HTTP入門

2020/6/17

本日のゴール

- ユーザの見えないところで行われていることを知る

  ブラウザとサーバで行われている通信の流れを知る

まずはHTTPとHTMLとはなにか

HTTP? HTML?

HTTP (Hyper Text Transfer Protocol)

htmlなどの情報をやり取りする際に使われる通信プロトコル

https://www.nic.ad.jp/ja/basics/terms/http.html

HTTP例

```
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:77.0) Gecko/20100101 Firefox/77.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

HTTP例

```
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:77.0) Gecko/20100101 Firefox/77.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

HTTP? HTML?

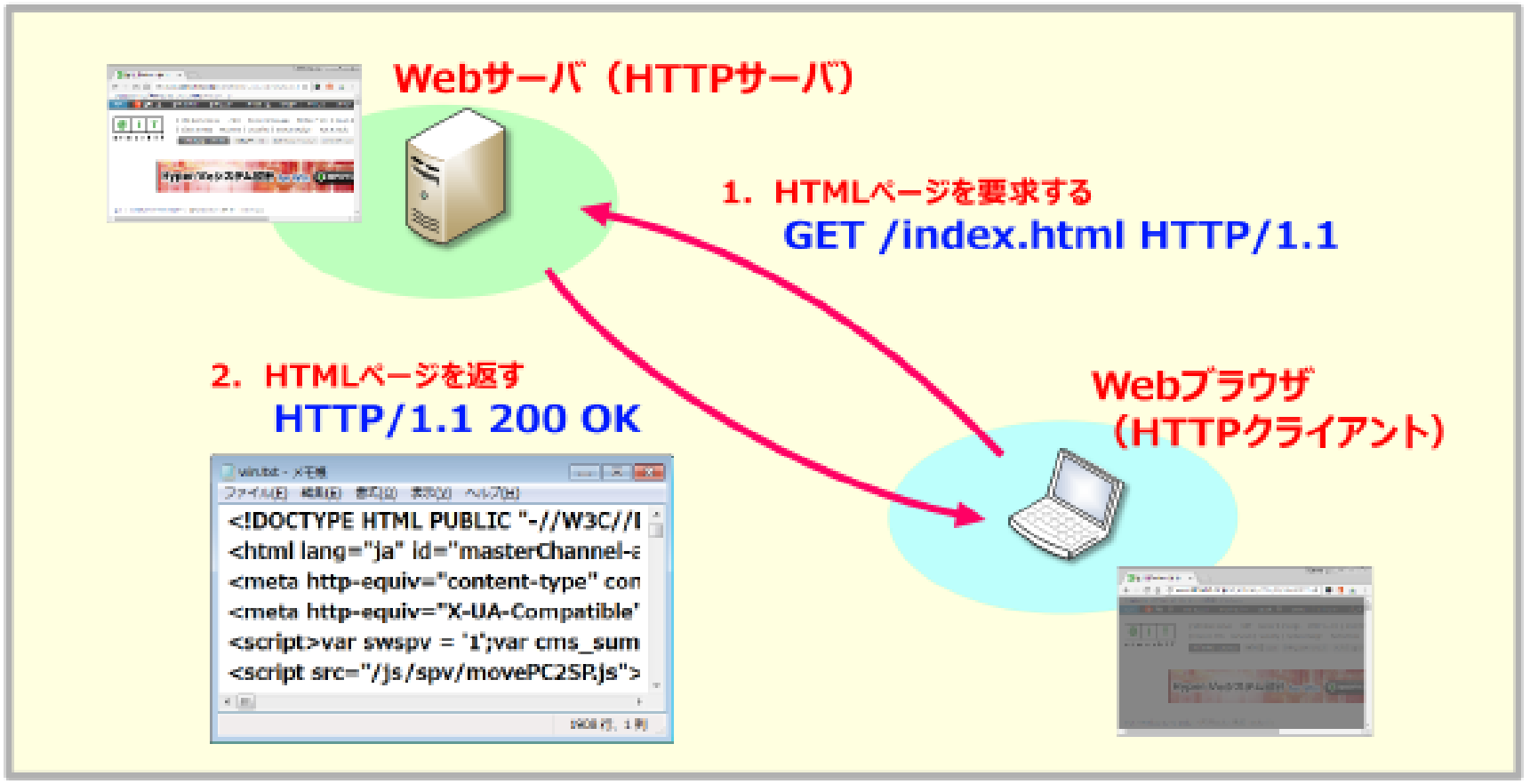
HTML(Hyper Text Markup Language)

ウェブページを作成するための言語

HTML例(抜粋)

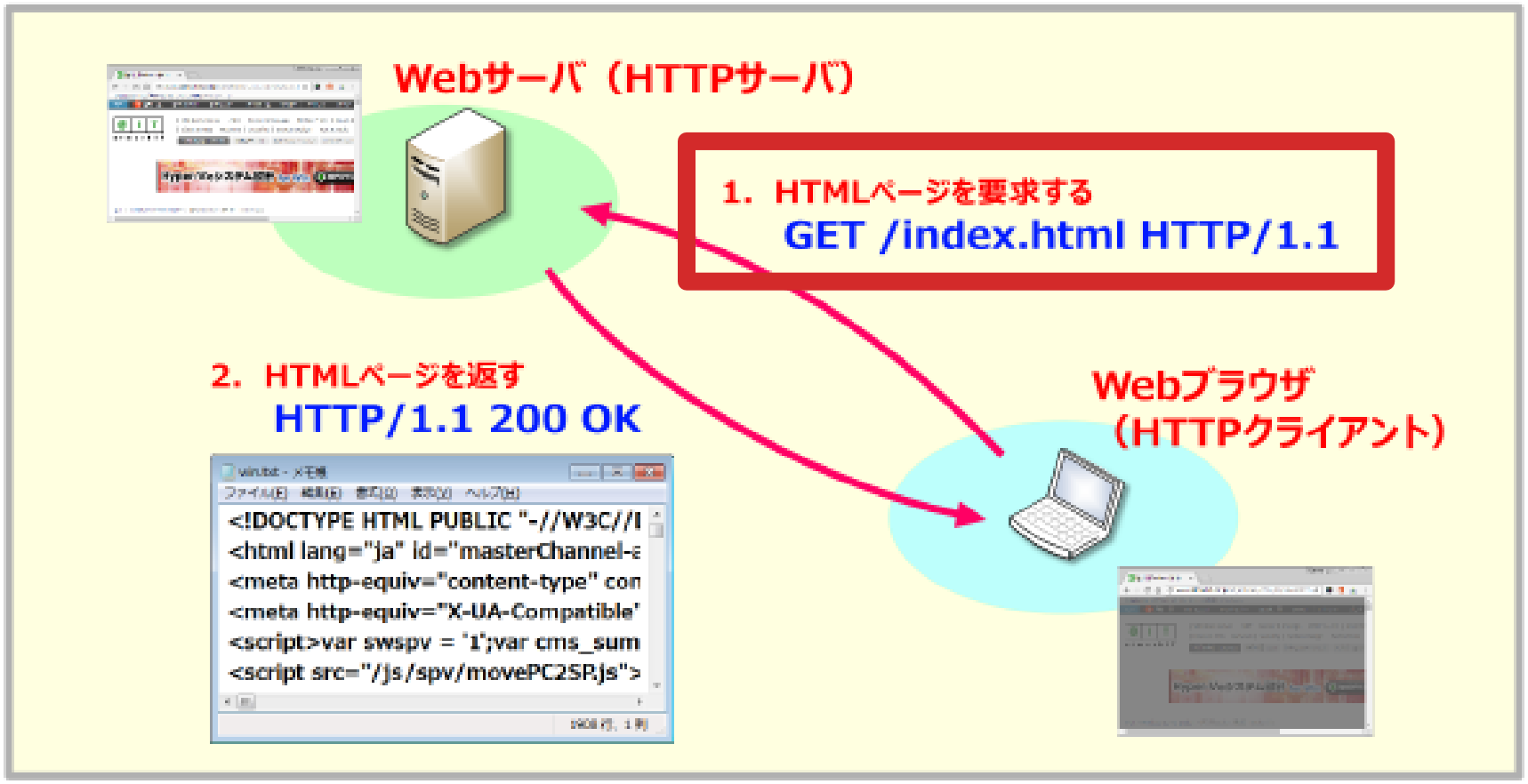
```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1">
    <title>SPY</title>
</head>
```

