Introduction

Glance Clock is the world's first smart clock to display information from all your wearables, smart home devices, web services and third-party applications. We believe that our unique product is the best way to interact with the modern technology.

With Glance Clock you will see only information you need and just when you need it.

Glance Clock integrates with your everyday apps, wearables and services and we would also like to you be a part of our product.

Getting Started

To get started, you need to follow the next steps:

- Register in our developer program. Just send an email to us and we'll enjoy to consider your request.
- Agree to the API Terms of Use.
- That's it!

Authentication

Glance Clock developers API uses OAuth 2.0 and API keys to allow access to the API and authenticate your application under specific user's rights.

After we approve your request described above, you'll receive following info:

- two API keys: Client ID and Client Secret.
- your application's internal name (you'll need to use it as username)
- Glance Clock's API gateway

Access Token

To authorize, use this code:

```
import httplib, urllib, base64

API_GATEWAY = 'https://api.gate.way'
CLIENT_APP_NAME = 'com.yourapp.name'
CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753'
CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f'
CLIENT_KEYS = base64.b64encode(CLIENT_ID + ':' + CLIENT_SECRET)

data = urllib.urlencode({
    'grant_type': 'password',
    'username': CLIENT_APP_NAME,
    'password': 'WDFmQm'
})

headers = {
    "Authorization": "Basic " + CLIENT_KEYS,
    "Content-type": "application/x-www-form-urlencoded"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/oauth/token", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()

<?php

const API_GATEWAY = 'https://api.gate.way';
const CLIENT_APP_NAME = 'com.yourapp.name';
const CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753';
const CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f';

$client_keys = base64_encode(CLIENT_ID . ':' . CLIENT_SECRET);

$data = http_build_query([
    'grant_type' => 'password',
    'username' => CLIENT_APP_NAME,
    'password' => 'WDFmQm'
]);

$headers = 'Authorization: Basic ' . $client_keys . '\r\n' .
           'Content-Type: application/x-www-form-urlencoded';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
```

```
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/oauth/token', false, $context);
```

```
const QS = require('querystring');
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const CLIENT_APP_NAME = 'com.yourapp.name';
const CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753';
const CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f';
const CLIENT_KEYS = new Buffer(CLIENT_ID + ':' + CLIENT_SECRET).toString('base64');

var data = QS.stringify({
    grant_type: 'password',
    username: CLIENT_APP_NAME,
    password: 'WDFmQm'
});

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/oauth/token',
  method: 'POST',
  headers: {
    'Authorization': 'Basic ' + CLIENT_KEYS,
    'Content-Type': 'application/x-www-form-urlencoded'
  }
};

var req = HTTP.request(OPTIONS, function(res) {
  res.setEncoding('utf8');

  var result = '';
  res.on('data', function (chunk) {
    result += chunk;
  });

  res.on('end', function () {
    console.log(result);
  });
});

req.write(data);
req.end();
```

```
curl -X POST \
    -H 'Authorization: Basic YzMwYzAyOGQ2MzYzMjFhMGEzYWEwZjAwZjU5NjgwODgzMmNlMjdmNTY2MmQyNDUwMjY5NzgyNTIwNjNmYTc1MzpmNGI3Y2VmZjlkZTY4ZmI0Njc5MWJhMTJi
    -H 'Content-Type: application/x-www-form-urlencoded' \
   -d 'grant_type=password&username=com.yourapp.name&password=WDFmQm' \
  'https://api.gate.way/v1/oauth/token'
```

Make sure to replace `dummy` values with your own.
The above command returns JSON structured like this:

```
{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
}
```

This info is necessary to get Access Token which is associated with a certain user's account.

To obtain an Access Token you should POST to https://{glanceclock-api-gateway}/v1/oauth/token.

You should include your client credentials in the Authorization header ("Basic " + client_id:client_secret base64'd), and then grant_type ("password"), username (application name) and password in the request body.

