

# IETF 117 HTTP関連報告

ごとう

# 自己紹介

- 後藤 浩之 (グリー)
  - インフラ担当
- ISOC-JP インターネット標準化推進委員会
- 興味: Web, HTTP・QUIC関連



# HTTP 関連 WG

- HTTP
- HTTPAPI
- OHAI
- MASQUE
- WebTransport

# Draft: Structured Field Values Bis

『RFC 8941 - Structured Field Values for HTTP』の改定版

そもそも Structured Field Valuesは、HTTPヘッダに構造(List, Dictionary, Boolean)などを定義する仕様。

新しいヘッダを定義する際に、この構造に準ずることでパースアルゴリズム/実装を再利用できる

Example-List: sugar, tea, rum

Example-Dict: en="Applepie", da=:w4ZibGV0w6ZydGU=:

# Draft: Structured Field Values Bis

SFV Bis では新しく型が追加

- Date: 時間データをUnixタイムで表現
- Display String: 非ASCIIの表現

Example-Date: @1659578233

Example-DisplayString: %"This is intended for display to %C3%BCsers."

# Draft: Retrofit Structured Fields

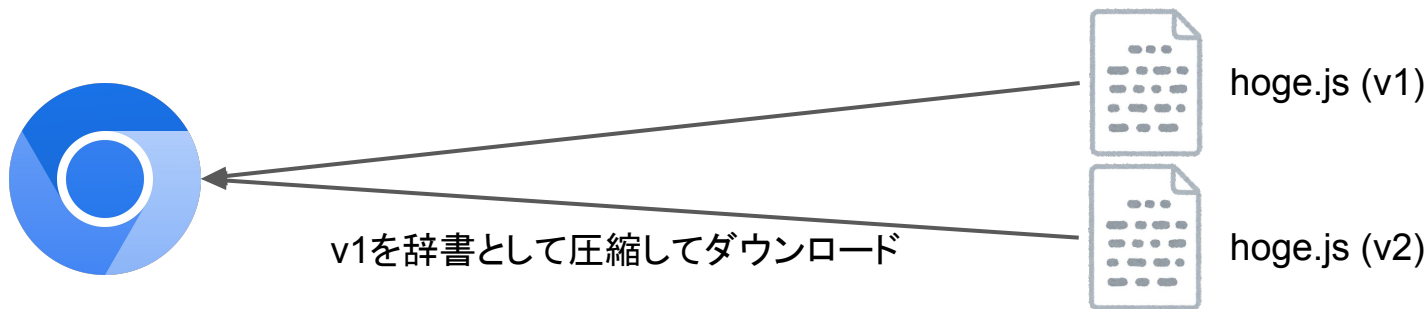
背景: Structured Fields使用自体は構造を定義するだけであり、そのアルゴリズムでパースしていいかはヘッダごとに指定される。

=> Retrofit Structured Fieldsではすでに定義済みのHTTPヘッダにStructured Fieldsを適応する draft

Field Name	Structured Type
Accept	List
Accept-Encoding	List
Accept-Language	List
Accept-Patch	List
Accept-Post	List
Accept-Ranges	List
Access-Control-Allow-Credentials	Item

# Draft: HTTP Compression Dictionary

- Chrome チームから提案されている、一度読み込んだリソースをdeflate dictionaryとして利用する
- ユースケース
  - 差分の少ないファイルとか
  - 『hoge.js を一部修正する』ケース。以前に読み込んだ修正前のファイルを deflate dictionaryとして保持しとけば、次の転送量を抑えられる
- 現在、W3C WICGで議論されてたり、Chromeのorigin trailが始まろうとしている



# Draft: Request-OTR Header

サーバ側からのレスポンスヘッダで

「Request-OTR: 1」

を付けて送ると、ブラウザはそのページの閲覧履歴(history, cookie, storage)を保存しないように振る舞う

DV相談サイト などセンシティブなサイトでの利用を想定



# WebSocket over HTTP/3 DT

- WebSocketを使うさい、クライアントはどちらで通信を開始するか決定できない
  - WebSocket over H3
  - WebSocket over H2

