

RubyでTwitterBotを作ろう!

2EP3-21 栗山 純平



もくじ

0. あらすじ、説明色々

1. Twitter側の設定

2. Rubyでコーディング!

3. Botの機能を増やしてみる

4. まとめ

TwitterBotって？

“自動で”つぶやくアカウント

ツイート
11,364

フォロー
1

フォロワー
155,975

飯元光輝〜

1

1

16

5%の確率で性器を露出するドラ●●ん @5percent_Dora

1

1

15

5%の確率で性器を露出するドラ●●ん @5percent_Dora

3

17

5%の確率で性器を露出するドラ●●ん @5percent_Dora

4

19

5%の確率で性器を露出するドラ●●ん @5percent_Dora

7

16

5%の確率で性器を露出するドラ●●ん @5percent_Dora

5

12

5%の確率で性器を露出するドラ●●ん @5percent_Dora

7

5

19

5%の確率で性器を露出するドラ●●ん @5percent_Dora · 6月15日

3

1

13

5%の確率で性器を露出するドラ●●ん @5percent_Dora · 6月15日

ツイート
207,221

フォロー
242,803

フォロワー
323,495

ツイート

ツイートと返信

しゅうまい君 @shuumai · 11 分

当たり前の三角関係っていうオタクもびっくり設定だとか言ってくれるんや...

4

21

94

しゅうまい君 @shuumai · 31 分

つぶらなこの出荷野郎...ってなってる私.....

10

48

197

しゅうまい君 @shuumai · 51 分

防犯カメラw

14

23

199

しゅうまい君 @shuumai · 1 時間

じわじわきて自重し本当になってる時が一番微妙とい

5

12

69

しゅうまい君 @shuumai · 2 時間

若く見えた

14

23

154

しゅうまい君 @shuumai · 2 時間

黒石がっけーって思ってた常人なりの含め円盤でいとのぼくからしたらコメントめっちゃ入ってるよ

8

82

121

しゅうまい君 @shuumai · 2 時間

小さい頃よく横断歩道の指消しとばさない間に帰れ〜)

7

39

142

ツイート
44,354

フォロー
5

フォロワー
2,740,212

いいね
1

リスト
4

地震速報 @earthquake_jp · 2 時間

【気象庁情報】26日 22時36分頃 静岡県西部 (N34.7/E137.8) にて 最大震度1 (M3.4) の地震が発生。 震源の深さは30km。 (j.mp/2IFnq2C) #saigai #jishin #earthquake

9

419

278

地震速報 @earthquake_jp · 5 時間

【気象庁情報】26日 20時03分頃 岩手県沖 (N39.3/E142.1) にて 最大震度3 (M4.3) の地震が発生。 震源の深さは50km。 (j.mp/2t3OrE5) #saigai #jishin #earthquake

1

227

149

地震速報 @earthquake_jp · 5 時間

【気象庁速報】26日 20時03分頃 岩手県沿岸南部にて 最大震度3の地震が発生。 (j.mp/2Kmsbs3) #saigai #jishin #earthquake

10

374

180

地震速報 @earthquake_jp · 5 時間

【速報LV1】26日 20時03分頃 岩手県沖 (N39.3/E142.1) (推定)にて M4.5 (推定) の地震が発生。 震源の深さは推定51.7km。 #saigai #jishin #earthquake

341

162

地震速報 @earthquake_jp · 5 時間

【気象庁情報】26日 19時46分頃 千葉県南部 (N35.3/E140.4) にて 最大震度4 (M4.4) の地震が発生。 震源の深さは30km。 (j.mp/2yFSi92) #saigai #jishin #earthquake