HTTP Request

```
POST /v1/oauth/token HTTP/1.1
Host: API_GATEWAY
Authorization: Basic YOUR_API_KEYS_IN_BASE64
Content-Type: application/x-www-form-urlencoded

grant_type=password&username=YOUR_APP_NAME&password=USER_UNIQUE_CODE
```

Attention! It's described below where and how you need to get a USER_UNIQUE_CODE.

The password is a unique temporary code which the user is able to find in one of sections of the Glance Clock's official mobile app.

To authorize your application under some user's account, you need to ask this user to give you the code, and use it as a "password" in authorization request.

Provided there weren't any errors, this will return the following:

HTTP Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

```
{
"access_token":"2YotnFZFEjr1zCsicMWpAA",
"token_type":"bearer",
"expires_in":3600,
"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
`}
```

After that, you can use Access Token to get access to any other API endpoints. Just pass the correct header with each request:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
```

Refresh Token

To authorize, use this code:

```
import httplib, urllib, base64

API_GATEWAY = 'https://api.gate.way'
REFRESH_TOKEN = 'tGzv3JOkF0XG5Qx2TlKWIA'
CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753'
CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f'
CLIENT_KEYS = base64.b64encode(CLIENT_ID + ':' + CLIENT_SECRET)

data = urllib.urlencode({
    'grant_type': 'refresh_token',
    'refresh_token': REFRESH_TOKEN
})

headers = {
    "Authorization": "Basic " + CLIENT_KEYS,
    "Content-type": "application/x-www-form-urlencoded"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/oauth/token", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()
```

```
<?php

const API_GATEWAY = 'https://api.gate.way';
const REFRESH_TOKEN = 'tGzv3JOkF0XG5Qx2TlKWIA';
const CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753';
const CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f';

$client_keys = base64_encode(CLIENT_ID . ':' . CLIENT_SECRET);

$data = http_build_query([
    'grant_type' => 'refresh_token',
    'refresh_token' => REFRESH_TOKEN
]);

$headers = 'Authorization: Basic ' . $client_keys . '\r\n' .
           'Content-Type: application/x-www-form-urlencoded';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/oauth/token', false, $context);
```

```
const QS = require('querystring');
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const REFRESH_TOKEN = 'tGzv3JOkF0XG5Qx2TlKWIA';
const CLIENT_ID = 'c30c028d636321a0a3aa0f00f596808832ce27f5662d245026978252063fa753';
const CLIENT_SECRET = 'f4b7ceff9de68fb46791ba12bdc6d2cc5998c16d59d6e325f685389fd147bf8f';
const CLIENT_KEYS = new Buffer(CLIENT_ID + ':' + CLIENT_SECRET).toString('base64');

var data = QS.stringify({
    grant_type: 'refresh_token',
    refresh_token: REFRESH_TOKEN
});

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/oauth/token',
  method: 'POST',
  headers: {
    'Authorization': 'Basic ' + CLIENT_KEYS,
    'Content-Type': 'application/x-www-form-urlencoded'
  }
};

var req = HTTP.request(OPTIONS, function(res) {
  res.setEncoding('utf8');

  var result = '';
  res.on('data', function (chunk) {
    result += chunk;
  });
```

```
  res.on('end', function () {
    console.log(result);
  });
});

req.write(data);
req.end();
```

```
curl -X POST \
    -H 'Authorization: Basic YzMwYzAyOGQ2MzYzMjFhMGEzYWEwZjAwZjU5NjgwODgzMmNlMjdmNTY2MmQyNDUwMjY5NzgyNTIwNjNmYTc1MzpmNGI3Y2VmZjlkZTY4ZmI0Njc5MWJhMTJi2
    -H 'Content-Type: application/x-www-form-urlencoded' \
   -d 'grant_type=refresh_token&refresh_token=tGzv3JOkF0XG5Qx2TlKWIA' \
  'https://api.gate.way/v1/oauth/token'
```

Make sure to replace `dummy` values with your own.
The above command returns JSON structured like this:

```
{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
}
```

Also, you can use Refresh Token to refresh expired Access Token without is necessary to disturb the user:

HTTP Request