ブラウザで表示できるまで



https://www.atmarkit.co.jp/ait/articles/1703/29/news045.html

ブラウザで表示できるまで



https://www.atmarkit.co.jp/ait/articles/1703/29/news045.html

HTTPフィールド

http://www.kogures.com/hitoshi/webtext/nw-http/index.html

HTTPフィールド

http://www.kogures.com/hitoshi/webtext/nw-http/index.html

HTTPフィールド

http://www.kogures.com/hitoshi/webtext/nw-http/index.html

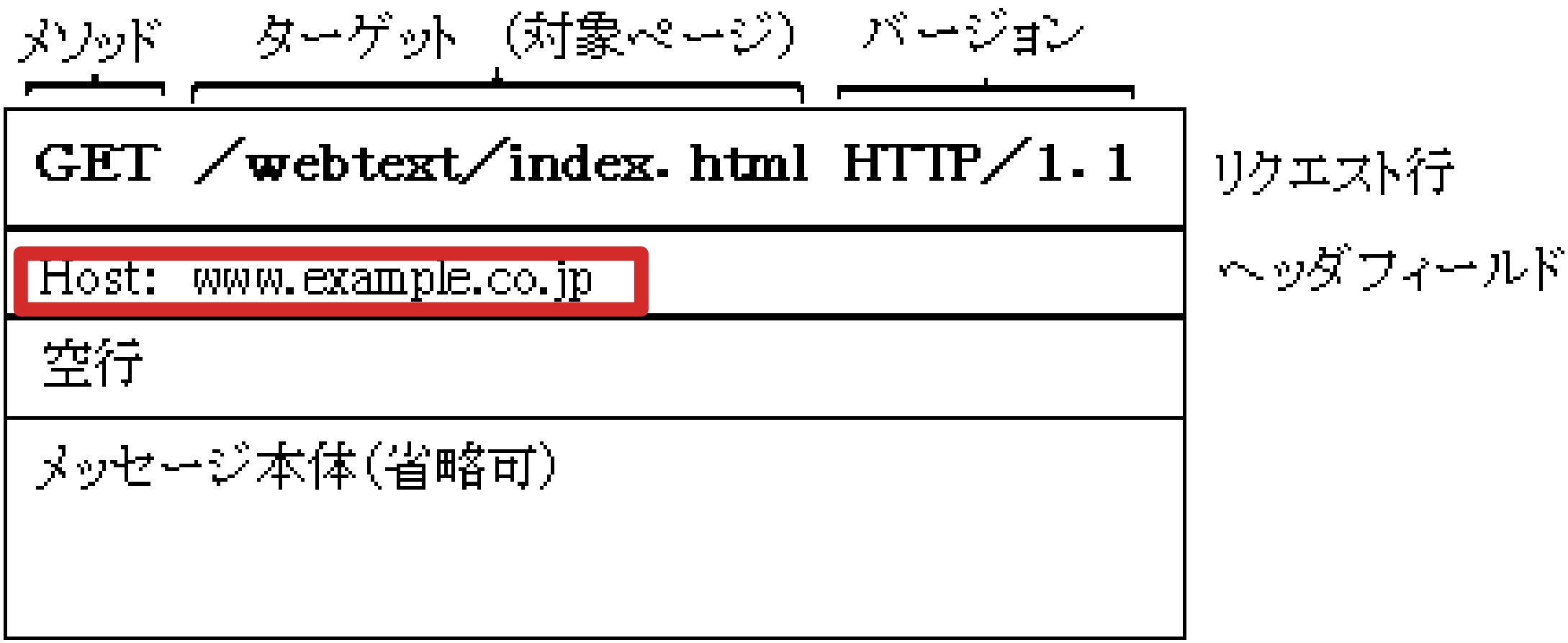
HTTPメソッド

- GET
  ヘッダとコンテンツを要求
- HEAD
  ヘッダのみを要求
- POST
  データを送信
- PUT
  ファイルをアップロード
- DELETE
  データの削除を要求する

HTTPメソッド

- OPTIONS
  使えるメソッドを要求

HTTPフィールド



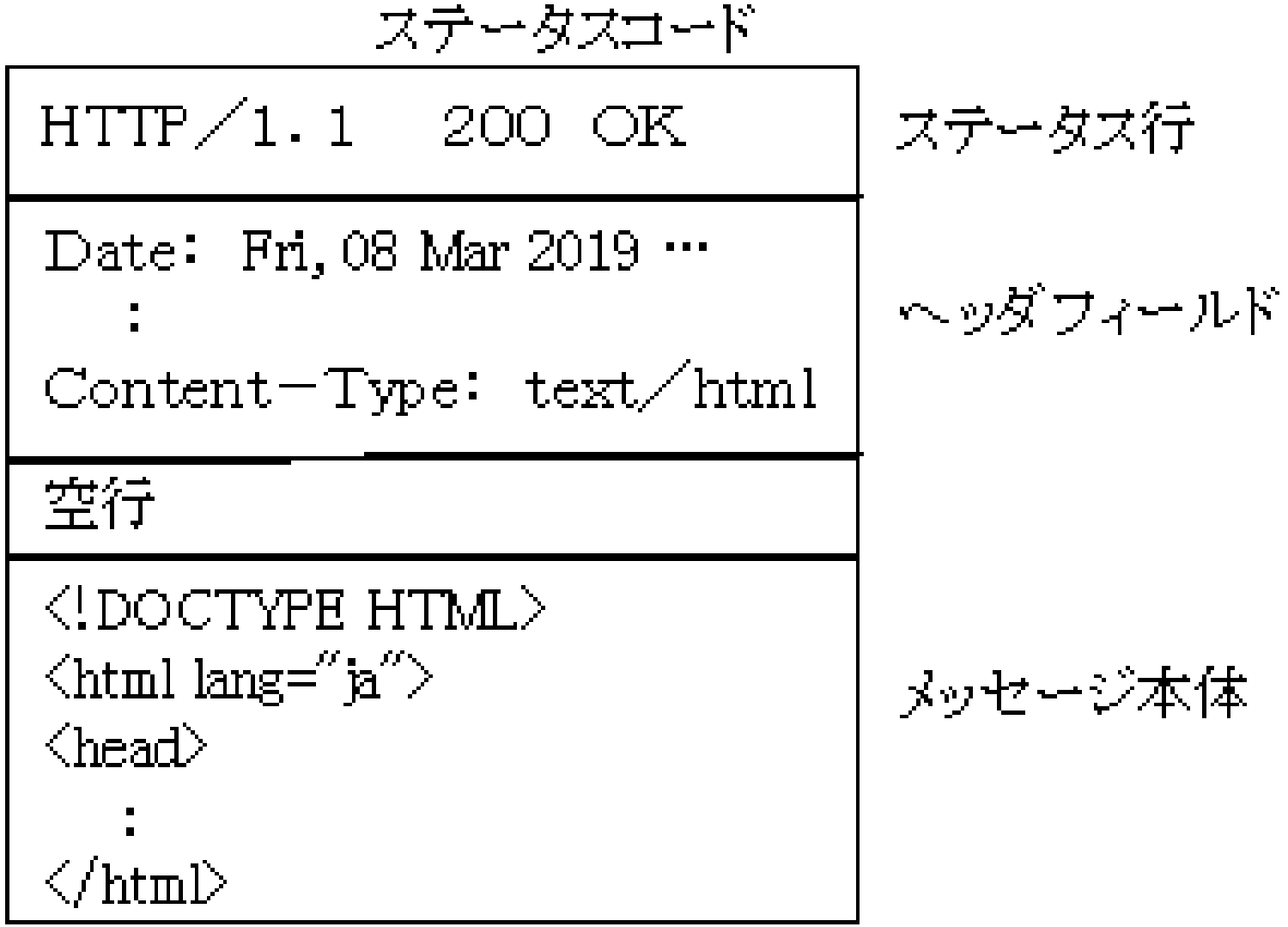
http://www.kogures.com/hitoshi/webtext/nw-http/index.html

