株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.09.11 10:04:08 +09'00'

2020年9月10日 投稿者: SATOXITS

GShell 0.3.7 – HTMLコメント付け

開発：今日は調べごとの1日でした。

社長：まずHTMLに埋め込む署名情報をどう表現するかですね。自前タグと自前属性で行くのか。

基盤：<signature> みたいな感じでしょうか。

開発：実際に色々やってみた結果として、今日のところは、汎用タグのspanを使い、data-* というユーザ定義用に用意されている属性を使うという結論です。

```
//<span hidden id=”gsh-digest” data-target-id=”gsh” data-crc32u=”3331567713″ data-length=”202771″ data-lines=”7141″ data-time=”1599750250639″></span>
```

開発：見た目はカッコ悪いですが、DOM + JavaScriptでの処理はスッキリします。innerHTML で、DOMをダンプというかHTMLにプリティプリントしてくれる時に内容と順序が再現されるところが決め手です。

基盤：innerHTMLが無いならhiddenもいらなそうです。

開発：あそうか。

社長：metaタグがnameとcontentを自前定義できて、署名という用途に馴染むと思われますが、我々は任意のエレメントに署名したいので、筋が違うという感じですね。

開発：それで、これに電子署名をつけようという段になって、JavaScriptの暗号APIがSSL接続中でないと使えないという不思議な制約にひっかかってしまったわけです。開発がしにくいので、こっちから片付けようかと。

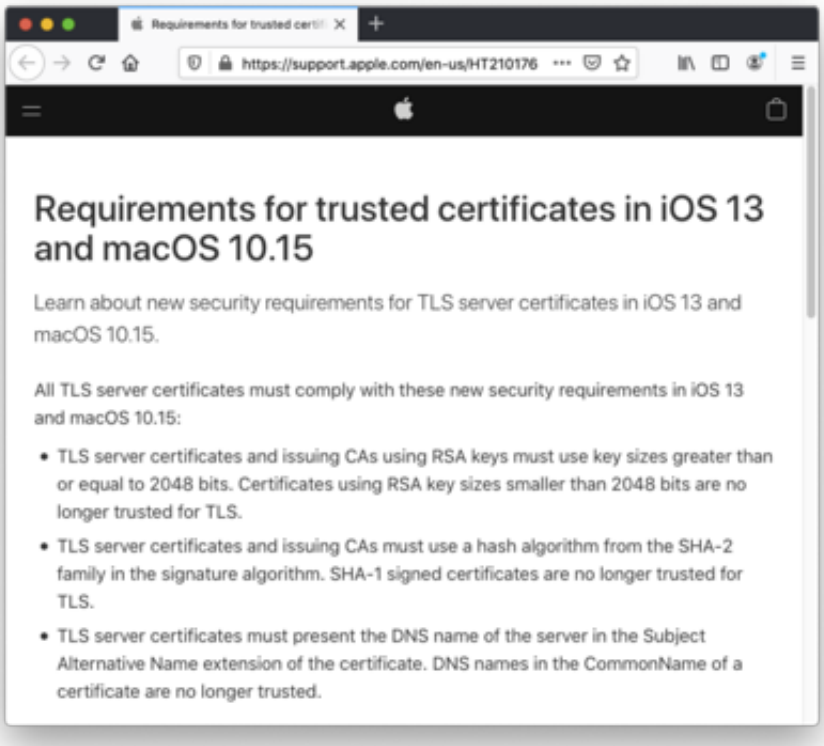
基盤：うちのサーバたちにもちゃんとした SSL証明書を付けてあげたいですね。

社長：ドメインが100あるから、ひとつ1万円でも100万円かかりますね。

経理：あり得ない。

基盤：最近特にSSLの証明書の内容にうるさくなって、なまじHTTPS/SSLでいい加減な証明書を付けてるとブラウザが開いてくれないんですよね。HTTPSのほうがましみたいな。

開発：ブラウザは「この証明書は信頼できない」とか「standardに従ってないからペケ」っていうだけで、具体的に何が問題にされてるの知らなかったので、今日は少し探しました。すると、Appleの macOS / iOSでの判断基準という面白いのがひっかかりました。日付は2019年11月03日とあります。



社長：鍵長は2048ビット以上、SHA-2以上というのは、まあ今日的な基準なんでしょうね。

基盤：うちはMD5使ってたりしますがw

開発：ただ、CommonName を信用しないというのはちょっとびっくりしました。あと、巨編の7月移行の証明書については、鍵の用途を提示する拡張機能を使っていることとか。鍵の有効期限は825日以内にしてあることとか。

