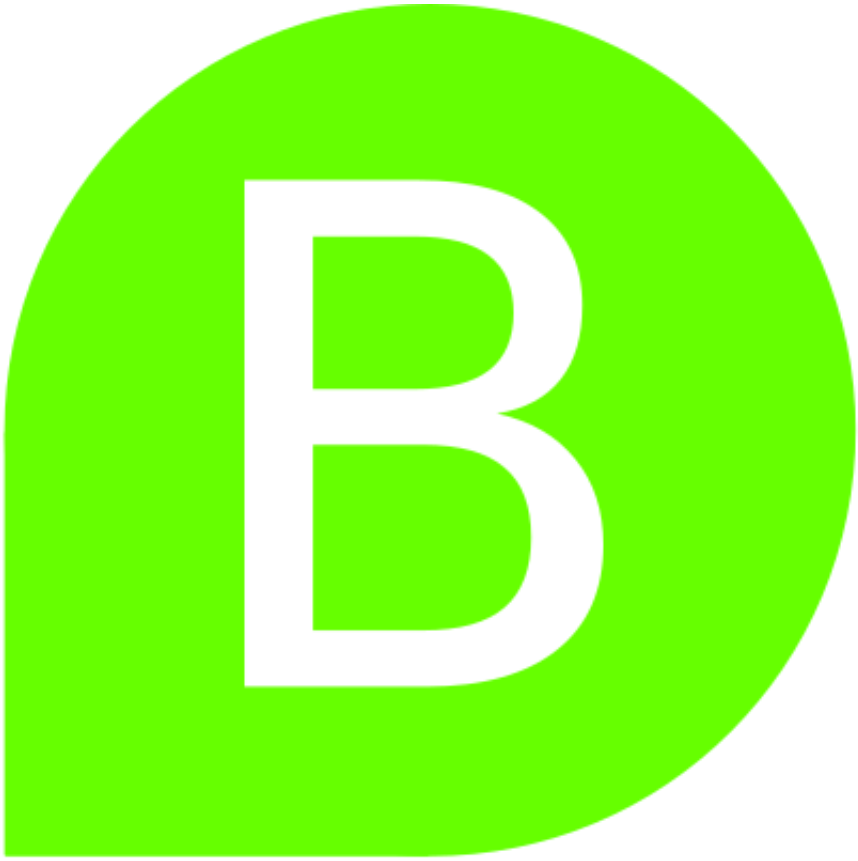


BARKER

User Manual

Contents

Barker

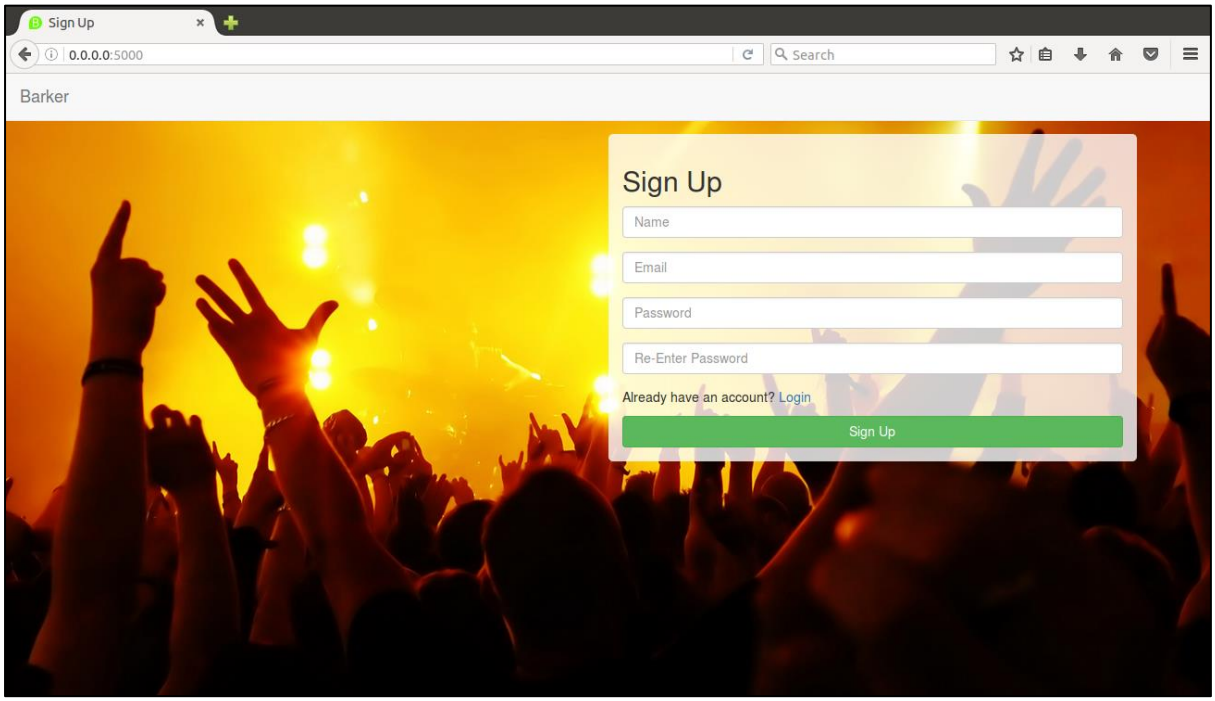
Barker is a simple social network modelled after Twitter, from which it took its name. The design was heavily influenced after other social networks to give so that it would feel familiar to the user. Its layout was designed for a range of different screen sizes. Although it is not as feature rich as similar web apps, it is designed to be easy to use.

Online users

Barker can be found at g00324844-data-rep-project.herokuapp.com and consists of five pages.