```
POST /v1/oauth/token HTTP/1.1
Host: API_GATEWAY
Authorization: Basic YOUR_API_KEYS_IN_BASE64
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=REFRESH_TOKEN
```

You'll receive a new relevant Access Token.

Drawing primitives

This section describes a high-level patterns to draw your data on Glance Clock. This list of patterns right now is short, but will be extended at development of UI/UX solutions for Glance Clock.

Segments

To draw segments, use this code:

```
import httplib, json

API_GATEWAY = 'https://api.gate.way'
ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA'

data = json.dumps([{
  'start': 1,
  'length': 3,
  'color': '#3384F4',
  'text': 'Hello World'
}])

headers = {
    "Authorization": "Bearer " + ACCESS_TOKEN,
    "Content-type": "application/json"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/draw/segments", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()
```

```
<?php

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

$data = json_encode([[
    'start' => 1,
    'length' => 3,
    'color' => '#3384F4',
    'text' => 'Hello World'
]]);

$headers = 'Authorization: Bearer ' . ACCESS_TOKEN . '\r\n' .
           'Content-Type: application/json';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/draw/segments', false, $context);
```

```
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

var data = JSON.stringify([{
    start: 1,
    length: 3,
    color: '#3384F4',
    text: 'Hello World'
}]);

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/draw/segments',
  method: 'POST',
  headers: {
    'Authorization': 'Bearer ' + ACCESS_TOKEN,
    'Content-Type': 'application/json'
  }
};

var req = HTTP.request(OPTIONS);

req.write(data);
req.end();
```

```
curl -X POST \
    -H 'Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA' \
    -H 'Content-Type: application/json' \
   -d '[{"start":"1","length":"3","color":"#3384F4","text":"Hello World"}]' \
  'https://api.gate.way/v1/draw/segments'
```

Make sure to replace `dummy` values with your own.

This primitive you may see using Google Calendar or Apple Calendar on your Glance Clock.

HTTP Request

```
POST /v1/draw/segments HTTP/1.1
Host: API_GATEWAY
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json

[{
"start": "...",
"length": "...",
"color": "...",
"text": "..."
}, ...]
```

JSON object properties

Parameter	Values	Default	Description
start	1-12	1	Start position of each segment. Can be a floating-point number, not only integer, to display parts of hour.
length	1-12	1	Length of each segment. Can be a floating-point number, not only integer, to display parts of hour.
color	hex	random	Color of each segment.
text	string	“”	Text string.

Important feature in spite of the fact that you can choose any color as parameter, API automatically will convert your color to nearest color in colors palette supported by Glance Clock. This feature is applied to any color parameters of API.

Developer’s API is also automatically will redistribute visual positioning of segments and their thickness depending on crossings of these segments’s “start” and “length” values. But don’t forget that only 4 lines is available.

Here and everywhere below, text string will be scrollable if will needed. We also calculate speed of scrolling regarding text string length.

Don’t forget include Authorization header with Access Code you received before. Here and everywhere below.

Provided there weren’t any errors, this will return the following:

HTTP Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

Here and everywhere below, response from server doesn’t contain any data. That means your content is accepted and will be transferred to the device.

Notice

To draw notice, use this code:

```
import httplib, json

API_GATEWAY = 'https://api.gate.way'
ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA'

data = json.dumps({
```

```
  'color': '#3384F4',
  'text': 'Hello World'
})

headers = {
    "Authorization": "Bearer " + ACCESS_TOKEN,
    "Content-type": "application/json"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/draw/notice", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()
```

```
<?php

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

$data = json_encode([
    'color' => '#3384F4',
    'text' => 'Hello World'
]);

$headers = 'Authorization: Bearer ' . ACCESS_TOKEN . '\r\n' .
           'Content-Type: application/json';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/draw/notice', false, $context);
```

```
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

var data = JSON.stringify({
    color: '#3384F4',
    text: 'Hello World'
});