> Connections to TLS servers violating these new requirements will fail and may cause network failures, apps to fail, and websites to not load in Safari in iOS 13 and macOS 10.15.
> Published Date: November 03, 2019

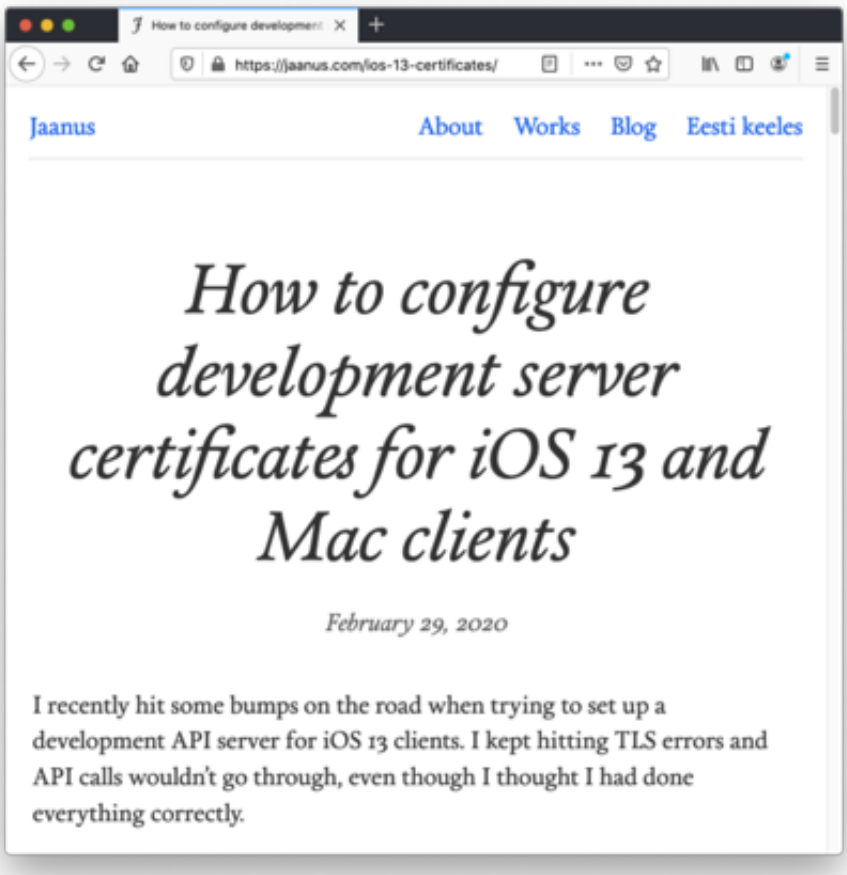
社長：なかなか過激というか高飛車というか。でも、iPhoneとmacOSのユーザとして、特に困ったという経験はしなかったです。

基盤：うちが Windows から Macに復帰した頃には混乱が収まっていたとか。

社長：実際iPhoneで問題を感じが事は無いですね。それがほぼ世間の常識状態になってたということかも知れません。

基盤：それで、具体的にはどういう作業が必要なんでしょう。

開発：基本的には鍵と証明書の項目を今日的なスタンダードに合わせて作り直すということかとは思います。OpenSSLで。で、検索したらまさにこの問題に対処した記録を公開してくれてた人がいました。



基盤：異様にフォントがデカイんですが。

社長：素晴らしい。見習いたいものです (^-^;

開発：「I am not sure if commonName does anything, but I specify the server name there.」なんてところはノリに共通感を感じますがw

