

Frontur doc

Introduction

This API is build with the Open Graph from Facebook in mind. Some similarities are between the two APIs so if you know the Graph API, it will be easy for you to adapt this API.

Every object in the API has a unique ID. You can access the properties of an object by requesting <https://api.bland.is/ID>. For example you can access a thread by going to https://api.bland.is/1_22622620 which results in

```
{
  id: "1_22622620",
  from: {
    name: "zoon",
    id: "80260"
  },
  top_id: "0",
  advtype: 1,
  advertise_type: 2
  headline: "Hvað finnst ykkur um tekju kjör kvenna á fæðing...",
  picture: "",
  created_time: "5.1.2011 09:11:42",
  updated_time: "6.1.2011 16:25:41",
  type: "messageboard",
  source:
"http://barnaland.is/messageboard/messageboard.aspx?advtype=52&advid=22622620",
  answers: "652321",
  eighteen: "False",
  poll_name: ""
}
```

Current available objects in the system are, object are listed below

1. Messageboard
2. Advertise
3. Weblog
4. File

You can access following services

- Messageboard: <https://api.bland.is/messageboard>
- Advertises: <https://api.bland.is/advertise>
- Profile picture: <https://api.bland.is/1/picture>
- Messages: https://api.bland.is/me/message?access_token=...
- Publishing options: https://api.bland.is/me/publishes?access_token=...

Before You Begin

If there are any questions, please email me at ingig@bland.is

Application access

Please request your API key at <http://bland.is/dev/> We prefer that url requests include the API key as a parameter. The name of the parameter is `api_key`. There is no limit on the keys at the moment, this is simply so we can estimate usage, detect accidental abuse and notify you if there are any changes. The `api_key` is only a MUST when doing POST but we would love to have it in all of your GET requests as well.

Example of a GET request

`https://api.bland.is/messageboard/categories?api_key=....`
and
`https://api.bland.is/messageboard?category_id=51&api_key=...`

Authorization

Some services require authorization others are public. To retrieve the `access_token` you send the users username and password as a POST request to <https://api.bland.is/me/login>

```
curl -F "username=[Username]"
     -F "password=[Password]",
     -F "api_key=...."
     https://api.bland.is/me/login
```

This will return

```
{
  "access_token":"XXXXX",
  "confirmed":true|false
}
```

if successful. The Confirmed value indicates if this user has confirmed his username or not. If `confirmed = false` then the user cannot post to the messageboard/advertises

The preferred option of sending the access_token is using the Http Header for both GET and POST request.

You can also send the access_token either in the form/querystring

An example of using the access_token in a querystring is

https://api.bland.is/me/info?access_token=.....

using form in curl

```
curl -F "parent_id=22640648"
      -F "message=this is test"
      -F "access_token=..."
      -F "api_key=..."
      https://api.bland.is/me/messageboard
```

Register

You can register user using a POST request to <https://api.bland.is/me/register>

```
curl -F "username=[Username]"
      -F "password=[Password]",
      -F "email=[Email]",
      -F "api_key=...."
      https://api.bland.is/me/register
```

This will return

```
{
  "success":true,
  "message": "You'll need to confirm your username before you continue",
  "access_token": "XXX",
  "confirmed": true|false
}
```

Basic info about user

To retrieve the avatar photo from a user you use the id of the user in the following url

https://api.bland.is/{user_id}/picture

for instance, the user id = 1, use the following url

<https://api.bland.is/1/picture>

Getting the basic information about the user that is logged in, the info returned is username, profile image, karma points, count of unread messages and notifications

To retrieve this information, go to the url below and send in the access_token
https://api.bland.is/me/info?access_token=.....

This will return

```
{
  username: "ingig",
  image: "https://bland.is/album/avatar/1/avatar_tiny.jpg?d=1756129700",
  karma: 2005,
  new_messages_count: 2,
  new_notifications_count: 0
}
```

Reading

Paging

When querying News feed you can ask for limited information or more info using the limit and offset parameters

- limit, offset: <https://api.bland.is/me/message?limit=30&offset=1>

This request will return 30 results from records between 30-60. offset 0 will return records between 0-30.

Publishing

You can publish to your website by issuing HTTP POST request to the appropriate URL.

Publish to

Users private publishing options are at https://api.bland.is/me/publishes?access_token=...

Weblog

For example to post on a weblog you sent a POST request to <https://api.bland.is/me/feed>

```
curl -F "access_token=..."
      -F "headline=This is headline text",
```

```
-F "message=Hello new world API"
-F "page_id=123"
-F "type=weblog"
-F "api_key=...."
-F "file_id=123" //optional
-F "file_position=top|bottom" //optional
https://api.bland.is/me/feed
```

Method	Description	Required arguments
/me/feed	Create a new weblog post on a specific website	access_token, message OR file_id, page_id, type

It's required to send either a message or file_id, this way a user can post just a file(e.g. picture, video or audio) and doesn't have to write any text

If you want to publish a photo/video/audio/file with the weblog you can send in the file_id, so if you are going to use a photo that you just captured you need to upload it first using <https://api.bland.is/me/file> (see below). There you'll get the file_id. By default the file is inserted at the top of the message inside `<div class="mobileUpload">`, if you want to position it at the bottom you can send in the file_position=bottom parameter

if the file is a picture, it's inserted using `` tag

if the file is a video, it's inserted using `<video src="" />` tag

if the file is a audio, it's inserted using `<audio src="" />` tag

Storage place

Each file is kept in a Place (also know as albums on the website). To retrieve available places https://api.bland.is/me/places?access_token=...

Files

To retrieve files from a Place you request

http://graph.barnalandid.is/me/files?place_id=123&access_token=...