2

592

335

地震速報 @earthquake_jp · 5 時間

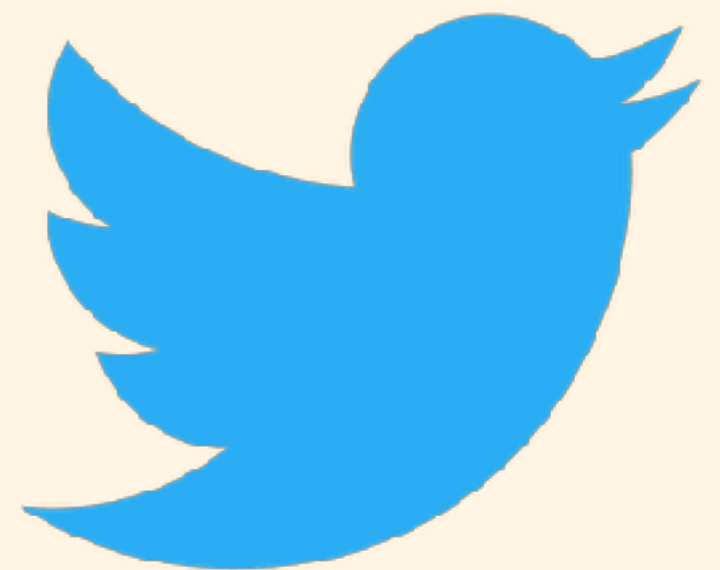
【気象庁速報】26日 19時46分頃 千葉県北東部にて 最大震度4の地震が発生。 (j.mp/2tFeg5Z) #saigai #jishin #earthquake

