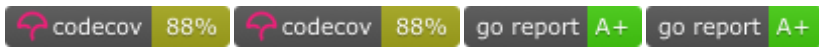

go-mastodon



Usage

Application

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6     "log"
7
8     "github.com/mattn/go-mastodon"
9 )
10
11 func main() {
12     app, err := mastodon.RegisterApp(context.Background(), &mastodon.
13         AppConfig{
14             Server: "https://mstdn.jp",
15             ClientName: "client-name",
16             Scopes: "read write follow",
17             Website: "https://github.com/mattn/go-mastodon",
18         })
19     if err != nil {
20         log.Fatal(err)
21     }
22     fmt.Printf("client-id : %s\n", app.ClientID)
23     fmt.Printf("client-secret: %s\n", app.ClientSecret)
24 }
```

Client

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6     "log"
7
8     "github.com/mattn/go-mastodon"
9 )
10
11 func main() {
```

```

12     c := mastodon.NewClient(&mastodon.Config{
13         Server:      "https://mstdn.jp",
14         ClientID:     "client-id",
15         ClientSecret: "client-secret",
16     })
17     err := c.Authenticate(context.Background(), "your-email", "your-
18         password")
19     if err != nil {
20         log.Fatal(err)
21     }
22     timeline, err := c.GetTimelineHome(context.Background(), nil)
23     if err != nil {
24         log.Fatal(err)
25     }
26     for i := len(timeline) - 1; i >= 0; i-- {
27         fmt.Println(timeline[i])
28     }

```

Client with Token

This option lets the user avoid storing login credentials in the application. Instead, the user's Mastodon server provides an access token which is used to authenticate. This token can be stored in the application, but should be guarded.

```

1  package main
2
3  import (
4      "context"
5      "fmt"
6      "log"
7      "net/url"
8
9      "github.com/matttn/go-mastodon"
10 )
11
12 func main() {
13     appConfig := &mastodon.AppConfig{
14         Server:      "https://stranger.social",
15         ClientName:  "client-name",
16         Scopes:      "read write follow",
17         Website:     "https://github.com/matttn/go-mastodon",
18         RedirectURIs: "urn:ietf:wg:oauth:2.0:oob",
19     }
20     app, err := mastodon.RegisterApp(context.Background(), appConfig)
21     if err != nil {
22         log.Fatal(err)
23     }

```

```

24
25     // Have the user manually get the token and send it back to us
26     u, err := url.Parse(app.AuthURI)
27     if err != nil {
28         log.Fatal(err)
29     }
30     fmt.Printf("Open your browser to \n%s\n and copy/paste the given
        token\n", u)
31     var token string
32     fmt.Print("Paste the token here:")
33     fmt.Scanln(&token)
34     config := &mastodon.Config{
35         Server:      "https://stranger.social",
36         ClientID:     app.ClientID,
37         ClientSecret: app.ClientSecret,
38         AccessToken:  token,
39     }
40
41     c := mastodon.NewClient(config)
42     err = c.AuthenticateToken(context.Background(), token, "urn:ietf:wg
        :oauth:2.0:oob")
43     if err != nil {
44         log.Fatal((err)
45     }
46
47     acct, err := c.GetAccountCurrentUser(context.Background())
48     if err != nil {
49         log.Fatal((err)
50     }
51     fmt.Printf("Account is %v\n", acct)
52 }

```

Status of implementations

- ☒ GET /api/v1/accounts/:id
- ☒ GET /api/v1/accounts/verify_credentials
- ☒ PATCH /api/v1/accounts/update_credentials
- ☒ GET /api/v1/accounts/:id/followers
- ☒ GET /api/v1/accounts/:id/following
- ☒ GET /api/v1/accounts/:id/statuses
- ☒ POST /api/v1/accounts/:id/follow
- ☒ POST /api/v1/accounts/:id/unfollow
- ☒ GET /api/v1/accounts/:id/block
- ☒ GET /api/v1/accounts/:id/unblock
- ☒ GET /api/v1/accounts/:id/mute