File

You can upload your files by sending a POST request to <https://api.bland.is/me/file>

```
curl -F "access_token=..."
-F "text=Hello new world API"
-F "file=@c:\file.jpg"
-F "api_key=...."
https://api.bland.is/me/file
```

or using base64 encoding

```
curl -F "access_token=..."
-F "text=Hello new world API"
-F "file=image/jpeg;base64,/9j/4S0GRXhpZgAATU0AKgAAAAGADQEO....."
-F "api_key=..."
https://api.bland.is/me/file
```

Method	Description	Required arguments
/me/file/	Upload a file, this can be any type of file	access_token, file

Response

```
{
  "file_id":7989237,
  "name":"20110224124242_0.jpg",
  "valid":true
}
```

Objects

Messageboard

You can get a list of available messageboard threads by requesting <https://api.bland.is/messageboard>

You can retrieve the list of available messageboards categories at the URL <https://api.bland.is/messageboard/categories>

Parameters

Name	Description
category_id	The category that you want the thread, you can get a list of categories at https://api.bland.is/messageboard/categories . Default is 52

sort	The way you want to sort the list. 1 = Latest update 2 = Latest created Default is 1
offset	What is the offset you want, the result always returns 35 items, so if you want item 36-70 the offset should be 1. Default is 0
access_token	Not required but if you want to be able to see threads that are marked eighteen you need to send in the token

You can retrieve a list of thread by requesting

https://api.bland.is/messageboard?category_id=51&sort=2&offset=1&access_token=..

This will return a list of items from 36-70 sorted by when it was created in the category Barnið

Properties

Name	Description
id	The thread id
from	The user that created the thread
category_id	The category id of this thread
advertise_type	For sale, requesting, for free, for rent
top_id	References to the first id in the thread
headline	Headline of the thread
text	The text in the thread
picture	If there is a picture with the thread this is a url to it
created_time	The time that this item was created
updated_time	The time the thread was last updated, this is only available in the first item

type	This is always "messageboard"
source	Link to this thread on the website
eighteen	Is true if this thread is restricted to payed users otherwise false
view_count	How many view this thread has been viewed
level	<p>The level of this item, this level indicates where in the tree it is. The first item is always level=0, the first answer is level=1. There is no limit on the level</p> <pre> 1 --- 2 ----- 3 ----- 3 --- 2 --- 2 ----- 3 </pre>
poll_name	The name of the poll that comes with the thread if there is one. This is only available in level=0
answers	Array of this object with each answer. This is only available in level=0

To retrieve a individual thread you request by it's id.

https://api.bland.is/1_22622620?access_token=...

If this is a thread marked as an eighteen thread, you need to send in the access_token.

Publishing

Create new thread. To create new advertise thread send these parameters

```

curl -F "message=this is test"
      -F "headline=test headline"
      -F "category_id=21"
      -F "advertise_type=1" //for sale
      -F "access_token=..."
      -F "source=@file.jpg" //if pictures are allowed
      -F "api_key=..."
https://api.bland.is/me/messageboard

```

Answer to a thread

```

curl -F "parent_id=22640648"
      -F "message=this is test"

```



```
-F "access_token=..."
-F "api_key=..."
https://api.bland.is/me/messageboard
```

Bought & Sold products (not available yet)

You can retrieve the list of bought and sold products from these URLs, this will return the last 50 bought/sold products

List of bought products

https://api.bland.is/me/boughtproducts?access_token=...

List of sold products

https://api.bland.is/me/soldproducts?access_token=...

This will return this results if any

[

```
{
  ReviewId: 164,
  ClassifiedId: 2885,
  ExchangePid: 43054,
  Created: "2013-02-19T17:20:58.903",
  Text: null,
  Stars: null,
  SellersText: null,
  Amount: 1500000,
  Address: "Njálsgata, 105 Reykjavík",
  AddressNr: "1229",
  Longitude: "-21.926798545507836",
  Latitude: "64.14401605899457",
  DeliveryType: 1,
  Sent: false,
  Recieved: false,
  OtherInfo: "Bjalla 55",
  SellerPid: 1,
  Headline: "Nýskoðaður Opel Astra",
  BuyerUsername: "maggi",
  Canceled: false,
  ModifiedDate: null,
  Phonenumber: "6995335",
  Deleted: false,
```

```
        DeletedDate: null,  
        DeletedIP: null  
    },  
]
```

Mark product as recieved

To mark product as recieved you send POST to this URL

```
curl -F "advertiseReviewId=123"  
-F "access_token=..."  
https://api.bland.is/me/productrecieved
```

Result

```
{  
    "success":true,  
    "sent":true,  
    "reviewId":123  
}
```

Mark product as sent

To mark product as sent you send POST to this URL

```
curl -F "reviewId=123"  
-F "access_token=..."  
https://api.bland.is/me/productsent
```

Result

```
{  
    "success":true,  
    "recieved":true,  
    "reviewId":123  
}
```

Auto Update Advertise

When you buy an ad in our system you have the option of allowing the ad to update it self from a external source. This can be nice when you have for example changing offers every day, this saves time so you don't have to log into our system every time to change the ad.

When creating the ad you need to send in a picture, add an url and text. This information will be used if the auto update fails or if it's returned empty.

The external source will be requested every 10 minutes so each update can take at least 10 minutes. If you need to force the update you edit the ad and click the Next button at the bottom

The json scheme to update advertise automatically is as followed, all fields are optional

```
{
  "text": "Ad text, we only display the first 80 characters",
  "url": "The url where the user should end up at when clicking the ad,
example http://example.com",
  "file": "Url to the file that should be displayed, this can be jpg, png,
gif or swf file, example http://example.com/ad.jpg",
  "likeLink": "Facebook like link, example http://example.com"
}
```

If the auto update fails, you will be emailed about the error with the description of the error so you'll have the opportunity to fix it.