14

1,128

521

1. Twitter側の登録



1-1.アカウントを用意



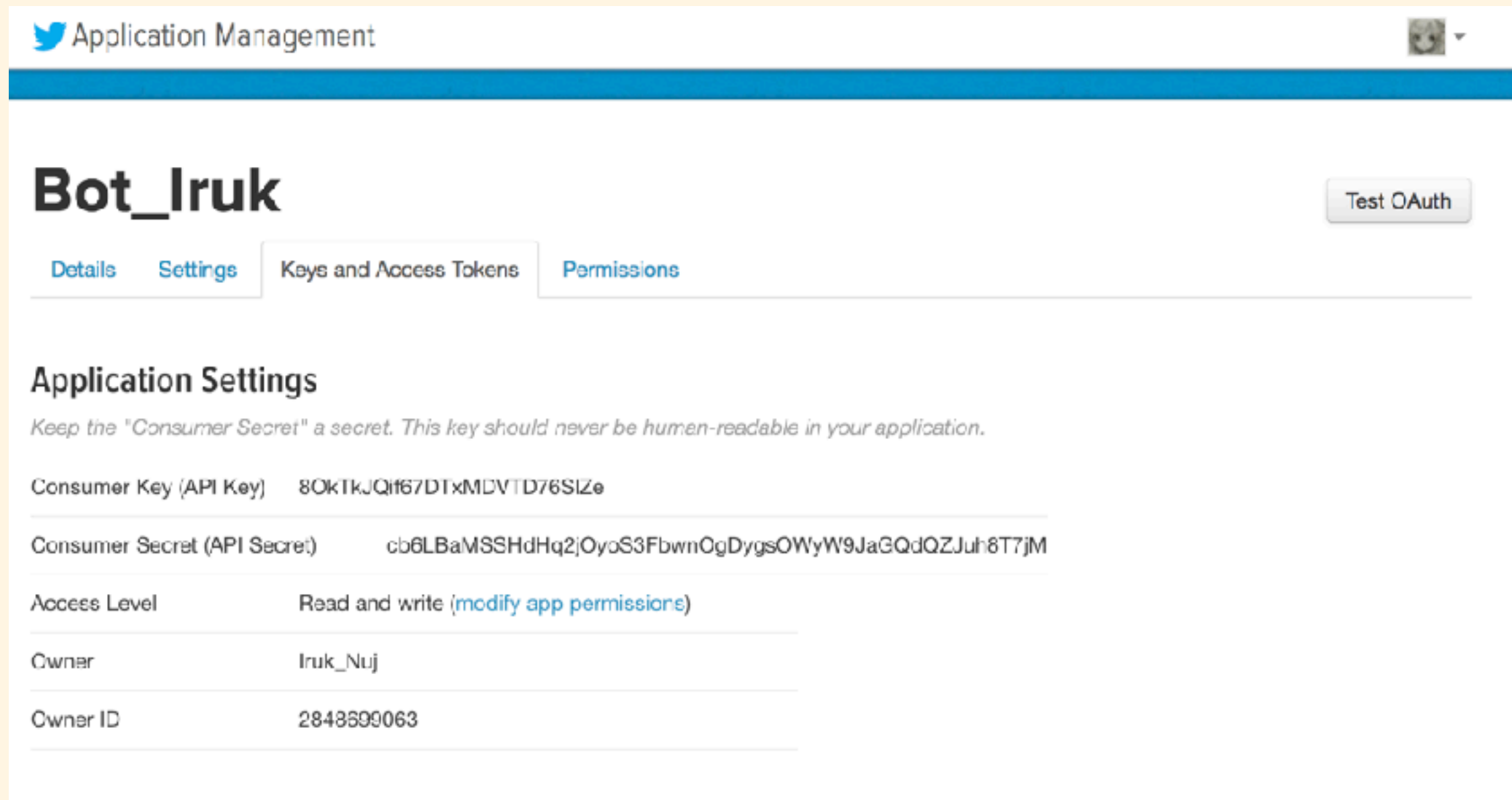
1-2. アプリ登録

- <https://apps.twitter.com/>



1-3. アクセストークン諸々の取得

- 同サイト



The screenshot shows the Twitter Application Management interface for an application named 'Bot_Iruk'. The page has a blue header with the Twitter logo and 'Application Management' text. Below the header, the application name 'Bot_Iruk' is displayed in large bold text, with a 'Test OAuth' button to its right. A navigation bar contains four tabs: 'Details', 'Settings', 'Keys and Access Tokens', and 'Permissions'. The 'Settings' tab is currently selected. Under the 'Application Settings' section, there is a warning: 'Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.' Below this, several settings are listed in a table-like format:

Consumer Key (API Key)	8OKtKJQif67DTxMDVTD76SIZe
Consumer Secret (API Secret)	cb6LBaMSSHdHq2jOyoS3FbwnOgDygsOWyW9JaGQdQZJuh8T7jM
Access Level	Read and write (modify app permissions)
Owner	Iruk_Nuj
Owner ID	2848699063

2.Rubyでコーディング!

```

1 require 'twilio'
2 require 'net'
3 require 'socket'
4 require_relative 'config.rb'
5
6 USERNAME = "BorgBot"
7
8 class Server
9
10   def initialize
11     @client = Twilio::REST::Client.new do |config|
12       config.consumer_key = ENV['CONSUMER_KEY']
13       config.consumer_secret = ENV['CONSUMER_SECRET']
14       config.access_token = ENV['ACCESS_TOKEN']
15       config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']
16     end
17
18     @stream_client = Twilio::StreamingClient.new do |config|
19       config.consumer_key = ENV['CONSUMER_KEY']
20       config.consumer_secret = ENV['CONSUMER_SECRET']
21       config.access_token = ENV['ACCESS_TOKEN']
22       config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']
23     end
24
25     @stream_client
26
27     if (rand(1..100) < 5)
28       tweets = []
29       File.open('tweets/tweets.txt') do |tweet_file|
30         tweet_file.each_line do |tweet_line|
31           tweets.push(tweet_line)
32         end
33       end
34       @client.update_tweets(tweets, sample: 1)
35     end
36
37     def stream_reply
38       @stream_client.user do |tweet|
39         if tweet.is_a?(Twilio::Tweet)
40           if !tweet.retweeted_status
41             random_timestamp(tweet)
42             reply_string(tweet)
43             reply_subheader(tweet)
44             reply_body(tweet)
45             reply_footer(tweet)
46             time_text(tweet)
47           end
48         end
49       end
50     end
51
52     end
53
54   end
55 end

```



2-1.作業用フォルダの作成

```
$ mkdir bot_test
```

```
$ cd bot_test
```

```
$ touch twitter_bot.rb
```

2-2. 'twitter' gemの導入

```
$ touch Gemfile
```

```
$ vim Gemfile
```

```
-----  
Source 'https://rubygems.org'
```

```
ruby '適当なバージョン'
```

```
gem 'twitter', '5.16.0'
```

```
-----  
$ bundle install
```

2-3.Botのコーディング

```
$ vim twitter_bot.rb
```

```
require 'twitter'
```

```
class Tweet
```

```
  def initialize
```

```
    @client = Twitter::REST::Client.new do |config|  
      config.consumer_key = ENV['CONSUMER_KEY']  
      config.consumer_secret = ENV['CONSUMER_SECRET']  
      config.access_token = ENV['ACCESS_TOKEN']  
      config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']  
    end
```