HTTPヘッダ

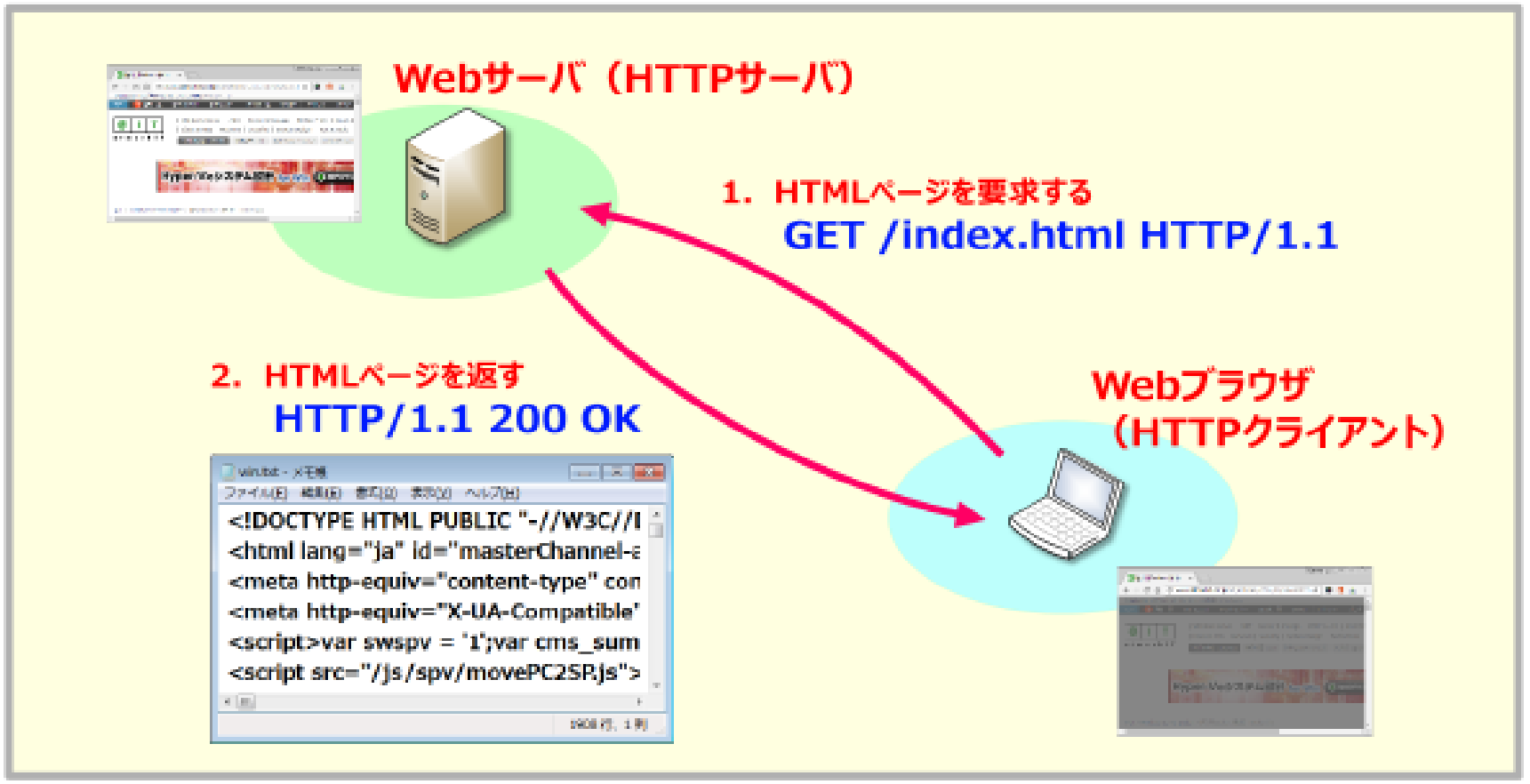
- User-Agent
  ブラウザの種類、OS情報
- Referer
  ハイパーリンクをクリックしたURL
- Accept、Accept-Language、Accept-Encoding、Accept-Charset
  どのようなデータを受け取りたいのか、言語、文字コード、画像の種類などの情報。

