

**HTTPステータスコード
完全に理解した**

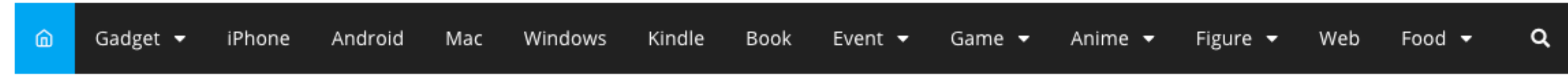


@unsoluble_sugar

TOPGATE, Inc.
Leader/Engineer

TIME TO LIVE FOREVER

いつかどこかの、誰かの未来のために。

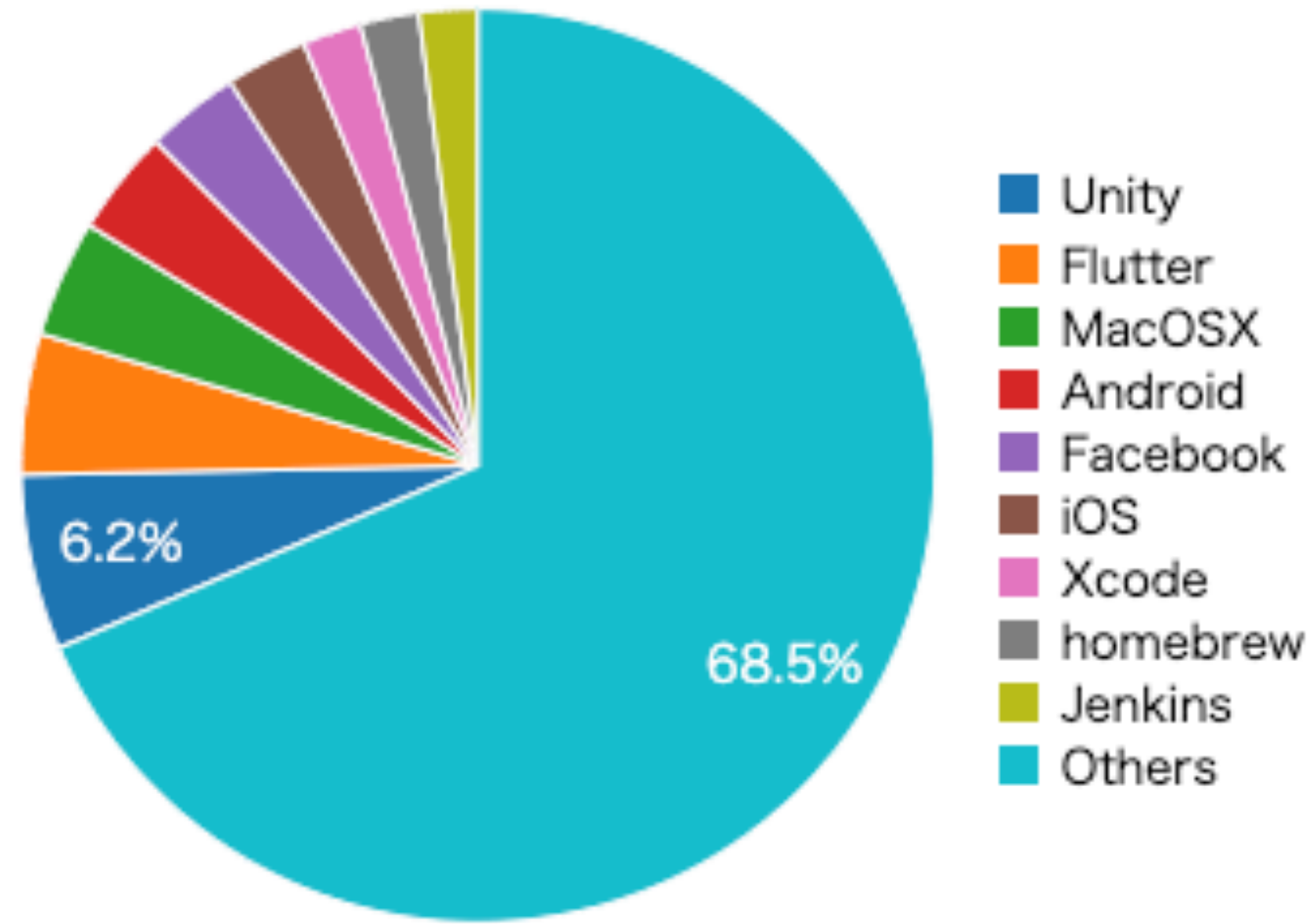


Breaking News Twitterで噂の「〇〇企画 ご当選確定者」リストの実態に迫る

Amazon



<https://unsolublesugar.com>



83

Items

1729

Contribution


59

Followers

https://qiita.com/unsoluble_sugar



HTTP ステータスコード
完全に理解していますか？

 聞いたことある

 完全に理解した

 何もわからない

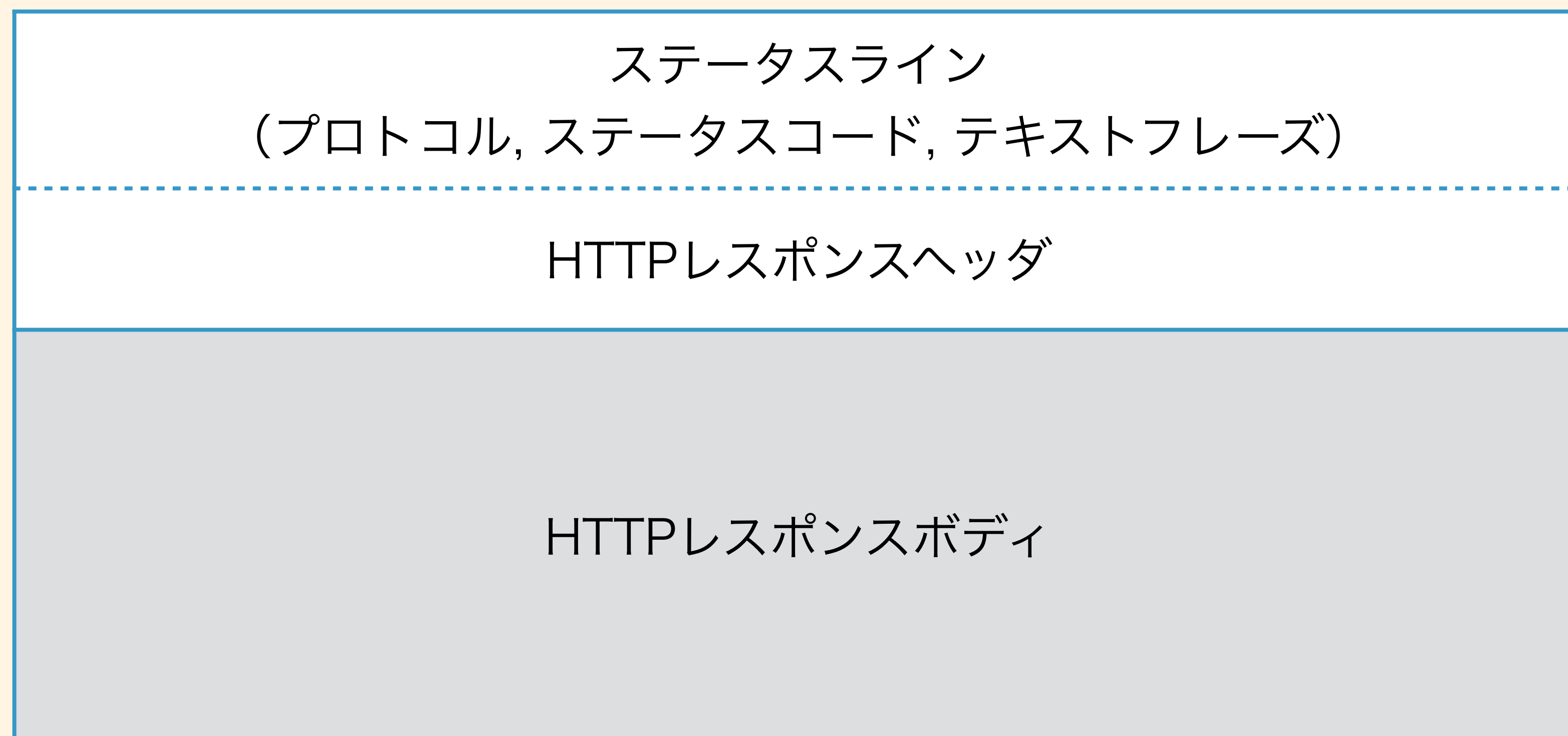
 チョットデキル

HTTPステータスコードとは

- 厳密にはHTTPレスポンスのステータスコード
- サーバからのレスポンスの意味を表す3桁の数字コードのこと
- 特定のHTTPリクエストが正常に完了したかを示す

HTTPレスポンス

レスポンスメッセージは大きく 3 種類の情報に分けられる



ステータスコードはヘッダ1行目

HTTP/1.1 **200** OK

Date: Sat, 22 Feb 2020 08:09:30 GMT

Content-Type: application/json; charset=utf-8

Connection: keep-alive

Server: nginx