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/draw/notice',
  method: 'POST',
  headers: {
    'Authorization': 'Bearer ' + ACCESS_TOKEN,
    'Content-Type': 'application/json'
  }
};

var req = HTTP.request(OPTIONS);

req.write(data);
req.end();
```

```
curl -X POST \
    -H 'Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA' \
    -H 'Content-Type: application/json' \
   -d '{"color":"#3384F4","text":"Hello World"}' \
  'https://api.gate.way/v1/draw/notice'
```

Make sure to replace `dummy` values with your own.

This primitive you may see using Weather service on your Glance Clock.

HTTP Request

```
POST /v1/draw/notice HTTP/1.1
Host: API_GATEWAY
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json

{
"color": "...",
"text": "...",
"segments": [...]
}
```

JSON object properties

Parameter	Values	Default	Description
color	hex	random	Background color of whole screen.
text	string	“”	Text string.
segments	array	null	(Optional) Array of objects just like in Segments primitive (see above).

Additional property “segments” allows you to combine full-screen painting of Glance Clock and segmental sectors.

Animation

To draw animation, use this code:

```
import httplib, json

API_GATEWAY = 'https://api.gate.way'
ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA'

data = json.dumps({
  'color': '#3384F4',
  'text': 'Hello World',
  'name': 'wheel'
})

headers = {
    "Authorization": "Bearer " + ACCESS_TOKEN,
    "Content-type": "application/json"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/draw/animation", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()
```

```
<?php

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

$data = json_encode([
    'color' => '#3384F4',
    'text' => 'Hello World',
    'name' => 'wheel'
]);

$headers = 'Authorization: Bearer ' . ACCESS_TOKEN . '\r\n' .
           'Content-Type: application/json';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/draw/animation', false, $context);
```

```
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

var data = JSON.stringify({
    color: '#3384F4',
    text: 'Hello World',
    name: 'wheel'
});

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/draw/animation',
  method: 'POST',
  headers: {
    'Authorization': 'Bearer ' + ACCESS_TOKEN,
    'Content-Type': 'application/json'
  }
};

var req = HTTP.request(OPTIONS);

req.write(data);
req.end();
```

```
curl -X POST \
    -H 'Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA' \
    -H 'Content-Type: application/json' \
   -d '{"color":"#3384F4","text":"Hello World","name":"wheel"}' \
  'https://api.gate.way/v1/draw/animation'
```

Make sure to replace `dummy` values with your own.

This primitive you may see using Uber service (“wheel” animation) on your Glance Clock.

HTTP Request

```
POST /v1/draw/animation HTTP/1.1
Host: API_GATEWAY
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json

{
"color": "...",
"text": "...",
```

```
"name": "..."
}
```

JSON object properties

Parameter	Values	Default	Description
color	hex	random	Color of animation.
text	string	“”	Text string.
name	flower, sun, fan, wheel, star	null	Name of one of pre-defined animations.

Here and everywhere below, frames are being calculated from FPS = 50.
More awesome animations are coming soon!

Custom scene

To draw custom scene, use this code:

```
import httplib, json

API_GATEWAY = 'https://api.gate.way'
ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA'

data = json.dumps([{
   'object': 'FillArea',
   'startTime': 1,
   'lifeTime': 1,
   'arguments': {
       'left': 0,
       'top': 0,
       'width': 20,
       'height': 20,
       'color': '#3384F4'
   }
}, {
   'object': 'ClearAll',
   'startTime': 5,
   'lifeTime': 1,
}])

headers = {
    "Authorization": "Bearer " + ACCESS_TOKEN,
    "Content-type": "application/json"
}

conn = httplib.HTTPConnection(API_GATEWAY)
conn.request("POST", "/v1/draw", data, headers)

response = conn.getresponse()
result = response.read()

conn.close()
```

```
<?php

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

$data = json_encode([[
    'object': 'FillArea',
    'startTime': 1,
    'lifeTime': 1,
    'arguments': {
        'left': 0,
        'top': 0,
        'width': 20,
        'height': 20,
        'color': '#3384F4'
    }
], [
    'object': 'ClearAll',
    'startTime': 5,
    'lifeTime': 1,
]]);