https://www.itra.co.jp/webmedia/http-header.html

HTTPレスポンス

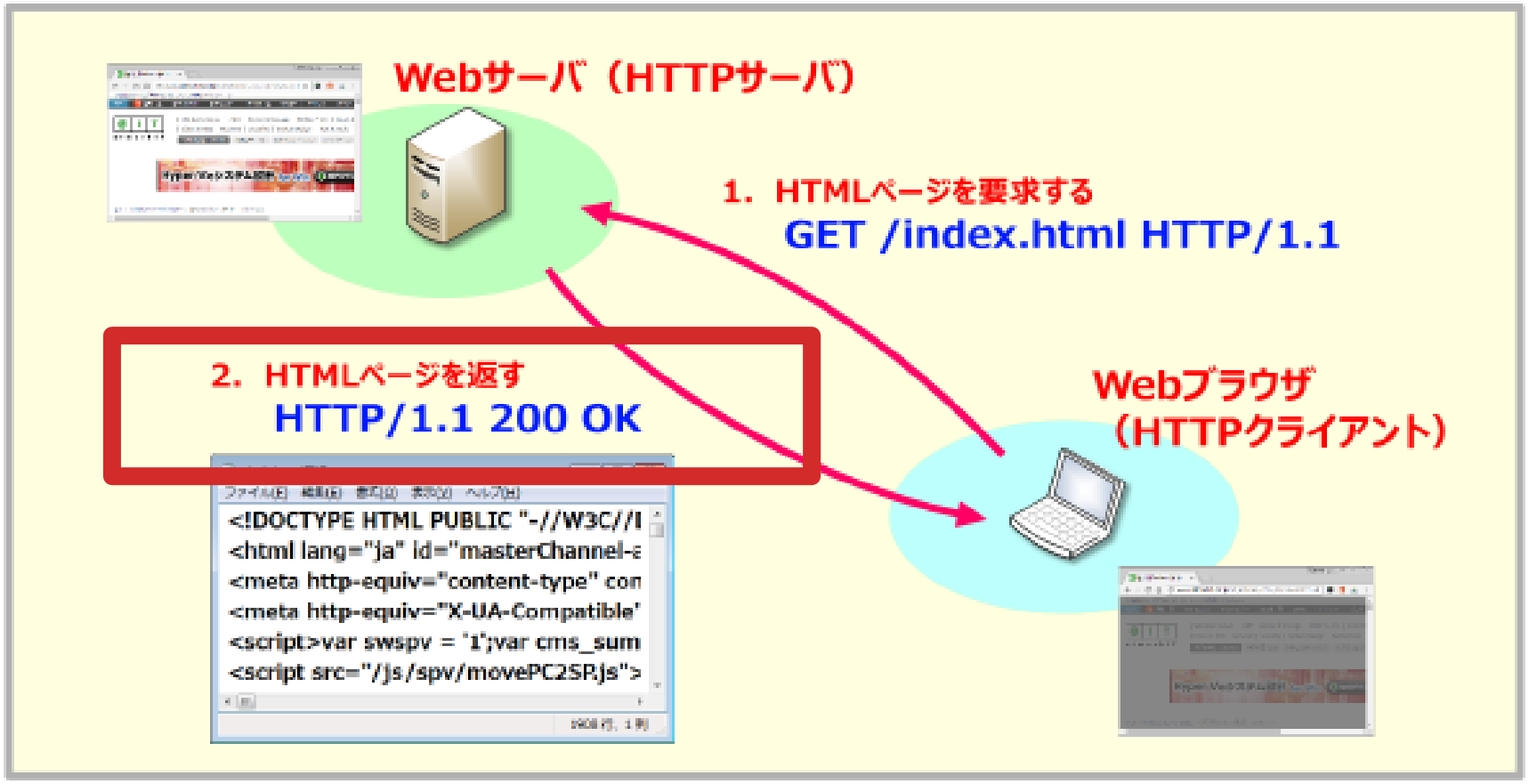

ステータスコード
HTTP／1.1　200 OK
ステータス行
Date: Fri, 08 Mar 2019 …
:
Content－Type: text／html
ヘッダフィールド
空行
<!DOCTYPE HTML>
<html lang="ja">
<head>
:
</html>
メッセージ本体

ブラウザで表示できるまで



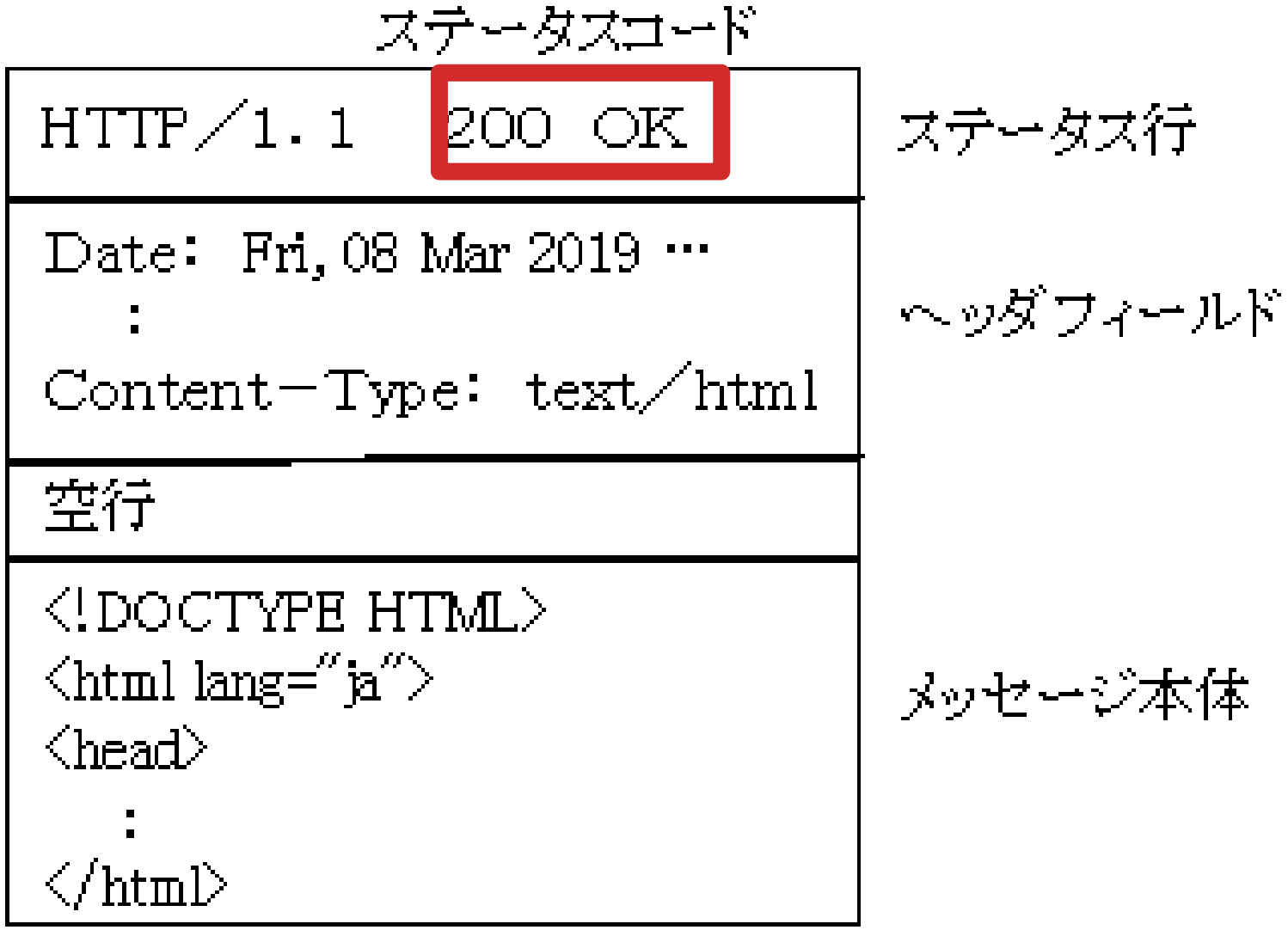
https://www.atmarkit.co.jp/ait/articles/1703/29/news045.html

ブラウザで表示できるまで



https://www.atmarkit.co.jp/ait/articles/1703/29/news045.html

HTTPレスポンス


ステータスコード
HTTP／1.1 200 OK
ステータス行
Date: Fri, 08 Mar 2019 …
:
Content-Type: text/html
ヘッダフィールド
空行
<!DOCTYPE HTML>
<html lang="ja">
<head>
:
</html>
メッセージ本体