-
- ☒ GET /api/v1/accounts/:id/unmute
 - ☒ GET /api/v1/accounts/:id/lists
 - ☒ GET /api/v1/accounts/relationships
 - ☒ GET /api/v1/accounts/search
 - ☒ GET /api/v1/apps/verify_credentials
 - ☒ GET /api/v1/bookmarks
 - ☒ POST /api/v1/apps
 - ☒ GET /api/v1/blocks
 - ☒ GET /api/v1/conversations
 - ☒ DELETE /api/v1/conversations/:id
 - ☒ POST /api/v1/conversations/:id/read
 - ☒ GET /api/v1/favourites
 - ☒ GET /api/v1/filters
 - ☒ POST /api/v1/filters
 - ☒ GET /api/v1/filters/:id
 - ☒ PUT /api/v1/filters/:id
 - ☒ DELETE /api/v1/filters/:id
 - ☒ GET /api/v1/follow_requests
 - ☒ POST /api/v1/follow_requests/:id/authorize
 - ☒ POST /api/v1/follow_requests/:id/reject
 - ☒ GET /api/v1/followed_tags
 - ☒ POST /api/v1/follows
 - ☒ GET /api/v1/instance
 - ☒ GET /api/v1/instance/activity
 - ☒ GET /api/v1/instance/peers
 - ☒ GET /api/v1/lists
 - ☒ GET /api/v1/lists/:id/accounts
 - ☒ GET /api/v1/lists/:id
 - ☒ POST /api/v1/lists
 - ☒ PUT /api/v1/lists/:id
 - ☒ DELETE /api/v1/lists/:id
 - ☒ POST /api/v1/lists/:id/accounts
 - ☒ DELETE /api/v1/lists/:id/accounts
 - ☒ POST /api/v1/media
 - ☒ GET /api/v1/mutes
 - ☒ GET /api/v1/notifications
 - ☒ GET /api/v1/notifications/:id
 - ☒ POST /api/v1/notifications/:id/dismiss

-
- ☒ POST /api/v1/notifications/clear
 - ☒ POST /api/v1/push/subscription
 - ☒ GET /api/v1/push/subscription
 - ☒ PUT /api/v1/push/subscription
 - ☒ DELETE /api/v1/push/subscription
 - ☒ GET /api/v1/reports
 - ☒ POST /api/v1/reports
 - ☒ GET /api/v2/search
 - ☒ GET /api/v1/statuses/:id
 - ☒ GET /api/v1/statuses/:id/context
 - ☒ GET /api/v1/statuses/:id/card
 - ☒ GET /api/v1/statuses/:id/history
 - ☒ GET /api/v1/statuses/:id/reblogged_by
 - ☒ GET /api/v1/statuses/:id/source
 - ☒ GET /api/v1/statuses/:id/favourited_by
 - ☒ POST /api/v1/statuses
 - ☒ PUT /api/v1/statuses/:id
 - ☒ DELETE /api/v1/statuses/:id
 - ☒ POST /api/v1/statuses/:id/reblog
 - ☒ POST /api/v1/statuses/:id/unreblog
 - ☒ POST /api/v1/statuses/:id/favourite
 - ☒ POST /api/v1/statuses/:id/unfavourite
 - ☒ POST /api/v1/statuses/:id/bookmark
 - ☒ POST /api/v1/statuses/:id/unbookmark
 - ☒ GET /api/v1/timelines/home
 - ☒ GET /api/v1/timelines/public
 - ☒ GET /api/v1/timelines/tag/:hashtag
 - ☒ GET /api/v1/timelines/list/:id
 - ☒ GET /api/v1/streaming/user
 - ☒ GET /api/v1/streaming/public
 - ☒ GET /api/v1/streaming/hashtag?tag=:hashtag
 - ☒ GET /api/v1/streaming/hashtag/local?tag=:hashtag
 - ☒ GET /api/v1/streaming/list?list=:list_id
 - ☒ GET /api/v1/streaming/direct

Installation

```
1 go install github.com/mattn/go-mastodon@latest
```

License

MIT

Author

Yasuhiro Matsumoto (a.k.a. mattn)