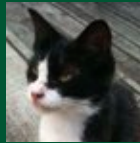


Golangでいろいろ 試してみた

Go弱の会 3弱

自己紹介



こばやし けんいち (@Niratama)

ソーシャルゲーム屋

Perlが主戦場

最近はインフラとかそっちの作業が多くて、
がっつりプログラム書いてない感じ

とりあえずなんか書いてみた

以前にChiba.pmで発表した、Perl+Mojoliciousを使って
スライドにTweetをリアルタイム表示する仕組みのサーバ
部分をGolangで実装してみた

ハッシュタグ `#gojaku` をつけてツイートしてみよう！

```
__人人人人人人人人人人__  
>  NHK NEWS WEBのパクリ  <  
—Y^Y^Y^Y^Y^Y^Y^Y^Y^Y—
```

どうやってんの？

1. WebSocketの接続を待ち受け
2. **Twitter Streaming API**に接続して、指定されたキーワードの入ったTweetのみフィルタリングして取得
3. 受け取ったTweetを再構成してWebSocketでブラウザに送信
4. ブラウザでスライドに重ねて表示

今回使ったモジュール

- `flag`
- `github.com/rakyll/globalconf`
- `net/http`
- `code.google.com/p/go.net/websocket`
- `darkhelmet/twitterstream`
- `encoding/json`
- `time`

flag

コマンドラインのオプションを解析するパッケージ

```
// オプションを設定する
name := flag.String("name", "niratama", "Your name")
// 最後にParseすると値が設定される
flag.Parse()
// nameに入るのはポインタ
fmt.Printf("Hello %s!", *name)
```

rakyll/globalconf

HOMEディレクトリにある設定ファイルや環境変数から値を読み込むパッケージ

```
// オプションを設定する
name := flag.String("name", "niratama", "Your name")
// 最後にParseすると値が設定される。flag側のParse()は不要
conf, _ := globalconf.New("appname")
conf.ParseAll()
// nameに入るのはポインタ
fmt.Printf("Hello %s!", *name)
```

RubyGemsのpit的な用途に使える

net/http

WAF使おうかと思ったけど、今回はstaticなファイルとWebSocketだけなのでシンプルに

staticなファイルの配信は `http.FileServer()` と `http.Dir()` を使うことで簡単にできる

```
http.Handle("/", http.FileServer(http.Dir("/tmp")))
```


websocket

クライアントからサーバに対してはリクエスト時以外はデータのやりとりはないので、本当はWebSocketである必然性はない

`websocket.Handler()` に渡した関数に `websocket.Conn` 型の構造体が渡るので、それに対して `Read()` したり `Write()` したりする

`http.Request` は `websocket.Conn` の `Request()` で取得できるので、パラメータの受け取りなんかはこれを利用するとい

websocket

// ハンドラ

```
func wsHandler(ws *websocket.Conn) {  
    req := ws.Request() // http.Requestが返る  
    name := req.FormValue("name")  
    for () {  
        message := []byte(fmt.Sprintf("Hello %s!", name))  
        ws.Write(message)  
    }  
}
```

```
func main() {  
    // websocket.Handler()がhttpのハンドラとWebSocketのハンドラを変換してくれる  
    http.Handle("/ws", websocket.Handler(wsHandler))  
    http.ListenAndServe(":3000", nil)  
}
```

darkhelmet/twitterstream

Twitter Streaming APIにアクセスするためのパッケージ

ChimeraCoder/anacondaがメジャーな感じだったけど
対応していなかったの

```
// Twitter Streaming APIに接続するクライアントを作成
client := twitterstream.NewClient(*consumerKey, *consumerSecret, *accessToken, *accessTokenSecret)
// statuses/filter APIに接続。検索条件は"gorlang"が含まれるツイート
conn, _ := client.Track('gorlang')
for {
    // Tweetを一つ受け取る
    tweet, _ := conn.Next()
    fmt.Printf("%s: %s\n", tweet.User.ScreenName, tweet.Text);
}
```

`conn.Next()` でブロックしてしまうのが微妙

encoding/json

darkhelmet/twitterstreamはAPIからのデータを自前でUnmarshalするので、今回必要なのはWebSocketで送信するときのデータをMarshalする時だけ

Marshalしたデータは[]byteなので、そのままWebSocketにWrite()できる

アノテーションを使えば実際の構造体の変数名とJSONでの名前を変えることができる

encoding/json

```
type WSTweet struct {  
    Text      string `json:"text"`  
    Name      string `json:"name"`  
    ScreenName string `json:"screen_name"`  
    ProfileImageUrl string `json:"profile_image_url"`  
}  
  
data := WSTweet{  
    tweet.Text,  
    tweet.User.Name,  
    tweet.User.ScreenName,  
    tweet.User.ProfileImageUrl,  
}  
// jsonは[]byteで出力される。  
json, _ := json.Marshal(data)
```

time

`time.ParseDuration()` で文字列表記から `time.Duration` 形式に変換できる

```
timeout, _ = time.ParseDuration(timeoutString)
```

クライアント

- スライド表示プログラムはremarkというJavaScript製のツールを利用
 - HTMLファイル内に埋め込んだMarkdownファイルをスライドとして表示できる
- ツイート表示部分とタイマーゲージ表示部分はjQueryを使って実装
 - スライド表示と独立しているので他のツールと組み合わせることもできるかも？

今回の資料とファイルはGitHubに置いておきます

<https://github.com/niratama/gojaku3>

ご自由にご利用ください