=> 追加のオーバーヘッドがかかる

- 案として
  - DNS (HTTPSレコード)
  - SETTINGS
  - ALPN
  - Option Request

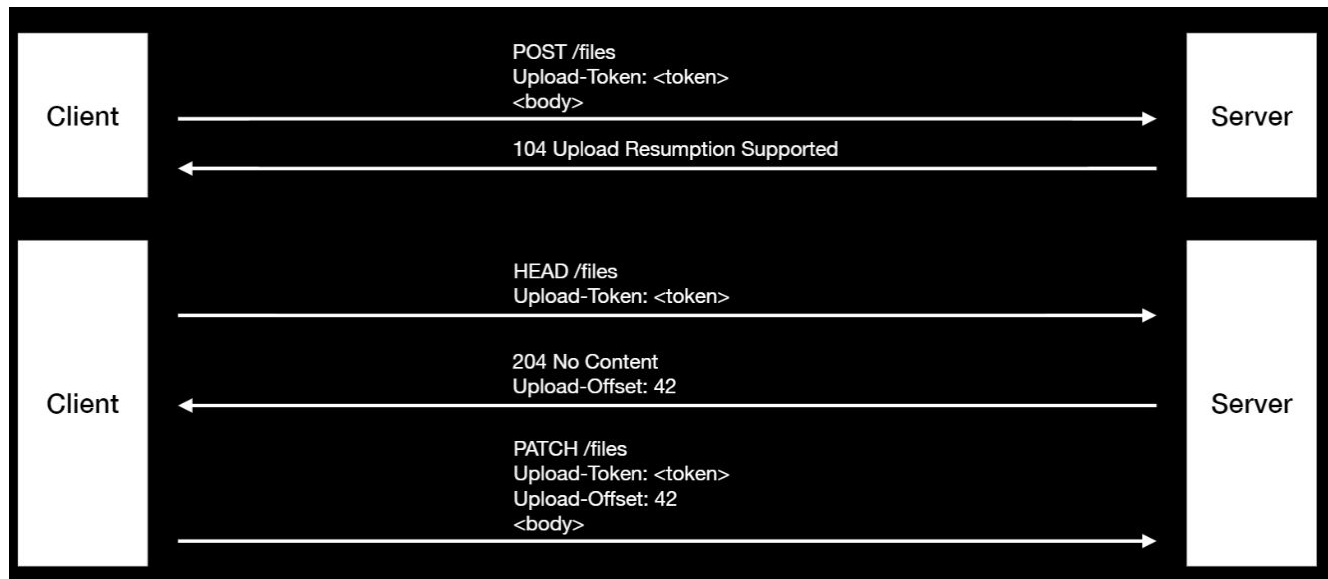
Property	Must / Nice	DNS	SETTING	ALPN	OPTIONS
Be able to establish websocket support without additional roundtrips	Must	4	8	6	2
The server needs to be able to roll back support for the feature without huge problems	Must	2	6	4	8
Support for WebSockets cannot be changed during an active connection	Must	8	8	8	2
Support needs to be Hop-by-Hop	Must	8	8	8	8
Logical consistency across HTTP/2 and HTTP/3	Nice	3	4	2	4
Determine status of support after a connection is established	Nice	4	4	4	1

Total

38	29	32	25
----	----	----	----

# Resumable Uploads

再開可能なHTTPアップロードの仕組み。  
再開用のトークンを払い出しておく。



# "Modern" HTTP Proxies

HTTPでは、CONNECTメソッドを使うことで通信をトンネリング出来る。

WebSocket over HTTP/2 で導入された拡張CONNECTメソッドでは、トンネリングするプロトコルを指定することが出来る(例: :protocol : websocket)

この仕組みを既存のTCP Proxyに取り込み、新しく :protocol : connect-tcp を定義する

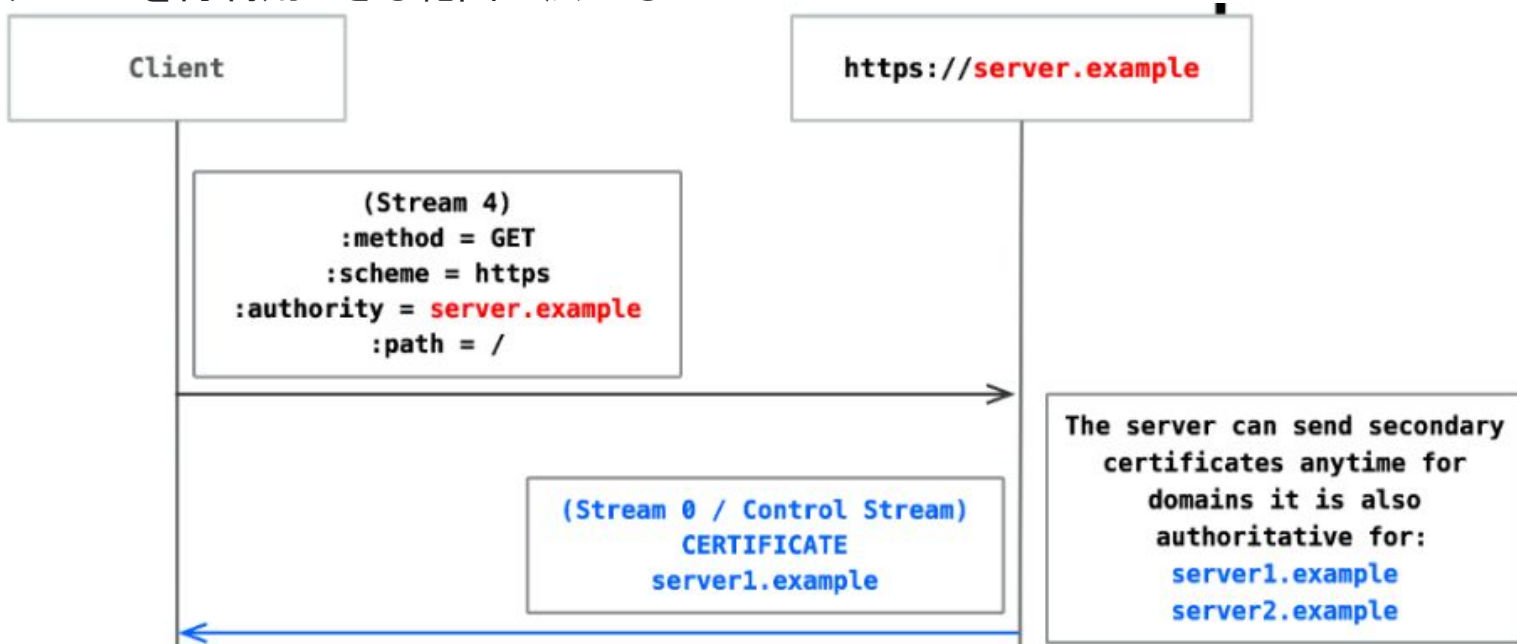
(PATHが有効に使えるようになる)

```
HEADERS
:method = CONNECT
:scheme = https
:authority = request-proxy.example
:path = /proxy?target_host=192.0.2.1,2001:db8::1&tcp_port=443
:protocol = connect-tcp
...
```

# Secondary Certificate

追加のサーバ証明書をHTTPレイヤで提供できるようにする提案

コネクションを再利用できる範囲が広がる



おわりに