Digitally signed by 佐藤 豊
Date: 2020.09.22 12:07:15 +09'00'

2020年9月21日 投稿者: SATOXITS

GShell 0.4.8 – 自分認証方式

社長：ずいぶん早く目が覚めました。

開発：今日は間に合いましたね。後半に入ったところです。

基盤：みなさん朝から応援してますねｗ

社長：そろそろ終わるというのにいつもの人はどうしてるんでしょう？

開発：しかたが無い、緊急出動で。コピペしてぷちっ。

基盤：やはり便利な20000chが必要なのではｗ

社長：これは不便が文化を作るという面白い世界ですからね。顔文字もしかり。AAもしかり。

基盤：Vivaldiの広告ブロック機能、ばっちりですね。

開発：まあ、大抵は指定した class とか id とかのエレメントを表示しなきゃいいだけの話ですけどね。URLでもいいし。いつも見ているサイトなら簡単ですよね。client-side CSS が使えれば。

基盤：インスペクタで要素を調査して条件を決めてブロックできると良いですね。

社長：data URI とか JavaScript で埋め込んできたりして。

開発：悪いイメージを付けてほしくないですね。

基盤：URLでブロックするならプロキシでもやればよいのではないかと。実際にプロキシは無くても、このサイトはプロキシ経由にするという指示にして。フォールバックしなければ切れますよね。

開発：そもそも、このサイト、このURLにはアクセスしないっていう設定をブラウザでできれば良いんですよね。

社長：複数ブラウザ使ってますからねぇ。

開発：まあそのへんをGShell経由で共有するのもよいかと。

基盤：これは、新着投稿を読み上げソフトで読ませとくと良いかもですねw

社長：ぜひ吉田くんの声で。

基盤：このブログも読ませて確認して校正すると良いかも。

開発：終了しました。

社長：復調めでたし。

自分認証方式

社長：さて、JavaScript から Golang の GShellに接続する時の認証方法ですが。

基盤：これは今ブラウザ上でクリックした自分からの接続だということを確認する事ですね。

社長：故に我有りみたいな。

開発：ユーザによるインタラクティブな承認、パスワード方式、スクリプト側のURLによる承認、スクリプトの署名による承認、スクリプトに埋め込んだ証明書での承認、あ

たりかと。

社長：まずスクリプトを含むURLを閲覧した時に、Golangサーバ側でユーザが承認してクレデンシャルを生成してブラウザのローカルストレージに保存する、その後変更がなければそのクレデンシャルが有効、っていうような流れが良いかなと思います。

基盤：ユーザがインタラクティブにOKする時って何かのポップアップですよね。ブラウザだと起動するのが重いことがあって嫌なんですが。

開発：ターミナルとか、あるいは専用の簡易窓とかが良いかも知れないですね。これもGoサーバで開いてやれば良いと思います。

基盤：承認を要求したブラウザに、Go側からポップアップを作れると良いのでは無いかと思うのですが。

社長：Golang側で一時パスワードを生成して表示、クリップボードに保存、ブラウザでペーストするとOKみたいのが良いかも。

基盤：クリップボードは魅力ですが、時々妙に重いことがあるのがちょっとですね。

開発：IMEの辞書に一時パスワードを登録して、特別な読みを入れるとそれがペーストされるとかが良いかも。

基盤：でも、10秒以内に yes / no ボタンを押す、だけというのが一番かなと思います。

社長：reCAPTCHAでは無いですが、サーバで画像をPNGで生成してクリッカブルマップで「そこ」をクリックさせると良いかも知れません。

開発：それ、それで行きましょう。「そこ」を何にするかが面白そうです。

基盤：物理的にキーが押されたというイベントが偽造できないなら、それでも良いですよね。「右シフトキーを押してください」みたいなのを画像で返す。

開発：それも良いですね。モールスで打てとか。

社長：キーが押されたことを確認するのはGolang側のサーバですよね。

開発：JavaScriptクライアントとGolangサーバの両方で確認して、押された時刻をマイクロ秒単位で確認するとか。

基盤：確かセキュリティ上の制限で、JavaScriptの時刻の精度はミリ秒に抑えられてますね。

開発：いずれにしても、合意したクレデンシャルで継続的に承認というのは、Go-GShell/HTTP とクライアントとの間の仮想セッションですから、一般化するならCookieと同じようなものになるか、そもそもCookieを使えば良いということになりそうです。