社長：GShellは立ち上がった時にSSL証明書を生成するようにしたいですね。サーバとしてもクライアントとしても。

開発：OpenSSLを動的ライブラリとして使えるので楽ですね。

基盤：有効期限が短すぎる証明書も、スタンダードじゃないって怒られたりして。

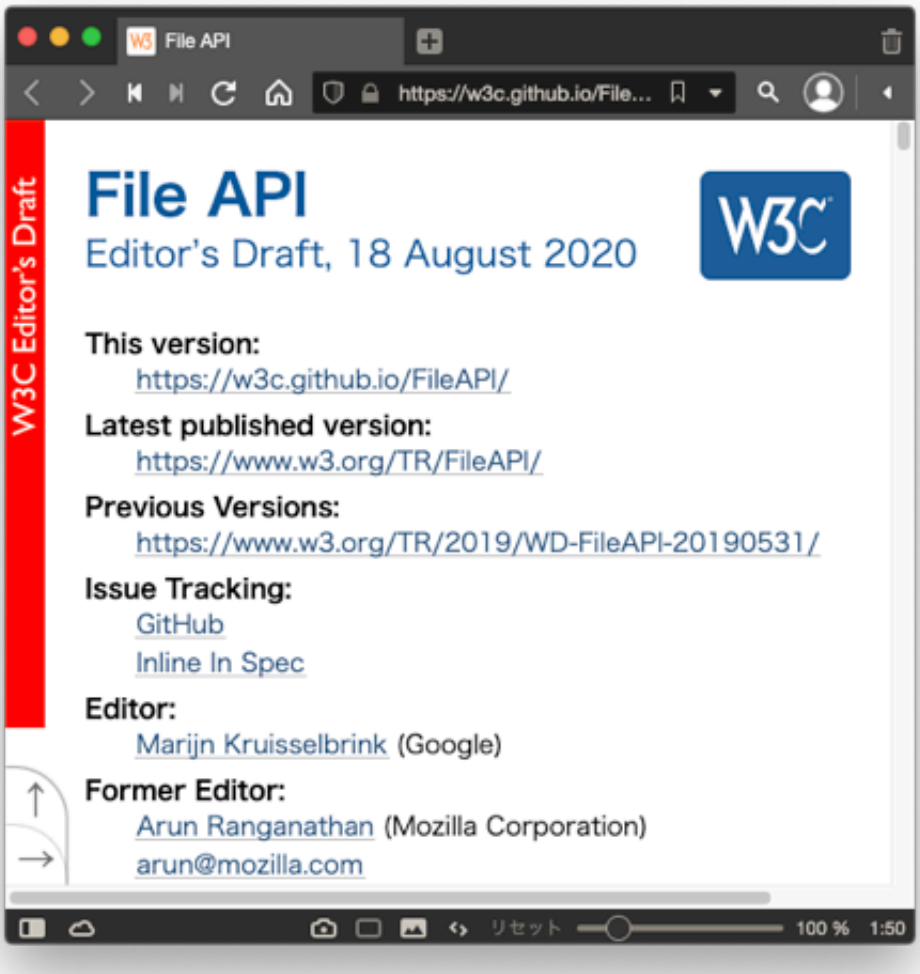
開発：あとは、localStorageとの関係で、JavaScript の FileReader というのが目に止まりました。Web Worker という、UI 側とは独立したバックグランド的な処理をす

る仕組みなんだそうで、一種のサーバ的な動作をするもののようにも見えます。

基盤：JavaScript でサーバも書いちゃうという話ですかね。なら、自分をHTTPS/SSLサーバにしてそこに繋げば、crypto の使用コンテクスト制約も解消できたりして。

開発：・・・　だと良いですね。

開発：で、MDNのFileReader から規格書をたどると、W3C のこれに行きます。



社長：W3Cの色使いとかフォント使いは、メリハリが効いてますよね。

基盤：Moz://a MDN は異様に暗いんですよね。

開発： ここが、JavaScript の GShell と、Golang の GShell との接合点になるのかも

知れません。readAsDataURL() なんていうメソッドもあったりします。

社長：data URL は GShellにとって欠かせない標準の一つですね。

基盤：読むのは良いとして、ローカルファイルに書くにはどうするんでしょう。

開発：これはふつう、人間が「ダウンロードボタンをクリック」するわけですが、JavaScript でクリックしちゃんだとか。

基盤：JavaSciript GShell で生成したGolang GShellコードを自動ダウンロードしたり、Golang GShellで実行した結果をJavaScript GShellで吸い上げるとかも。

社長：これって、よっぽど JavaScript の実行権限管理がしっかりしてないと恐ろしように思います。

開発：どこからやってきた風来坊かわかりませんから、やはりスクリプトの電子署名で認証するのが良いように思います。

基盤：ところで今日の題目は「HTMLコメント付け」ってなってますけど。

社長：localStorage なり FILE API なりでローカルに高速なストレージが使えることを想定すると、使い方が大きく広がると思うんですよね。その一つが、HTMLページのエレメントにコメントを付けたりメモを書いたりという使い方。自分専用でも良いしアップロードして共有しても良い。コメントはでっかい画像つきだったりとか。ともかく、HTMLページをハブ的なフロントエンドにして、ローカルな機能やリソースをコントロールする。