HTTPステータスコード

- 100番台
  処理中の情報伝達

- 200番台
  成功時のレスポンス

- 300番台
  サーバからのクライアントへの命令

HTTPステータスコード

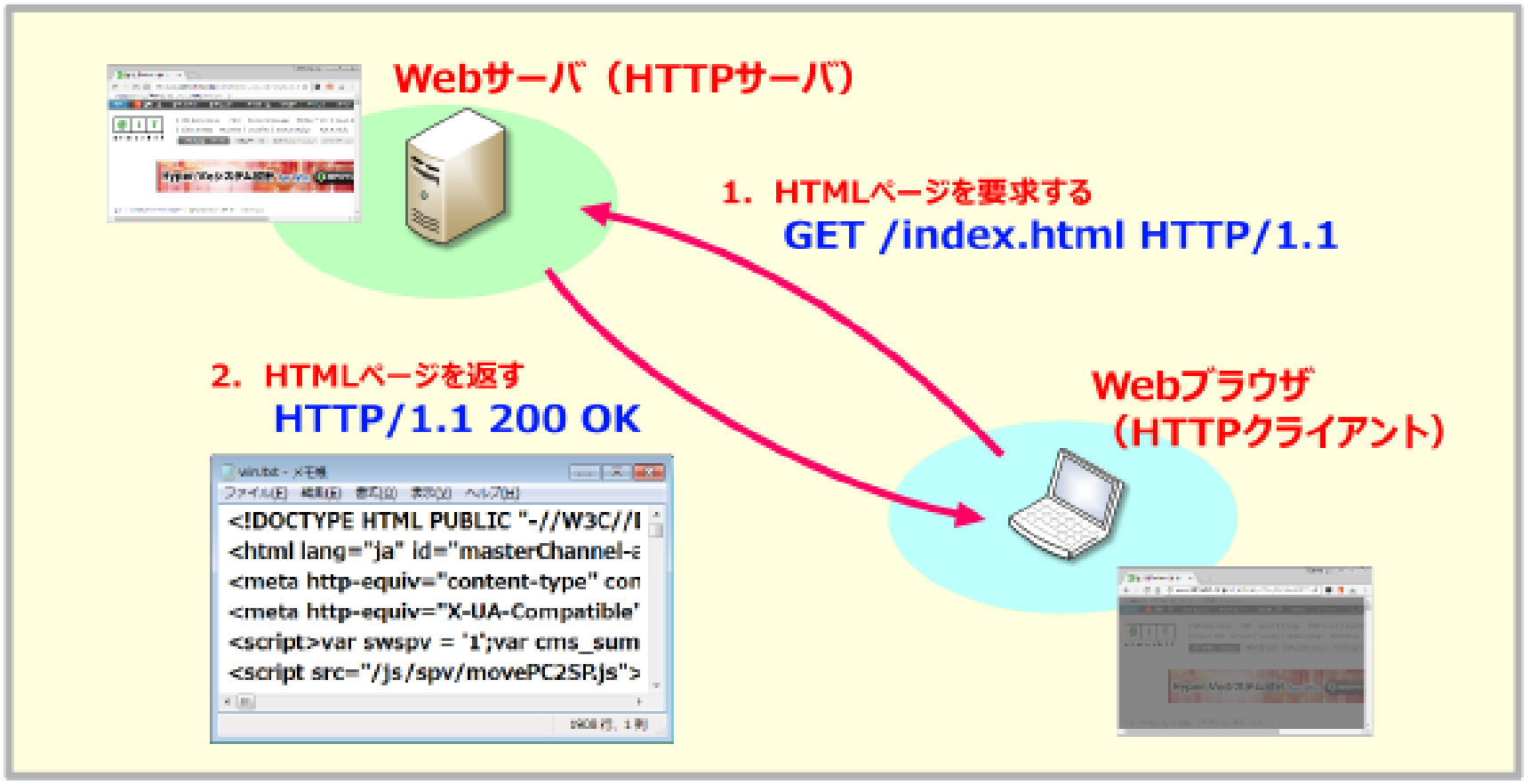
- 400番台
  クライアントからのリクエストでエラー

- 500番台
  サーバ内部でエラーが発生

よくみるHTTPステータスコード

- 200 OK
  要求の成功
- 404 not found
  webページが存在しない
- 301 moved permanently
  リダイレクト
- 500 Internal Server Error
  サーバ内部のエラー発生