社長：でも結局、普通にパスワードでいいんじゃないかって気もしますけどね。長いランダムなのを生成して、ブラウザとGShellサーバに記録させる。

開発：まあそうかも知れません…

Goからブラウザを制御する

開発：表示系の調整が未完ですが、だいたいできました。

社長：結局、ブラウザからGoを使うより、Goからブラウザを使うほうが面白いですね。

開発：まあそっちのほうが安全ということもあるんですが。

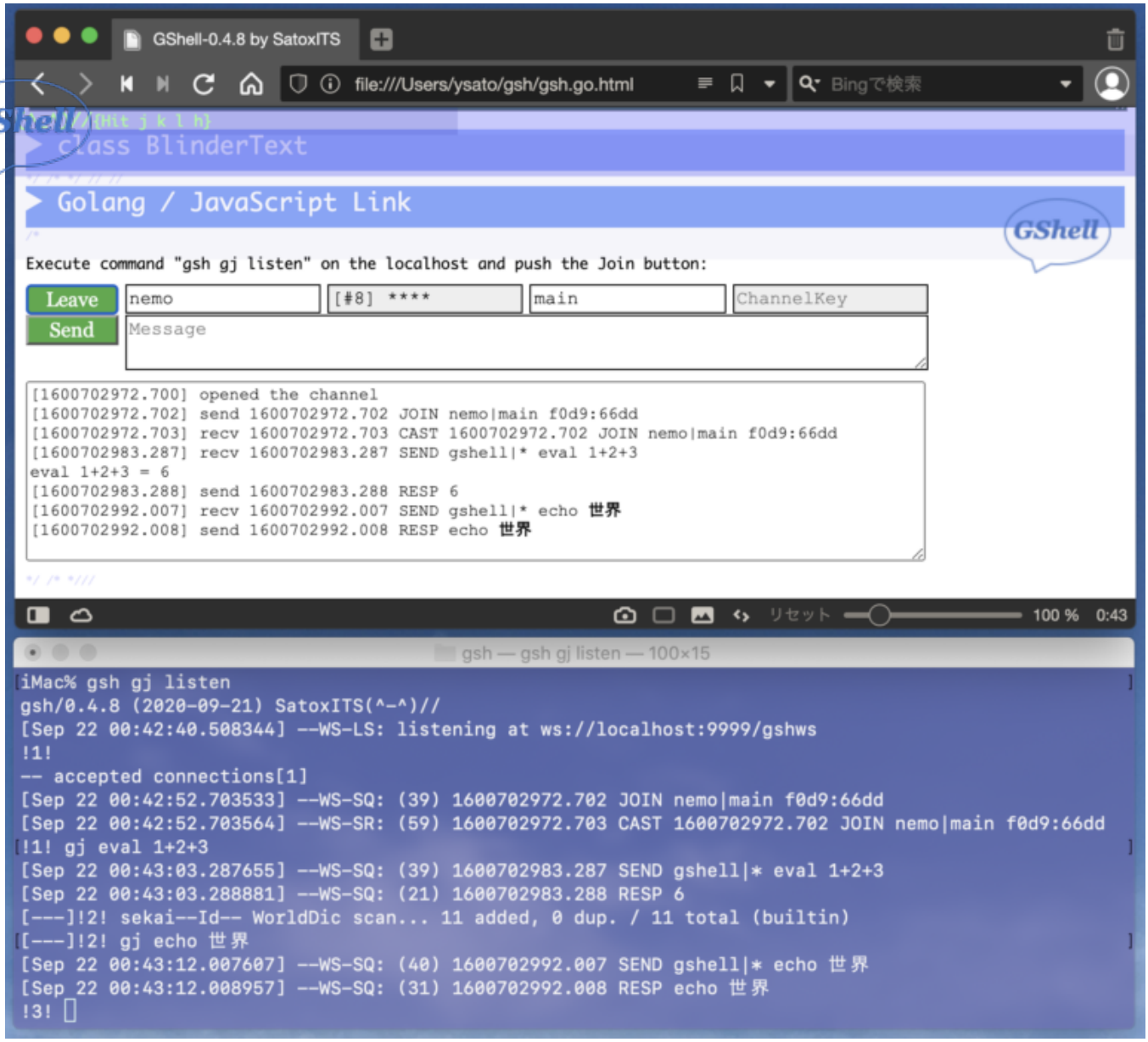
```
(^_^)//{Hit j k l h}
```

社長：長い間やりたいと思っていたことが、ようやく実現しました。

開発：やってみたらあっさりでしたけどね。WebSocketのおかげです。

社長：自前プロトコルはVIABUSを参考にして設計しましょう。

開発：まだ標準命に毒されてなかった頃の発想に戻るというわけですね。



社長：この、パスワードを入力する時にチェックディジットを表示する機能は、単体でも使えますね。

開発：まあ、そのようにBlinderTextというクラスを作りました。特にパスワード専用ということではなく、その場の入力を隠すだけでもなく、読めないようにテキストを表示しつつ編集したり、暗号化とかをその場でできるIMEとして拡充していきたいと思います。