開発：OfficeのWordのコメント機能はいけてますよね。DOMのエレメントに属性としてコメントとか付加情報とか変更履歴とか残せば。

社長：見た目的にはGoogleマップのピン立てですね。文章中のコメント付きのところにこの、 

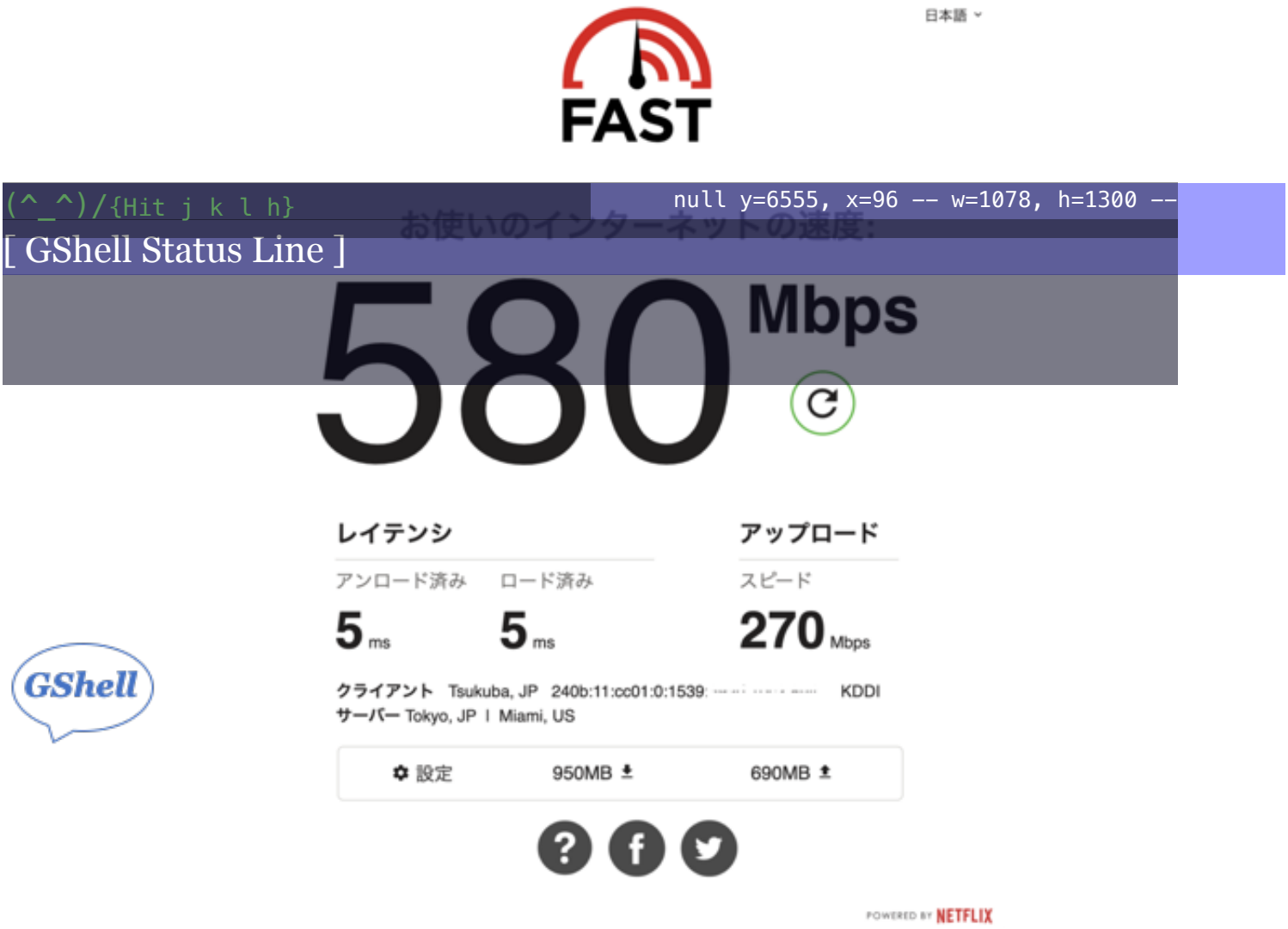
マークを立てる。

開発：ひょっとすると、ウェブサーバに繋ぐのはそのページを最初に見る時だけかも知

れないですね。というか、そもそも、html.gz とかしてダウンロードして、ローカルに実行するだけかも知れない。

社長：必要な時にだけサーバにつないで情報交換ができるのが良いですね。自在でシームレス。

基盤：ところで今、Chromeのタブを整理してたらこんなのが出てきました。



開発：驚速。いつのデータでしょうね？

社長：測定ミスですかね。時計の進みが遅かったとか・・・

基盤：でも、クライアントはこの間まで牛久て表示されてたと思うのですが、今はTsukubaになってますね。

社長：なぜ牛久だったんですかね？

開発：大仏の胎内にサーバがあったとか。

— 2020-0910 SatoxITS

http-im3-gsh-gsh-0.3.7.go_ ダウンロード

// /*

GShell version 0.3.7 // 2020-09-10 // 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

```
*/ /*
```

```
*///
```