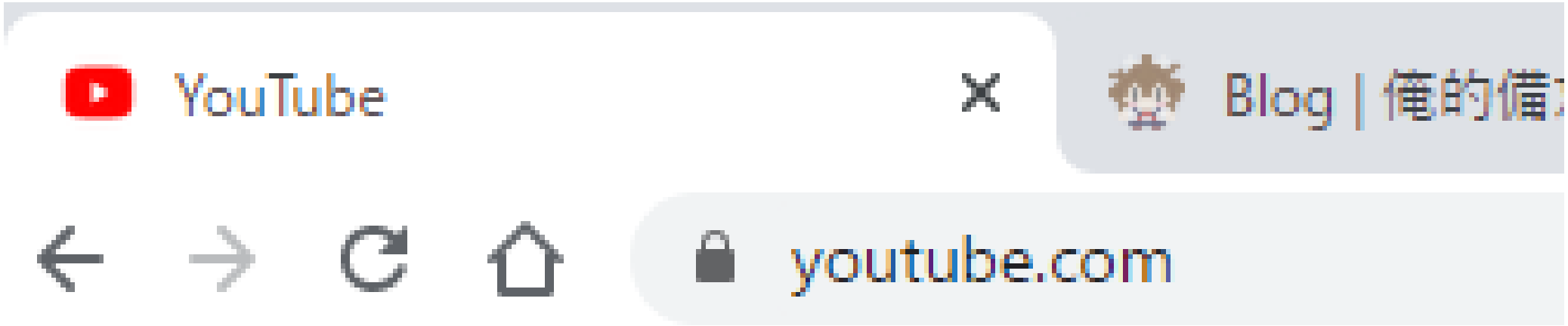
ブラウザで表示できるまで



https://www.atmarkit.co.jp/ait/articles/1703/29/news045.html

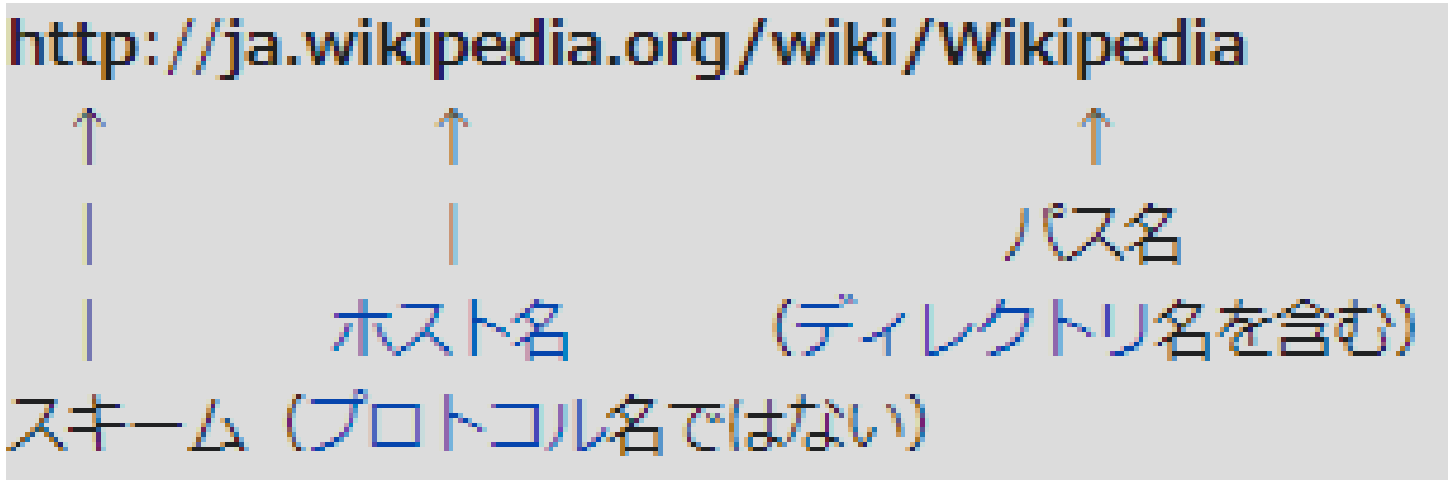
ブラウザでwebを閲覧するには

- ブラウザでURLを入力する必要がある



URLについて

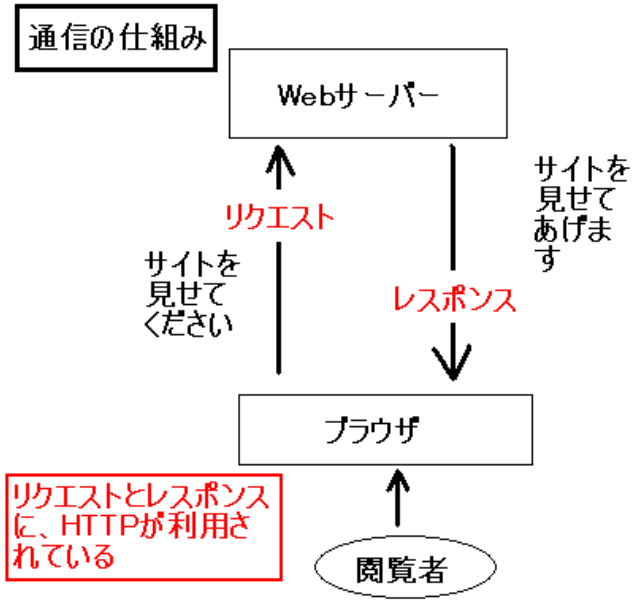
- リソースの場所を特定する「住所」のようなもの



https://ja.wikipedia.org/wiki/Uniform_Resource_Locator

Webコンテンツの種類

- 静的コンテンツ：要求されたデータをブラウザに返すのみ
- 動的コンテンツ：データを受け取り、
  それを処理したうえでユーザーに返す



- http://juen-cs.dl.juen.ac.jp/html/www/005/

サーバにデータを送信する方法

1. URLに添付
2. リクエストボディに添付

サーバにデータを送信する方法

1. URLに添付
2. リクエストボディに添付

1. URLのクエリパラメータ

サーバのプログラムに渡す情報を記述

http://example.com**?key=value&key2=value2**

使用時

- GETメソッドでデータ送信するとき

サーバにデータを送信する方法

1. URLに添付
2. リクエストボディに添付

2. リクエストボディに記述

POSTでデータの送信

- リクエストボディにデータを記述する

```
POST /post.php HTTP/1.1
Host: http2020.herokuapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64,
Accept: text/html,application/xhtml+xml,application/x
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 6
Origin: https://http2020.herokuapp.com
Connection: close
Referer: https://http2020.herokuapp.com/post.php
Upgrade-Insecure-Requests: 1

post=a
```

実際に、やっていきます！

ハンズオン

- https://http2020.herokuapp.com/

やること

- ブラウザからの接続と
  curlコマンドからの接続を比較してイメージをつかむ

Curlコマンドとは

- ファイルを送信または受信するコマンドラインツール

```
curl www.example.com
```

- 基本的にURLを打ち込めば動作する

Curlオプション紹介

HTTPメソッドの指定（-X）

$ curl -X **POST** http://対象のURL

＃POSTリクエストとしてフォームを送信する(-d,--data PARAM)

$ curl -d **“key=value”**（URL)

レスポンスヘッダを出力に含ませる –i ,--include

$ curl **–i** (URL)

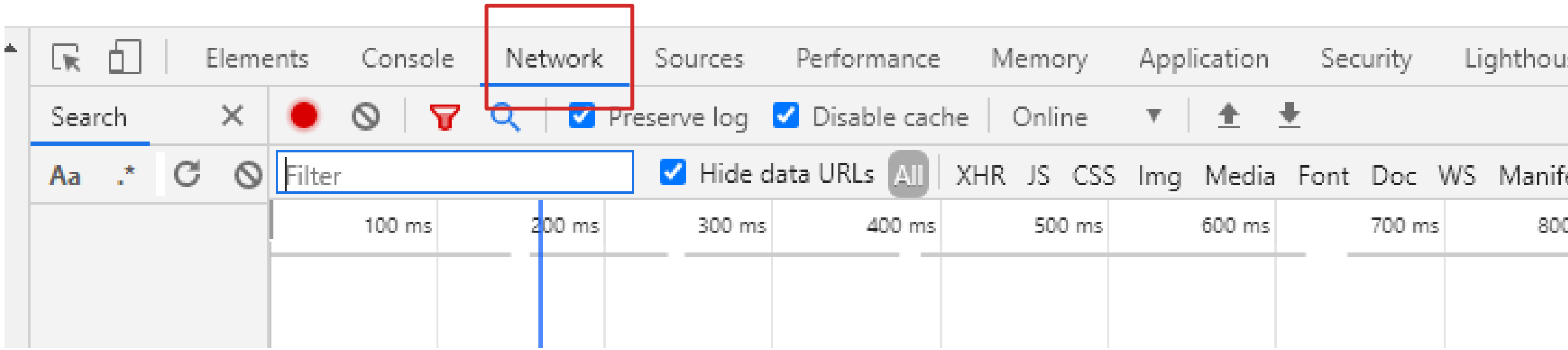
通信のデバック –v,--verbose

$ curl **–v** (URL)

