

Webの基礎

目的

- HTTPについて(研修よりは)詳しく知ってもらう
- Webアプリケーションがどう動いているか知ってもらう

対象

- 研修を終えてこれからWebアプリケーション開発に従事する人
- 2課の人

HTTPは何の略？

Hypertext Transfer Protocol

"すごい文書"を転送するための規約

HTTPとは

- リクエスト-レスポンス型プロトコル
クライアントからサーバに向けてリクエストを送付し、サーバがクライアントにレスポンスを返す
- テキストベースで情報をやり取り
- 当初は文書をやり取りするためのプロトコルだったが、現在では画像や動画をはじめとしたさまざまなデータのやり取りに利用される

バージョン

- HTTP/0.9, 1.0
現在は使われていない
- HTTP/1.1
現在主流のバージョン
- HTTP/2
後方互換性を維持したまま高速化
- HTTP/3
後方互換性を維持したまま高速化

> 機能としてはHTTP/1.1をわかっていれば問題ない

以後、このスライドではHTTP/1.1について説明する

HTTPリクエストの中身

http://hoge.jp/fuga にリクエストを投げる例

```
POST /fuga HTTP/1.1
Host: hoge.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0~略
Content-Type: application/json; charset=utf-8

{
  "name": "太郎",
  "race": "dog"
}
```

↑のようなテキストデータがサーバに送信される

1行目：リクエストライン。左からHTTPメソッド、URI、使用するHTTPのバージョン

2行目以降から空行まで：リクエストヘッダ

空行以降: リクエストボディ


```
POST /fuga HTTP/1.1
Host: hoge.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0~略
Content-Type: application/json; charset=utf-8
```

```
{
  "name": "太郎",
  "race": "dog"
}
```

HTTPメソッド

- そのリクエストがサーバ上のリソースに対し何をしたいかを意思表示する

メソッド一覧

- GET
リソースの参照
- POST
リソースの登録/更新
- PUT
リソースの登録/入れ替え
- DELETE
リソースの削除
- その他
HEAD, OPTIONS, TRACE, CONNECT

現実で使われているメソッドはほぼGET/POSTのみ

- 参照はGET、サーバ上のリソースに変更を加える場合はPOST
-> REST
- HTTPの規約に厳密に従って実装されたもの
-> RESTful

※めんどくさい、トランザクションの関係上使い分けができないなどの理由でRESTfulな実装はほぼ存在しない

-> GET, POSTのみ本研修で解説する

GET

- サーバ上のリソースを参照する場合のメソッド
- リクエストにボディ部を持たない

```
GET /fuga HTTP/1.1
Host: hoge.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0~略
```

- パラメータを渡す場合は、URIに埋め込む
name=太郎、race=dogのパラメータを渡す場合

```
GET /fuga?name=太郎&race=dog HTTP/1.1
Host: hoge.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0~略
```

POST

- サーバ上のリソースに変更を加える場合のメソッド
- パラメータを渡す場合はボディに埋め込む
- リクエストボディの形式はContent-Typeヘッダで宣言する

```
POST /fuga HTTP/1.1
Host: hoge.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0~略
Content-Type: application/json; charset=utf-8
```

```
{
  "name": "太郎",
  "race": "dog"
}
```

HTTPレスポンス

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close

<html>
  <body>Hello World</body>
</html>
```

- ↑のようなテキストデータがサーバから返却される
- 1行目：ステータスライン。プロトコルバージョン、ステータスコード、メッセージが入る
2行目以降から空行まで：レスポンスヘッダ
空行以降: レスポンスボディ

ステータスコード

ステータスコードは100~500番台の数値で、3桁目毎に分類わけされている

コード	意味
100番台	情報を返す
200番台	リクエストに成功
300番台	追加の処理が必要
400番台	クライアントエラー
500番台	サーバエラー

よく使うステータスコード

コード	意味
200	OK
301	Moved Permanently (リソースが別URIに移動したことを知らせる)
303	Found (リダイレクトに用いられる)
304	Not Modified
400	Bad Request 不正なリクエスト
401	Unauthorized 認証エラー
403	Forbidden 認可エラー
404	Not Found リソースがない