

Все запросы отправляются по HTTP или HTTPS на  
`https://$HOST/api/$API_VERSION/$PATH`

Где PATH - собственно путь для конкретного метода.

HOST - для общих запросов predetermined известный хост-сервер. Для запросов относительно конкретной игры - сервер с данными конкретной игры.

API\_VERSION - версия протокола

Общие параметры, передаваемые через строку запроса:

user\_id - идентификатор пользователя (если требуется авторизация)

sign - подпись, MD5 HMAC всех значений параметров запроса (сначала все GET, потом все POST в порядке задания) (если требуется авторизация)

ts - таймстемп времени отправки запроса в UTC (если требуется авторизация)

## 0. Ошибки

В ответ на любую ошибку приходит объект error:

```
{
  "error": {
    "code": "CODE", // textual code of the error
    "message": "MESSAGE_ID", // identifier of the text message (for localization) (optional)
    "params": [ ], // parameters to localized message (optional)
    "message_en": // preformed message text in English (optional)
  }
}
```

Коды ошибок:

"server\_error" - ошибка сервера

"maintenance" - сервер временно недоступен

"request\_error" - ошибка запроса

"invalid\_api\_version" - неправильная версия api

"not\_supported\_api\_version" - версия api больше не поддерживается

"invalid\_ts" - таймстемп запроса явно неправильный

Другие коды, специфичные для запросов.

## 1. Логин

метод: POST

путь: login

авторизация: нет

SSL

*Параметры запроса:*

а) Логин через Фейсбук

fb\_id - айди фейсбука

db\_access\_token - аксес токен фейсбука

б) Анонимный логин  
[device\_id] - идентификатор девайса

в) Повторный логин  
login\_key - ключ для логина

Ответ:

```
{
  "status": "" // статус логина
  "session": {
    "login_key": "", // ключ для быстрого повторного логина
    "session_key": "" // сессионный ключ для подписи запросов
  }
  "user": {
    "id": "", // идентификатор пользователя в системе
    "fb_id": "", // фейсбук идентификатор (опционально)
    "name": "", // имя пользователя
    "avatar_url": "" // ссылка на аватар (опционально)

    "game": { // текущая игра, если активна
      "host": "", // сервер на котором хранятся данные игры, все запросы относительно
игры, должны идти на него
      "id": "", // идентификатор игры
      "opponent": { // соперник
        "id": "", // идентификатор пользователя в системе
        "fb_id": "", // фейсбук идентификатор (опционально)
        "name": "", // имя пользователя
        "avatar_url": "" // ссылка на аватар (опционально)
      }
    }
  }
}
```

status может иметь одно из значений:

signed\_up - новый пользователь зарегистрирован (данные в объекте user)  
returning\_user - пользователь уже есть и теперь успешно залогинен (данные в объекте user)  
fb\_invalid\_token - был предоставлен неверный токен (также предоставляется объект error с кодом fb\_invalid\_token)

## 2. Получить апдейты

метод: GET (long polling!)  
путь: [games/GAME\_ID/]updates  
авторизация: да

Параметры запроса:

GAME\_ID (через путь) - идентификатор текущей игры

token - строка, идентифицирующая последний апдейт, полученный игроком в игре (может быть не задана)

Ответ:

```
{
  "match": { // если в запросе нет GAME_ID, а есть активная игра
    "game_id": "", // идентификатор игры
    "opponent": { // соперник
      "id": "", // идентификатор пользователя в системе
      "fb_id": "", // фейсбук идентификатор (опционально)
      "name": "", // имя пользователя
      "avatar_url": "" // ссылка на аватар (опционально)
    }
  },

  "declined": { // пользователь отклонил инвайт
    "game_id": "", // идентификатор игры
    "opponent": { // соперник
      "id": "", // идентификатор пользователя в системе
      "fb_id": "", // фейсбук идентификатор (опционально)
      "name": "", // имя пользователя
      "avatar_url": "" // ссылка на аватар (опционально)
    }
  },

  "invites": [ // список инвайтов в игру
    {
      "opponent": { // соперник
        "id": "", // идентификатор пользователя в системе
        "fb_id": "", // фейсбук идентификатор (опционально)
        "name": "", // имя пользователя
        "avatar_url": "" // ссылка на аватар (опционально)
      }
    }
  ]

  "game": { // если есть активная игра
    "status": "" // not_found, out_of_sync, synced, started, playing, configuring или over
    "actions": [ // список данных для новых экшенов в порядке применения (если started,
      playing, configuring или over)
    ],
    "sync": "" // данные синхронизации, (если synced)
    "token": "" // новый токен
  }
}
```

### 3. Инвайт

метод: POST

путь: invite

авторизация: да

Параметры запроса:

opponent\_id - айди оппонента

Ответ:

```
{
  "status": ""
}
```

Статусы:

sent - инвайт отправлен

busy - в данный момент играет в другую игру

not\_found - нет такого пользователя

#### 4. Подтвердить инвайт

метод: POST

путь: invite/OPPONENT\_ID/accept

авторизация: да

Ответ:

```
{
  "status": "",
  "match": { // если статус match
    "game_id": "", // идентификатор игры
    "opponent": { // соперник
      "id": "", // идентификатор пользователя в системе
      "fb_id": "", // фейсбук идентификатор (опционально)
      "name": "", // имя пользователя
      "avatar_url": "" // ссылка на аватар (опционально)
    }
  },
}
```

Статусы:

match - игра началась

already\_playing - отправитель уже играет в другую игру

invalid - инвайт более не валиден

#### 5. Отклонить инвайт

метод: POST

путь: invite/OPPONENT\_ID/decline

авторизация: да

Ответ:

```
{
  "status": ""
}
```

Статусы:

ok

already\_playing - отправитель уже играет в другую игру  
invalid - инвайт более не валиден

## 6. Подписаться на рандом игру

метод: POST  
путь: random/enlist  
авторизация: да

*Ответ:*

```
{  
  "status": ""  
}
```

Статусы:

ok - операция успешно завершена  
already\_playing - отправитель уже играет в другую игру

## 7. Отписаться от рандом игры

метод: POST  
путь: random/remove  
авторизация: да

*Ответ:*

```
{  
  "status": ""  
}
```

Статусы:

ok - операция успешно завершена  
not\_enlisted - пользователь и так не был в списке