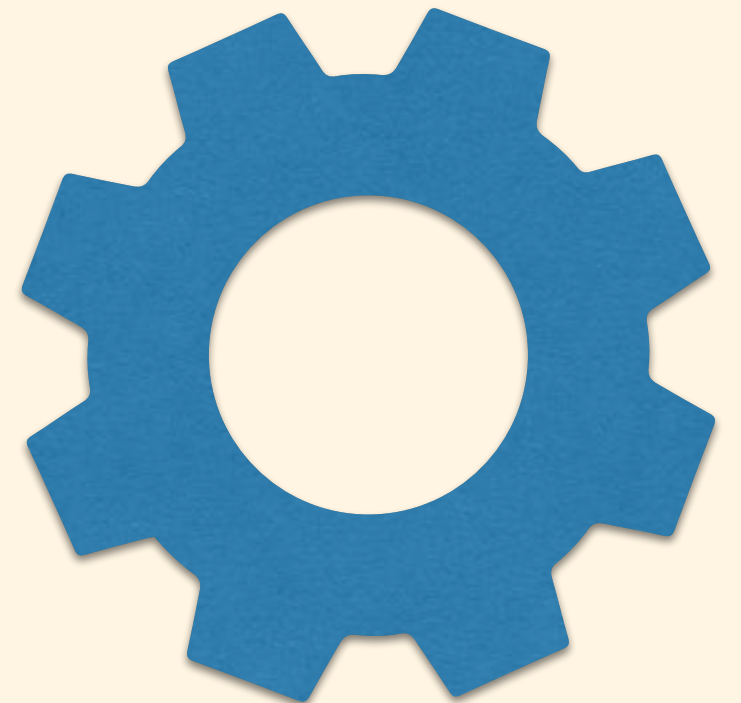
```
    @stream_client = Twitter::Streaming::Client.new do |config|  
      config.consumer_key = ENV['CONSUMER_KEY']  
      config.consumer_secret = ENV['CONSUMER_SECRET']  
      config.access_token = ENV['ACCESS_TOKEN']  
      config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']  
    end
```

```
  end  
end
```

```
  @client.update("Hello TwitterBot!")  
end
```

2-4.実行してみよう！

3.Botの機能を増やしてみる



3-1.確率で呟くように設定

```
$ vim twitter_bot.rb
```

```
•  
•
```

```
if (rand(1..100) < 6)  
  @client.update("ボロン")  
else  
  @client.update("Hello Twitter_Bot!")  
end
```



3-2.返信機能に対応

@client

```
@client = Twitter::REST::Client.new do |config|  
  config.consumer_key = ENV['CONSUMER_KEY']  
  config.consumer_secret = ENV['CONSUMER_SECRET']  
  config.access_token = ENV['ACCESS_TOKEN']  
  config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']  
end
```

呟き周りの権限を取得

3-2.返信機能に対応

@stream_client

```
@stream_client = Twitter::Streaming::Client.new do |config|  
  config.consumer_key = ENV['CONSUMER_KEY']  
  config.consumer_secret = ENV['CONSUMER_SECRET']  
  config.access_token = ENV['ACCESS_TOKEN']  
  config.access_token_secret = ENV['ACCESS_TOKEN_SECRET']  
end
```

タイムライン読み取りの権限を取得

→これらを組み合わせて返信機能を作成！

```
$ vim twitter_bot
```

```
@stream_client.user do |tweet|  
  if tweet.text.include?("おやすみ") && (tweet.user.screen_name != USERNAME)  
    @client.update("おやすみなさい。", options = {:in_reply_to_status_id =>  
tweet.id})  
  end  
end
```

やってること

- @stream_client.user do || でTLの取得
- if文で呟き内に目的の語句があるかの判定
- さらに呟きが自分のものでないかの判定(自分に返信して無限ループしてしまうから)
- 条件文が通ったら@client.updateでツイート
- 後ろのoptionはリプライ時に書くそういうもの

3-3.実際に動かしてみよう！

4.まとめ

4-1.まとめ

```
$ ruby tweet_bot.rb
```

-
- ここまでの機能で簡単なTwitterBotが作れる！
 - Botに実行させるコーディングは貴方次第！
 - 好きなものを作ってプログラミングを楽しんでいこう！

4-2. ...あれ？

呟く機能は作成できたけど

呟く度にCL起動してたら普通のツイートと変わらないんじゃないじゃ...？

手を動かさずにツイートしたい！

次回は
Paasにあっぷろーどして
常時起動させてみよう！



- Heroku
- Heroku Scheduler
- Sinatra

辺りについて話すよ