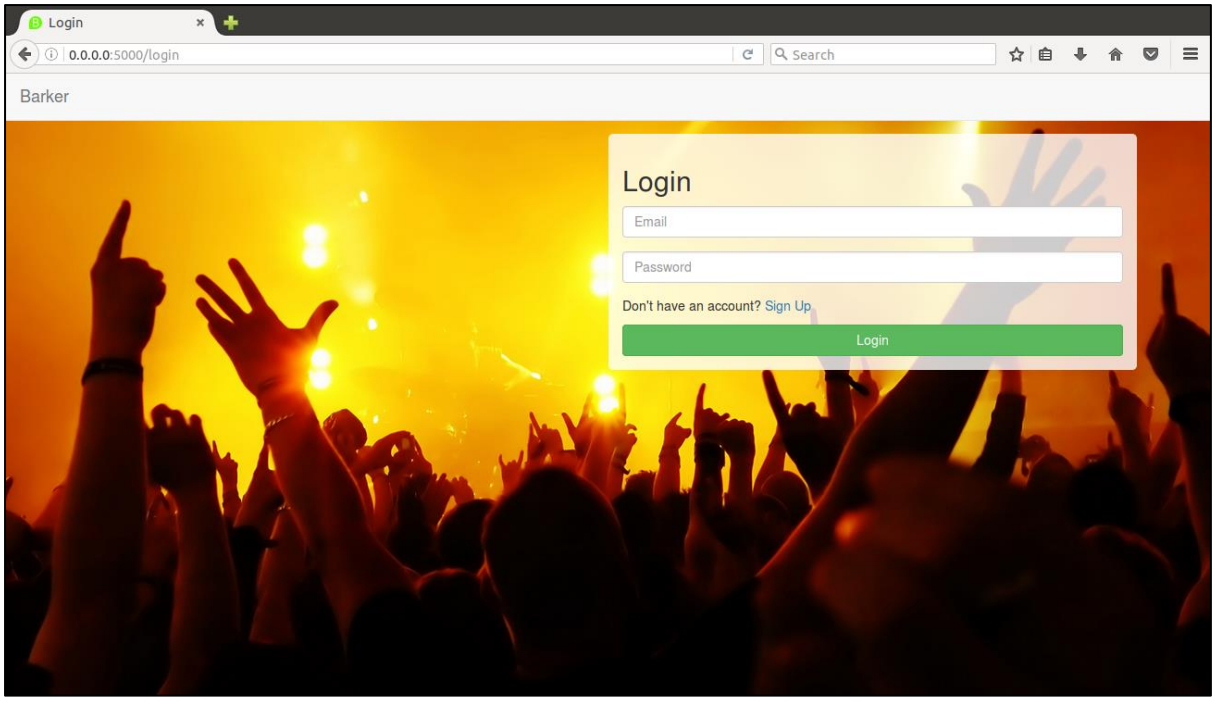
Sign Up

You can sign up using your email address and password. The email address will be used in the future for resetting passwords. Once the user successfully signs up a random username will be generated for them and they will be redirected to the feed page. From here the user will also be able to get to the login page.



