https://api.example.com/method/timer.get?access_token={token}
```

2.3 Пользователи

Методы для работы с пользователями

2.3.1 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.

2.3.2 user.get - modify

Думаю что метод должен требовать access_token

2.3.3 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
```

2.3.4 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

2.3.5 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}
```

2.3.6 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}
```

2.3.7 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 инфо по ip адресам)
- **city** - город (собранный инфо по ip адресам)
- **team** - по текущей команде
- **activation** - По статусу активации

2.4 Квесты

Методы для работы с квестами

2.4.1 quest.create

Позволяет добавить квест

Метод требует access_token с правами admin

Входные параметры:

- **title** - Название квеста
- **section** - id категории квеста
- **short_text** - Краткий текст к квесту
- **full_text** - Полный текст к квесту
- **answer** - Ответ на квест
- **score** - Количество очков
- **game** - Игра (в разработке)

Example

```
http://example.com/method/quest.create?title=Cookie&section=2&short_text=testinggg&full_text=this%20is%20just%20string_test_test&answer=12345678&score=300&access_token=1b79bf282670dec5869b9835e598bb5f1910ea0e47bfce4fb099696f8b2cef9d
```

2.4.2 quest.save

Позволяет редактировать квест

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор квеста
- **title** - Название квеста

- **section** - id категории квеста
- **short_text** - Краткий текст к квесту
- **full_text** - Полный текст к квесту
- **answer** - Ответ на квест
- **score** - Количество очков
- **game** - Игра(в разработке)

Example

```
http://93.91.166.30/api/method/quest.save?id=5&title=Cookie&section=2&short_text=testinggg&full_text=this%20is%20just%20string_test_test&answer=12345678&score=300&access_token=1b7
```

2.4.3 quest.delete

Позволяет удалить квест

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор квеста

Example

```
http://example.com/method/quest.delete?id=5&access_token=1b79bf282670
```

2.4.4 quest.get

Возвращает информацию о квесте

Метод требует access_token

Входные параметры:

- **id** - Идентификатор квеста

Example

```
http://example.com/method/quest.get?id=11&access_token=1b79bf282670dec
```

2.4.5 quest.delete

Получить список квестов

Метод требует access_token

Входные параметры:

- Нет

Example

```
http://example.com/api/method/quest.list?access_token=1b79bf28
```

2.4.6 quest.take

Позволяет взять квест обработки

Метод требует access_token

Входные параметры:

- **id** - Идентификатор квеста

Example

```
http://example.com/method/quest.take?id=2&access_token={token}
```

2.4.7 quest.take

Позволяет сдать квест

Метод требует access_token

Входные параметры:

- **id** - Идентификатор квеста
- **answer** - Ответ на квест

Example

```
http://example.com/method/quest.pass?id=2&answer=&access_token={token}
```

2.4.8 quest.addSection

Позволяет добавить тему квеста

Метод требует access_token с правами admin

Входные параметры:

- **title** - Название темы

Example

```
http://example.com/method/quest.addSection?title=test&access_token={token}
```

2.4.9 quest.getSection

Позволяет получить темы квеста

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор темы квеста

Example

```
http://example.com/method/quest.getSection?id=4&access_token={token}
```

2.4.10 quest.editSection

Позволяет редактировать тему квеста

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор темы
- **title** - Название темы

Example

```
http://example.com/method/quest.editSection?id=5&title=ok&access_token={token}
```

2.4.11 quest.deleteSection

Позволяет удалить тему квеста

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор темы квеста

Example

```
http://example.com/method/quest.deleteSection?id=4&access_token={token}
```

2.4.12 quest.listSection

Позволяет получить список квестов

Метод требует access_token с правами admin

Входные параметры:

- Нет

Example

```
http://example.com/method/quest.listSection?access_token={token}
```

2.4.13 quest.addGames

Позволяет прикрепить квест к игре

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор квеста
- **games_id** - Идентификатор игры

Example

2.4.14 quest.removeGames

Позволяет открепить квест от игры

Метод требует access_token с правами admin

Входные параметры:

- **id** - Идентификатор квеста

Example

2.5 Аккаунт

Методы для работы с аккаунтом пользователя

2.5.1 account.getProfileInfo

Получение расширенной информации о своем аккаунте.

Метод требует access_token

Входные параметры:

- нет

Example

https://api.example.com/method/account.getProfileInfo?access_token={token}

2.5.2 account.saveProfileInfo

Позволяет редактировать информацию текущего профиля

Метод требует access_token

Входные параметры:

- **nick** - Новый никнейм пользователя
- **mail** - Новый адрес электронной почты пользователя
(se-kg: по мне так это надо вынести в отдельный метод, так как операция не тривиальная)
- **json_data** - Новая информация пользователя типа “О себе”

Example

2.5.3 account.saveProfileInfo - modify

Позволяет редактировать некоторую информацию текущего профиля

Метод требует access_token

Входные параметры:

- **param** - название параметра
- **value** - значение параметра

Example

http://example.com/method/account.saveProfileInfo?nick=mynewnick&json_data={}&access_

2.5.4 account.changePassword

Смена пароля для пользователя

Метод требует access_token

Входные параметры:

- **oldpass** - Старый пароль пользователя
- **newpass** - Новый пароль пользователя

Example

https://api.example.com/method/account.changePassword?oldpass=123&newpass=345&access_

2.5.5 account.changeMail - modify

Метод требует access_token Позволяет редактировать информацию об электронном ящике

1. Отправить инфу пользователю на мыло о подтверждении смены ящика
2. И если только пользователь подтвердил то производить смену ящика

Мое мнение - это очень опасная функция так как таким образом можно угнать аккаунт. Но функция и полезная вдруг нужно будет сменить мыло. Так что предлагаю эту функцию оставить но только для админа.

Входные параметры:

- **oldmail** - Старый адрес электронной почты пользователя
- **newmail** - Новый адрес электронной почты пользователя
- **pass** - пароль для подтверждения

Example

2.5.6 account.changeNick - modify

Позволяет изменить никнейм у текущего пользователя

Метод требует access_token

Входные параметры:

- **nick** - Новый никнейм пользователя

Example

2.5.7 account.changeStyle

Позволяет изменять стиль отображения

Метод требует access_token

Входные параметры:

- **newstyle** - Новый стиль

Example

2.6 Авторизация

Методы для работы с аккаунтом

2.6.1 auth.signup

Регистрирует нового пользователя

Метод не требует access_token

Входные параметры:

- **mail** - E-mail пользователя
- **nick** - Ник пользователя
- **password** - Пароль пользователя

Example

```
https://api.example.com/method/auth.signup?mail={e-mail}&nick=testnick&password=12345
```