基盤：バックエンドのGoでパスワード管理して、ブラウザのJavaScriptから参照できると便利ですよね。

開発：暗号化、署名も、JavaScriptでやると制約がかったるいので、Goに送ってやり

たいと思います。

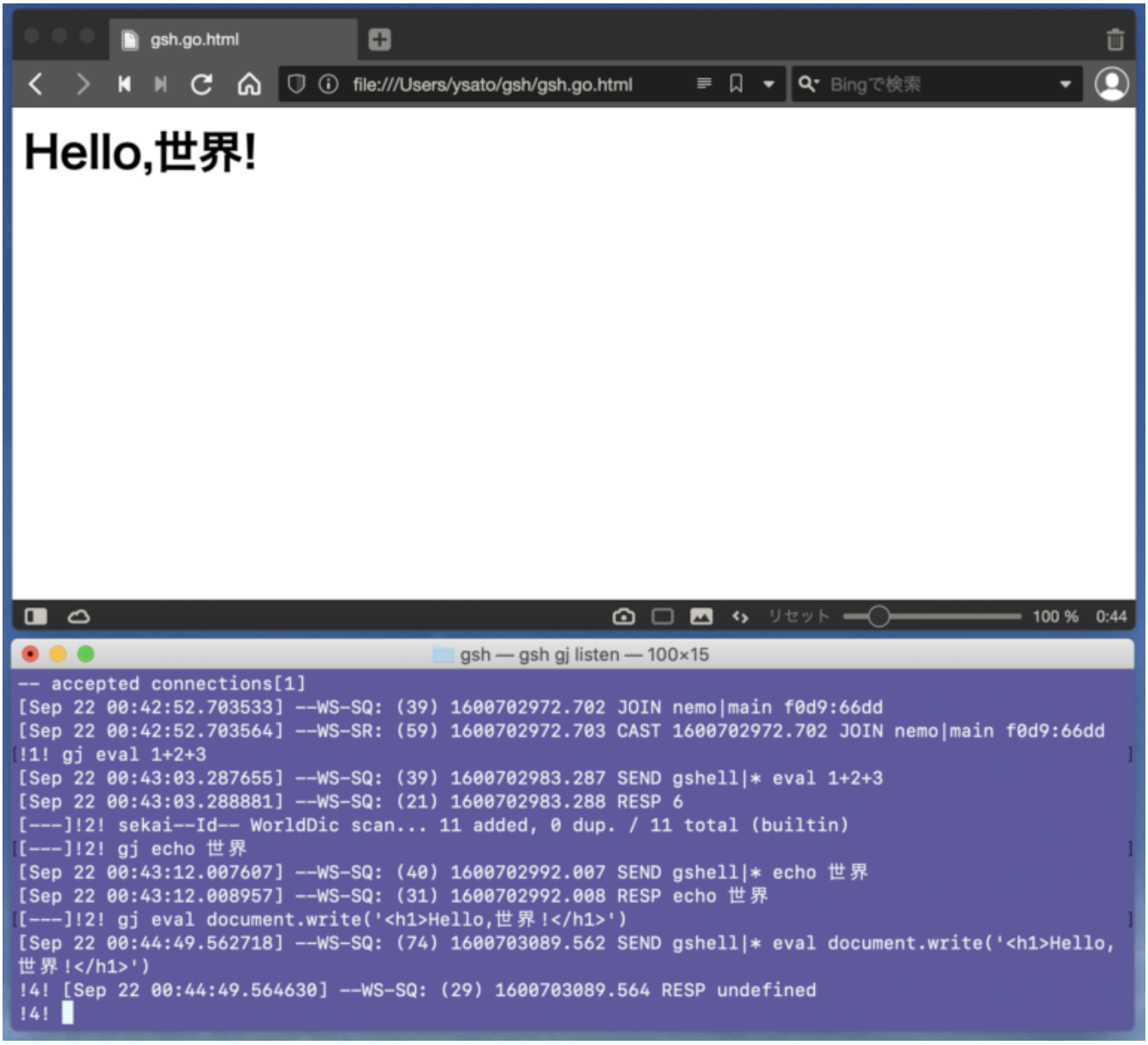
社長 :さっきから、Vivaldiの入力の応答が遅くてアタマがおかしくなりそうです。Google IMEでは変換して送信終了しているのに。実際にVivaldiでブロックエディタに出てくるのに何秒も掛かる…

基盤：さっきMacMiniをリブートした後に、何かの初期化がまだ終わってないんでしょうかね。CPU負荷は全然問題ないのですが。

社長：あれ？改善しました。良くなったり、悪くなったりです。

基盤：iMacのほうならサクサクですが。

開発：でもってシメはやはりこれで。



基盤：viで書いて保存したら即ブラウザで確認できるというのが良いですね。

社長：ブラウザの一つの窓でHTMLとして編集すると、他の窓でリアルタイムに表示されるというのも。しかも複数の種類のブラウザで同時に。

開発：もはや、原理的には簡単です。

— 2020-0921 SatoxITS

gsh-0.4.8.go_-1 ダウンロード

```
/*
```

GShell version 0.4.8 // 2020-09-21 // SatoxITS

GShell // a General purpose Shell built on the top of Golang

```
It is a shell for myself, by myself, of myself. -SatoxITS(^-^)

0 Fork Stop Unfold Digest Source */ /*
```

▶ Statement

```
*/ /*
```

▶ Features

```
*/ /*
```

▶ Index

```
*/ //
```

▶ Go Source

```
//
```

▶ Considerations

```
// /*
```

▶ References

```
*/ /*
```

▶ Raw Source

```
*/ /*
```

▶ GJScript

```
*/ /*
```

```
GJShell Console // gsh-0.4.8-2020-09-21-SatoxITS
%
```

```
*/ /*
```

class BlinderText

```
*/ /* */ // //
```

Golang / JavaScript Link

```
/*
```

Execute command "gsh gj listen" on the localhost and push the Join button:

UserName | UserKey | ChannelName | ChannelKey

Message

```
*/ /* *///
```