
Twitter Scraper



Twitter's API is annoying to work with, and has lots of limitations — luckily their frontend (JavaScript) has its own API, which I reverse-engineered. No API rate limits. No tokens needed. No restrictions. Extremely fast.

You can use this library to get the text of any user's Tweets trivially.

Installation

```
1 go get -u github.com/n0madic/twitter-scraper
```

Usage

Authentication

Now all methods require authentication!

Login

```
1 err := scraper.Login("username", "password")
```

Use username to login, not email! But if you have email confirmation, use email address in addition:

```
1 err := scraper.Login("username", "password", "email")
```

If you have two-factor authentication, use code:

```
1 err := scraper.Login("username", "password", "code")
```

Status of login can be checked with:

```
1 scraper.IsLoggedIn()
```

Logout (clear session):

```
1 scraper.Logout()
```

If you want save session between restarts, you can save cookies with `scraper.GetCookies()` and restore with `scraper.SetCookies()`.

For example, save cookies:

```
1 cookies := scraper.GetCookies()
2 // serialize to JSON
3 js, _ := json.Marshal(cookies)
4 // save to file
5 f, _ = os.Create("cookies.json")
6 f.Write(js)
```

and load cookies:

```
1 f, _ := os.Open("cookies.json")
2 // deserialize from JSON
3 var cookies []*http.Cookie
4 json.NewDecoder(f).Decode(&cookies)
5 // load cookies
6 scraper.SetCookies(cookies)
7 // check login status
8 scraper.IsLoggedIn()
```

Open account If you don't want to use your account, you can try login as a Twitter app:

```
1 err := scraper.LoginOpenAccount()
```

Get user tweets

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6     twitterscraper "github.com/n0madic/twitter-scraper"
7 )
8
9 func main() {
10     scraper := twitterscraper.New()
11     err := scraper.LoginOpenAccount()
12     if err != nil {
13         panic(err)
14     }
15     for tweet := range scraper.GetTweets(context.Background(), "Twitter", 50) {
16         if tweet.Error != nil {
17             panic(tweet.Error)
18         }
19         fmt.Println(tweet.Text)
20     }
21 }
```

It appears you can ask for up to 50 tweets.

Get single tweet

```
1 package main
2
3 import (
4     "fmt"
5
6     twitterscraper "github.com/n0madic/twitter-scraper"
7 )
8
9 func main() {
10     scraper := twitterscraper.New()
11     err := scraper.Login(username, password)
12     if err != nil {
13         panic(err)
14     }
15     tweet, err := scraper.GetTweet("1328684389388185600")
16     if err != nil {
17         panic(err)
18     }
19     fmt.Println(tweet.Text)
20 }
```

Search tweets by query standard operators

Now the search only works for authenticated users!

Tweets containing “twitter” and “scraper” and “data“, filtering out retweets:

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6
7     twitterscraper "github.com/n0madic/twitter-scraper"
8 )
9
10 func main() {
11     scraper := twitterscraper.New()
12     err := scraper.Login(username, password)
13     if err != nil {
14         panic(err)
15     }
16     for tweet := range scraper.SearchTweets(context.Background(),
17         "twitter scraper data -filter:retweets", 50) {
```

```

17         if tweet.Error != nil {
18             panic(tweet.Error)
19         }
20         fmt.Println(tweet.Text)
21     }
22 }

```

The search ends if we have 50 tweets.

See Rules and filtering for build standard queries.

Set search mode

```
1 scraper.SetSearchMode(twittercraper.SearchLatest)
```

Options:

- `twittercraper.SearchTop` - default mode
- `twittercraper.SearchLatest` - live mode
- `twittercraper.SearchPhotos` - image mode
- `twittercraper.SearchVideos` - video mode
- `twittercraper.SearchUsers` - user mode

Get profile

```

1 package main
2
3 import (
4     "fmt"
5     twittercraper "github.com/n0madic/twitter-scraper"
6 )
7
8 func main() {
9     scraper := twittercraper.New()
10    scraper.LoginOpenAccount()
11    profile, err := scraper.GetProfile("Twitter")
12    if err != nil {
13        panic(err)
14    }
15    fmt.Printf("%+v\n", profile)
16 }

```

Search profiles by query

```

1 package main
2

```

```

3 import (
4     "context"
5     "fmt"
6     twitterscraper "github.com/n0madic/twitter-scraper"
7 )
8
9 func main() {
10     scraper := twitterscraper.New().SetSearchMode(twitterscraper.
        SearchUsers)
11     err := scraper.Login(username, password)
12     if err != nil {
13         panic(err)
14     }
15     for profile := range scraper.SearchProfiles(context.Background(), "
        Twitter", 50) {
16         if profile.Error != nil {
17             panic(profile.Error)
18         }
19         fmt.Println(profile.Name)
20     }
21 }
```

Get trends

```

1 package main
2
3 import (
4     "fmt"
5     twitterscraper "github.com/n0madic/twitter-scraper"
6 )
7
8 func main() {
9     scraper := twitterscraper.New()
10    trends, err := scraper.GetTrends()
11    if err != nil {
12        panic(err)
13    }
14    fmt.Println(trends)
15 }
```

Use Proxy

Support HTTP(s) and SOCKS5 proxy

with HTTP

```

1 err := scraper.SetProxy("http://localhost:3128")
```

```
2 if err != nil {
3     panic(err)
4 }
```

with SOCKS5

```
1 err := scraper.SetProxy("socks5://localhost:1080")
2 if err != nil {
3     panic(err)
4 }
```

Delay requests

Add delay between API requests (in seconds)

```
1 scraper.WithDelay(5)
```

Load timeline with tweet replies

```
1 scraper.WithReplies(true)
```