通信のデバック　ボディも含める　–trace-ascii

$ curl (URL) -d "post=datadesu" **--trace-ascii (出力先)**

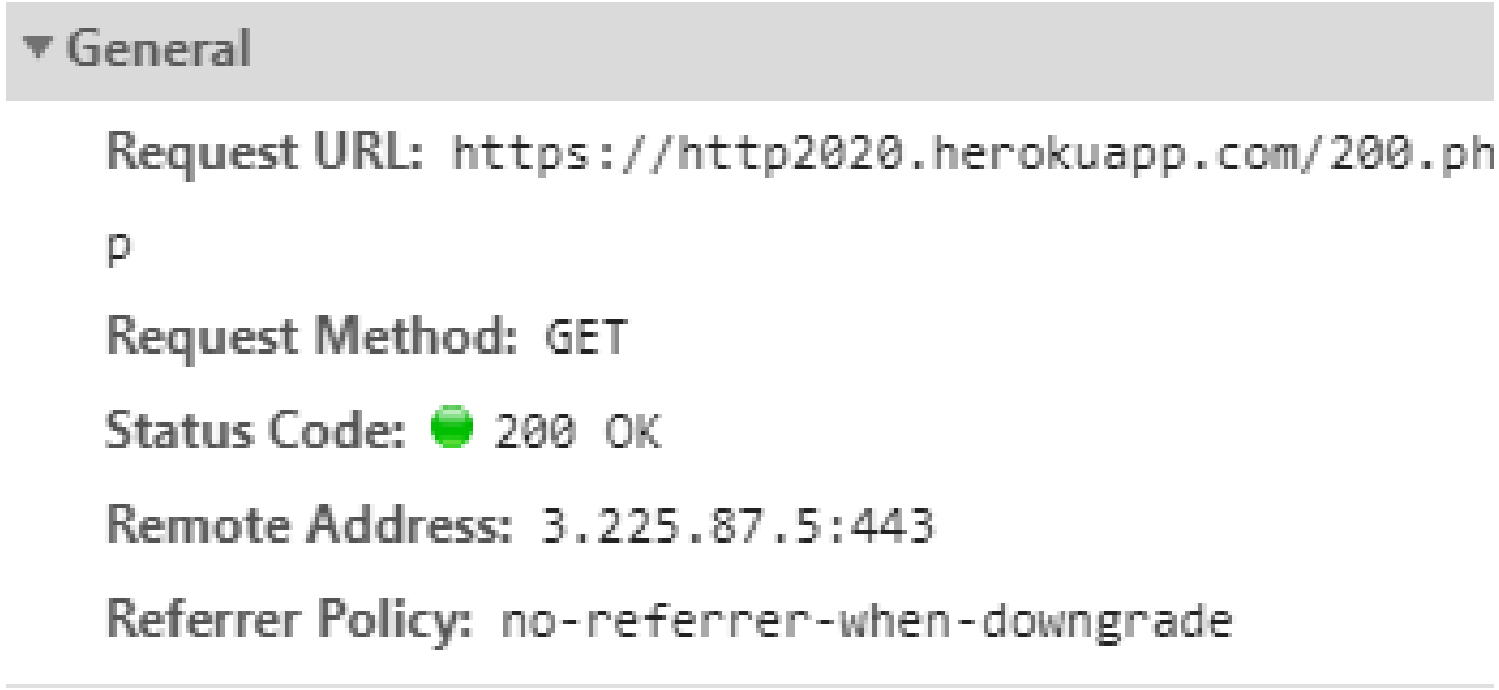
200.php
get.php
404.php
301.php
referer.php
useragent.php
README.md
post.php


Elements
Console
Network
Sources
Performance
Memory
Application
Security
Search
Preserve log
Disable cache
Online
Aa
.*
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms

200 OK

curl http://http2020.herokuapp.com/200.php -v

- ブラウザでのレスポンス(F12->network)

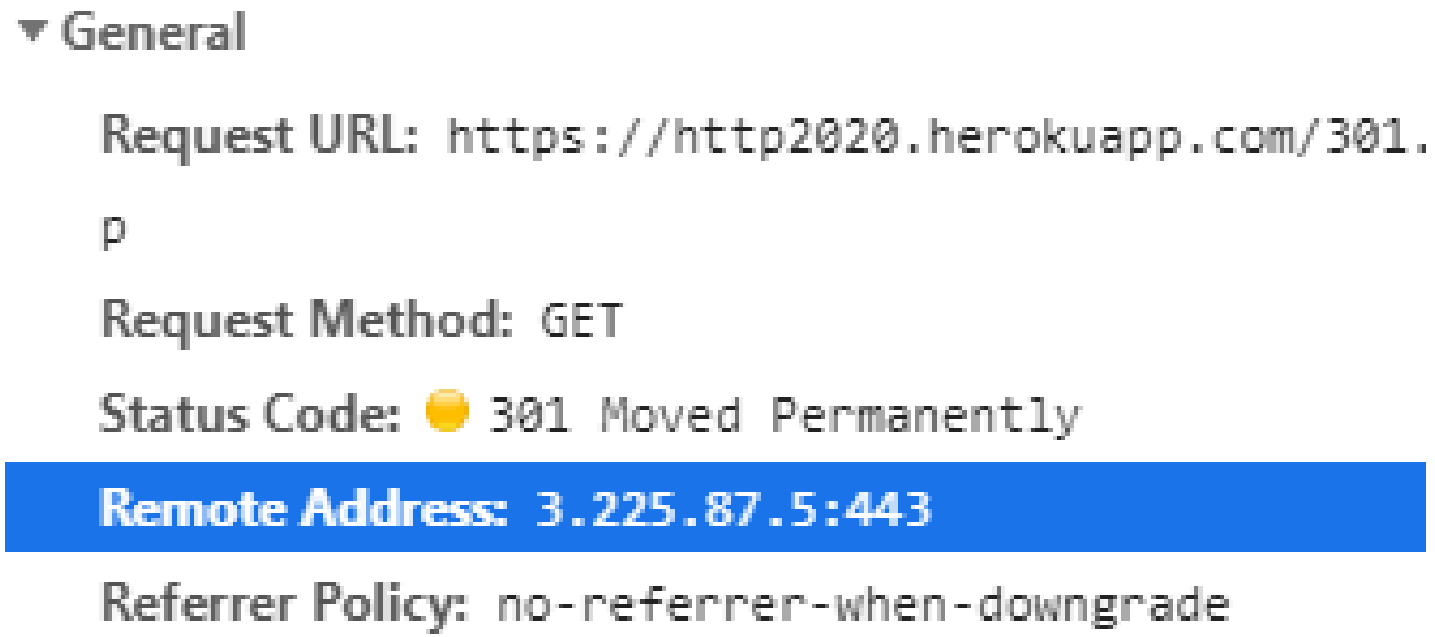


- curl --verbose http://http2020.herokuapp.com/200.php

< HTTP/1.1 200 OK
< Connection: keep-alive
< Date: Sun, 14 Jun 2020 16:55:26 GMT
< Server: Apache
< Transfer-Encoding: chunked
< Content-Type: text/html; charset=UTF-8
< Via: 1.1 vegur