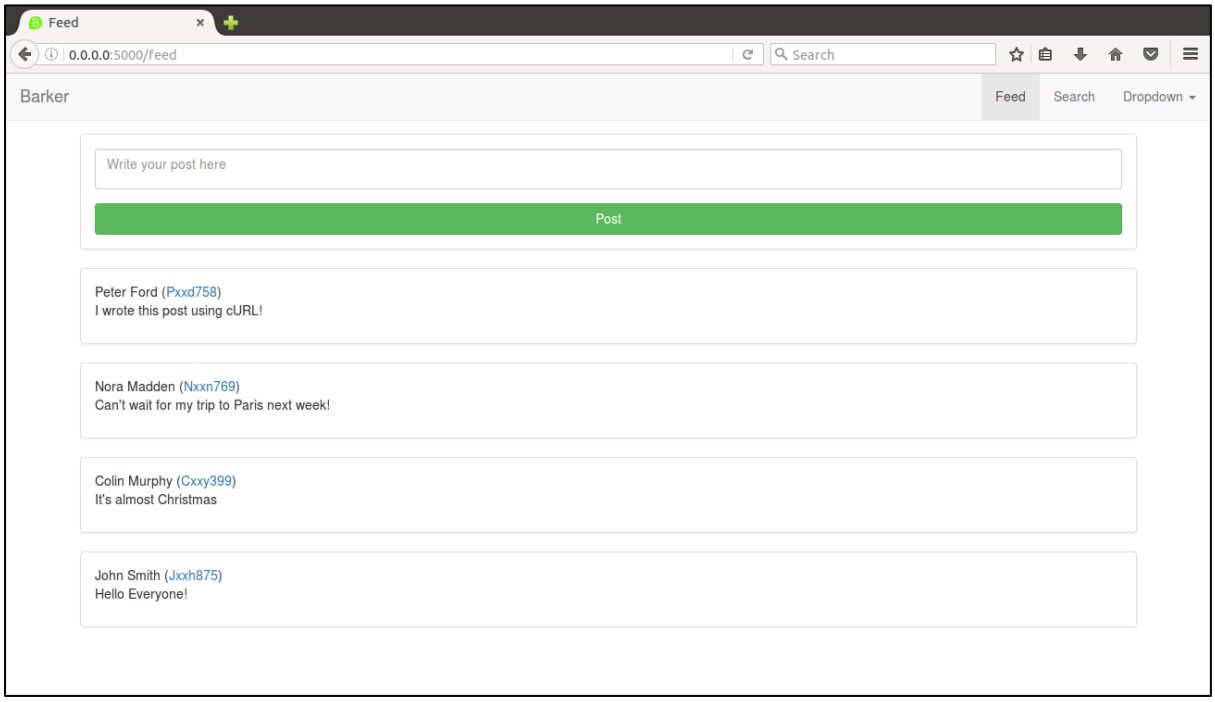
Login

From here you can login using your email address and password, or they can go back to the signup page. Once you log in correctly you will be redirected to the feed page. If your login attempt failed, an error message will appear to inform you.