$headers = 'Authorization: Bearer ' . ACCESS_TOKEN . '\r\n' .
           'Content-Type: application/json';

$options = [
    'http' => [
        'method'  => 'POST',
        'header'  => $headers,
        'content' => $data
    ]
];

$context = stream_context_create($options);
$result = file_get_contents(API_GATEWAY . '/v1/draw', false, $context);
```

```
const HTTP = require('http');

const API_GATEWAY = 'https://api.gate.way';
const ACCESS_TOKEN = '2YotnFZFEjr1zCsicMWpAA';

var data = JSON.stringify([{
    object: 'FillArea',
    startTime: 1,
    lifeTime: 1,
    arguments: {
        left: 0,
```

```
        top: 0,
        width: 20,
        height: 20,
        color: '#3384F4'
    }
}, {
    object: 'ClearAll',
    startTime: 5,
    lifeTime: 1,
}]);

const OPTIONS = {
  host: API_GATEWAY,
  path: '/v1/draw',
  method: 'POST',
  headers: {
    'Authorization': 'Bearer ' + ACCESS_TOKEN,
    'Content-Type': 'application/json'
  }
};

var req = HTTP.request(OPTIONS);

req.write(data);
req.end();

curl -X POST \
    -H 'Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA' \
    -H 'Content-Type: application/json' \
   -d '[{"object":"FillArea","startTime":1,"lifeTime":1,"arguments":{"left":0,"top":0,"width":20,"height":20,"color":"#3384F4"}},{"object":"ClearAll",
  'https://api.gate.way/v1/draw'
```

Make sure to replace `dummy` values with your own.

Developer's API is also provides you extended access to drawing on the Glance Clock device.

Using a little bit more complex syntax than pre-defined primitives, you can feel free to draw all what you need to draw.

HTTP Request

```
POST /draw HTTP/1.1
Host: API_GATEWAY
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json

[{
"object": "...",
"startTime": "...",
"lifeTime": "...",
"arguments": {}
}]
```

JSON object properties

Parameter	Values	Default	Description
object	string	none	One of the pre-defined actions described in table below.
arguments	object	null	Additional arguments depending on the object.
startTime	frames	1	(Optional) Frame to start an object.
lifeTime	frames	1	(Optional) Number of frames when object need to be redrawn.

Objects list

Objects	Arguments	Description
ClearAll	none	To clear whole screen.
ClearArea	left (0-47), top (0-3), width (1-48), height (1-4)	To clear exactly area what you need.
FillArea	left (0-47), top (0-3), width (1-48), height (1-4), color (hex)	To fill with color exactly area what you need.
Sound	sound (0-13)	Play pre-defined sound.
Text	text (str), cycles (num)	To set specific text string in text area.
FadeArea	left (0-47), top (0-3), width (1-48), height (1-4), riseTime (frames), fadeTime (frames)	Area are blinking from "fade" to "rise" time in frames (fps=50).

A little bit complicated? Don't worry, it's just a first version of our API and we'll improve it in future. Really.

SDKs

SDKs for server-side apps (NodeJS, PHP, Python) and for mobile apps (iOS, Android) are coming soon.

Terms of Use

This section is under construction.

Errors

This section is still under construction.

Developer's API uses the following error codes:

Error Code	Meaning
400	Bad Request – Your request has bad parameters.
401	Unauthorized – Your Access Token or API keys are wrong.
403	Forbidden – The endpoint requested out of your scope.
404	Not Found – The specified endpoint could not be found.
405	Method Not Allowed – You tried to access to endpoint with an invalid method.
406	Not Acceptable – You requested a format that isn't json.
429	Too Many Requests – You're requesting too many times! Slow down!
500	Internal Server Error – We had a problem with our server. Try again later.
503	Service Unavailable – We're temporarily offline for maintenance. Please try again later.

bash javascript php python