301 moved parmanently

- curl http://http2020.herokuapp.com/301.php -v



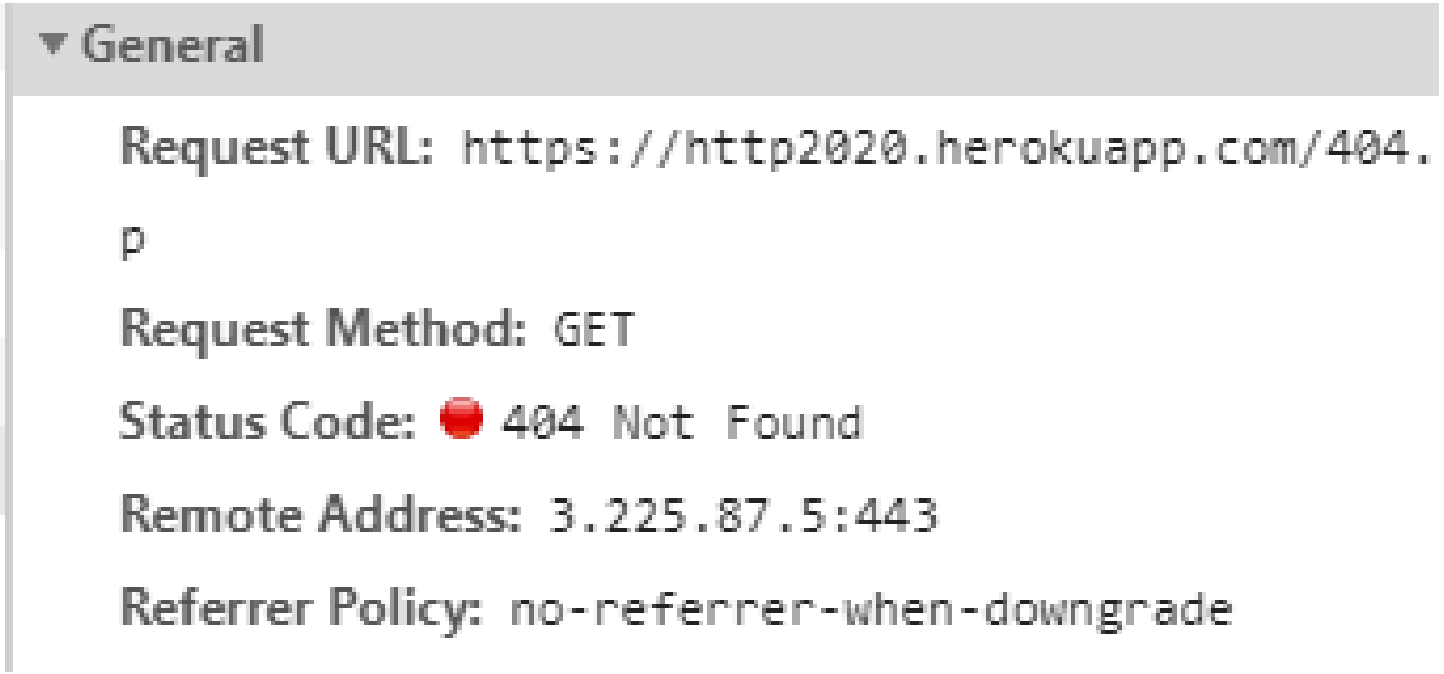
Location　ヘッダ

- リダイレクト先の URL を示します

- https://developer.mozilla.org/ja/docs/Web/HTTP/Headers/Location#:~:text=Location%20%E3%83%98%E3%83%83%E3%83%80%E3%83%BC%E3%81%AF%E3%80%81%E3%83%AA%E3%83%80%E3%82%A4%E3%83%AC%E3%82%AF%E3%83%88%E3%81%AE,%E3%81%AAURL%E3%82%92%E6%8C%87%E3%81%97%E3%81%BE%E3%81%99%E3%80%82

404 notfound

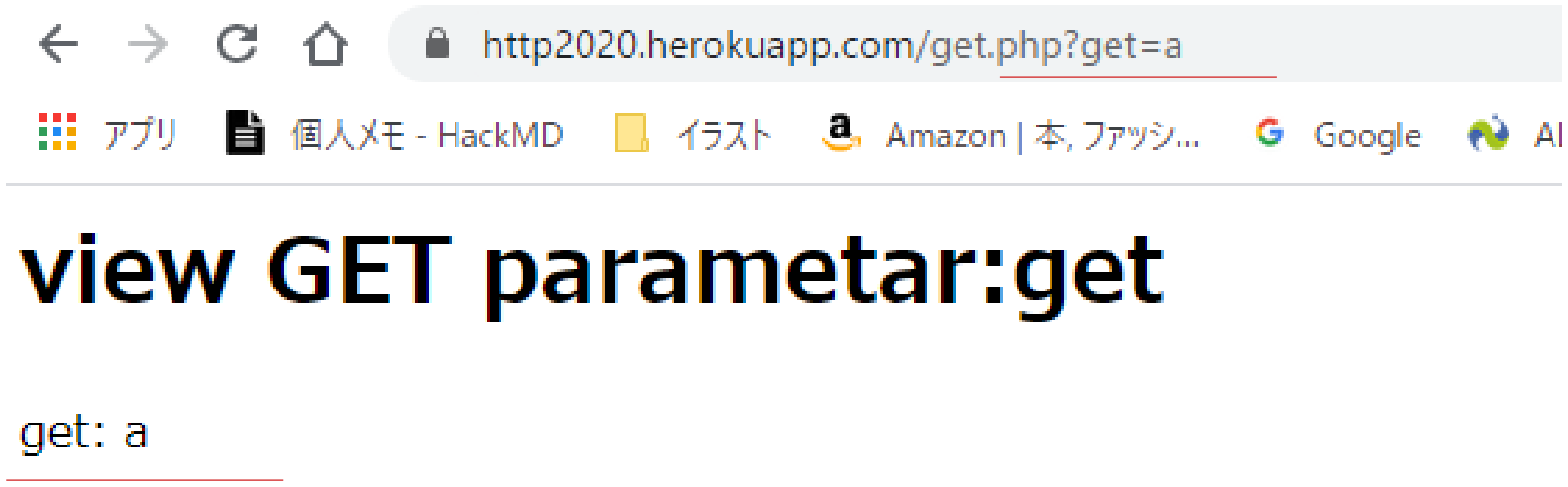
curl http://http2020.herokuapp.com/404.php -v



GETパラメータ

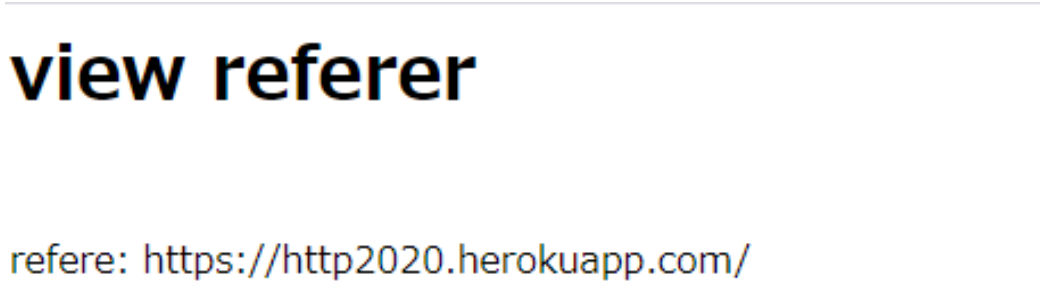
- curl http://http2020.herokuapp.com/get.php?get=a -v

GETパラメータで送信した内容が表示されます
Key名:get



リファラ

- curl https://http2020.herokuapp.com/referer.php -H "Referer:https:google.com"



ユーザエージェント

- curl http://http2020.herokuapp.com/useragent.php -v -A "ta"

view get useragent

useragent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/83.0.4103.97 Safari/537.36

最後にOPTIONSメソッド実行

$curl -X OPTIONS -i https://curl.haxx.se

- Curlのサイトに対してoptionsメソッドを飛ばす
- Ngixではoptionsメソッドはデフォルトで許可されていない

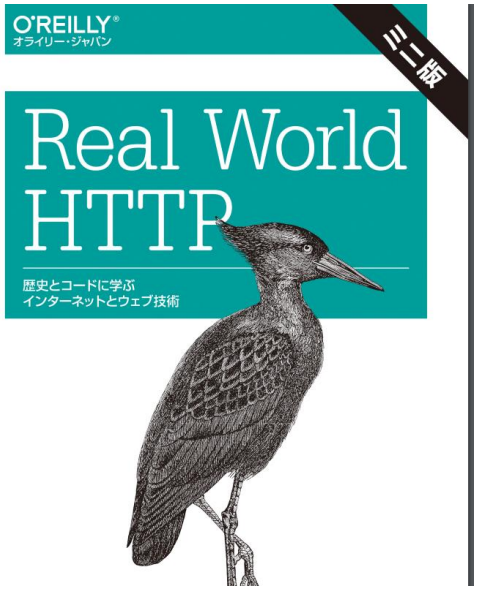
ソースコード

- 今回のハンズオン環境
- https://github.com/grapesota/HTTP2020_6

おすすめ関連資料

- Real world http

  https://www.oreilly.co.jp/books/9784873118789/



- Mozilla

  https://developer.mozilla.org/ja/docs/Web/HTTP