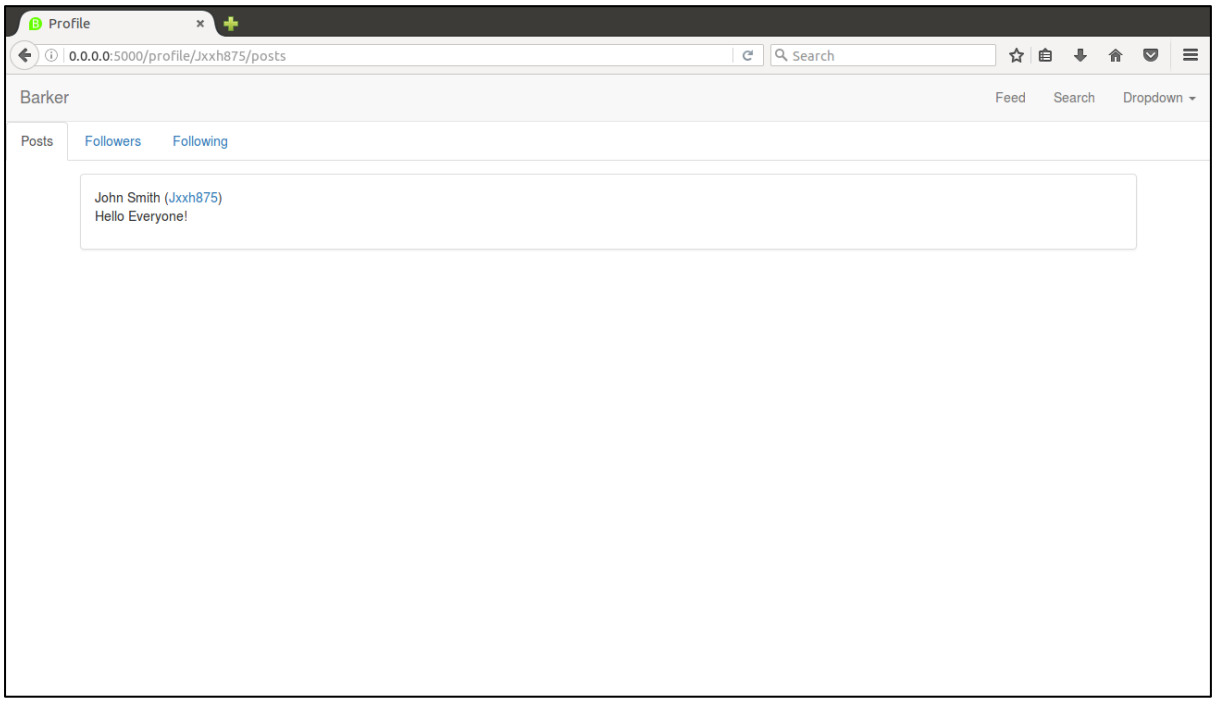
Feed

The feed page is where the users feed will be displayed. The feed consists of posts written both by the user and the users they follow. The user can also write new posts here. When a post is written the feed will be updated with all the newest posts as well as your post. If new posts were written since the user loaded the page, a button will appear at the top asking if they want these posts to be shown. If the user scrolls to the end of the page then older posts will be loaded. A post is simply a message with the name and username of the author at the top. You can click the username to see the users' profile.



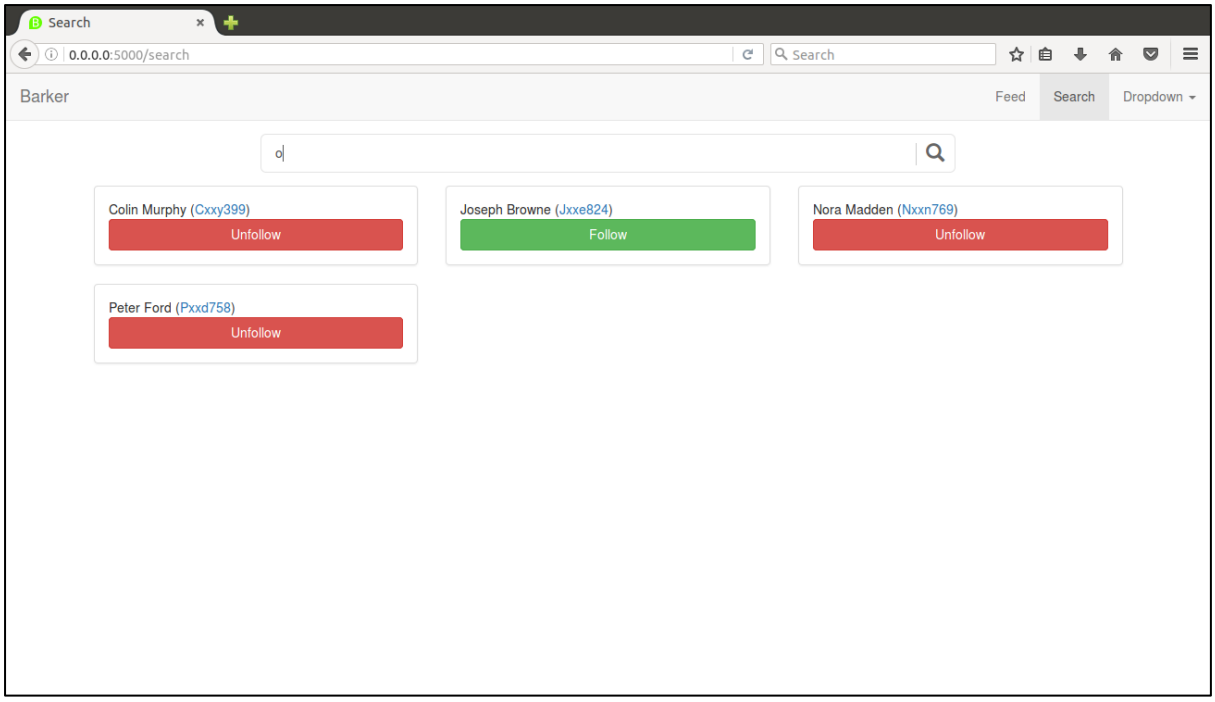
Profile

The profile page consists of three views. They are posts, followers and following. Posts displays your twenty most recent posts. Followers displays all the people that follow you and gives you an option to follow or unfollow them. They're username will also be displayed and you can click it to view their profile. The following view is similar to followers except it displays the people that you follow.



Search

The search view can be found in the menu at the top of the page. It searches for users by comparing their name to the search query. If no user was found a message will be displayed to inform you. The format of the search results will be similar to that of the followers and following views in on the profile page where you will be given an option to follow, unfollow or view the their profile.



cURL

Barker's API was designed to follow RESTful principles and is therefore useable from the command line using cURL. There are a number of different resources available as shown below.

Register a new user

To create a new user account using use the following command. It is a post request with your name, email address and password in JSON format. This is the only command you can use without authentication.

```
curl -X POST http://0.0.0.0:5000/register -H 'Content-Type:
application/json' -d '{"name": "Peter Ford", "email":
"peter.ford@gmit.ie", "password": "PeterFord"}'
```

RESPONSE:

```
{"user": {"email": "peter.ford@gmit.ie", "username": "Pxxd937"},
"status": "success", "message": "User successfully registered."}
```

Search for a user

To search for a user by their name use a GET request and passing in your query in the URL. The following HTTP will return all the users with "o" in their name. The query is case insensitive. Also note that you most have authentication to use this and subsequent commands as mentioned earlier.

```
curl -u peter.ford@gmit.ie:PeterFord
http://0.0.0.0:5000/search_users?query=o
```

RESPONSE:

```
{"message": "Successfully searched for user.", "status": "success",
"users": [{"email": "john.smith@gmit.ie", "following": false,
"username": "Jxxh875", "name": "John Smith"}, {"email":
"colin.murphy@gmit.ie", "following": false, "username": "Cxxy399",
"name": "Colin Murphy"}, {"email": "joseph.browne@gmit.ie",
"following": false, "username": "Jxxe824", "name": "Joseph Browne"},
{"email": "nora.madden@gmit.ie", "following": false, "username":
"Nxxn769", "name": "Nora Madden"}]}
```


Follow a user

To follow a user you must first get their email address and then send it in JSON format, using a post request, to /following.

```
curl -u peter.ford@gmit.ie:PeterFord -X POST
http://0.0.0.0:5000/following -H 'Content-Type: application/json' -d
'{"email": "joseph.browne@gmit.ie"}'
```

RESPONSE:


```
{"message": "Successfully followed user.", "status": "success"}
```


Get a user's following

To retrieve a list of users that follow a particular user follows use a get request to /following and pass the username of the that user as a key value pair in the URL.

```
curl -u peter.ford@gmit.ie:PeterFord
http://0.0.0.0:5000/following?username=Pxxd758
```

RESPONSE:


```
{"message": "Successfully retrieved the users this user is
following.", "status": "success", "users": [{"email":
"joseph.browne@gmit.ie", "following": true, "username": "Jxxe824",
"name": "Joseph Browne"}]}
```


Unfollow user

To unfollow a user use a delete request to following and pass in the uses email address.

```
curl -u peter.ford@gmit.ie:PeterFord -X DELETE
http://0.0.0.0:5000/following?email=joseph.browne@gmit.ie
```

RESPONSE:


```
{"message": "Successfully unfollowed user.", "status": "success"}
```


Get followers

This is similar to the following command except you send the request to /followers.

```
curl -u peter.ford@gmit.ie:PeterFord
http://0.0.0.0:5000/followers?username=Pxxd758
```

RESPONSE:

```
{"message": "Successfully retrieved this users followers.",
"status": "success", "users": [{"email": "john.smith@gmit.ie",
"following": false, "username": "Jxxh875", "name": "John Smith"}]}
```


Write a new post

To write a new post use the post command and send your message in a JSON string with a key called ‘text’ as follows

```
curl -u peter.ford@gmit.ie:PeterFord -X POST
http://0.0.0.0:5000/posts -H 'Content-Type: application/json' -d
'{"text": "I wrote this post using cURL!"}'
```

RESPONSE:


```
{"status": "success", "message": "Successfully added post."}
```


Get feed

To get your feed use the following command. Notice the two parameters. Timestamp determines the oldest post to be retrieved. If you want to retrieve any post regardless of when it was written set timestamp to zero. The second parameter is skip. This determines the number of posts to be skipped and is used for pagination. The posts are returned in descending order of their timestamp.

```
curl -u peter.ford@gmit.ie:PeterFord
"http://0.0.0.0:5000/posts?timestamp=0&skip=0"
```

RESPONSE:


```
{"message": "Posts retrieved successfully.", "status": "success",
"posts": [{"timestamp": 1480599626.455929, "username": "Pxxd937",
"name": "Peter Ford", "message": "I wrote this post using cURL!"}]}
```


Get users posts

To retrieve the most recent 20 posts written by a particular user use the following command where you pass the username of the user as part of the URL.

```
curl -u peter.ford@gmit.ie:PeterFord
http://0.0.0.0:5000/Pxxd758/get_users_posts
```

RESPONSE:


```
{"message": "Posts retrieved successfully.", "posts": [{"message":
"I wrote this post using cURL!", "name": "Peter Ford", "username":
"Pxxd758", "timestamp": 1480620094.368788}], "status": "success"}
```