ETag: "e287c729017cc9785487098b6b103af6"

Cache-Control: max-age=0, private, must-revalidate

X-UA-Compatible: IE=Edge,chrome=1

X-Runtime: 0.003487

ステータスコードは5つのクラスに分類

100番台：🤖情報レスポンス

200番台：😊成功レスポンス

300番台：😏リダイレクト

400番台：😞クライアントエラー

500番台：😇サーバエラー



100番台
情報レスポンス

100番台

- 処理が継続していることを示す
- クライアントはそのままリクエストを継続するか、サーバーの指示に従ってプロトコルをアップデートして再送信する
- HTTP/1.0規約に1xx番号の定義がないため、100番台はあまり利用されない

100番台

- 100 Continue
サーバがリクエストの受信をし拒否していない。クライアントはリクエストを継続できる
- 101 Switching Protocols
サーバはプロトコルの切り替えを要求している



200番台

成功レスポンス

200 OK

- 成功時に返すステータスコードとして最も多用される
- リクエストした処理が成功、指定したデータの取得に成功
- GET：ボディにリソースが含まれる
- PUT, POST：ボディに処理結果が含まれる

201 Created

- POST, PUT : リクエストが成功しリソースが作成された
- POSTの場合はレスポンスのLocationヘッダにURIが入る
- ユーザー新規登録、画像アップロード、DBのテーブル追加など
- ボディには新しく作成したリソースを入れることが多いが、特に何も入れなくても良い

202 Accepted

- リクエストは受理されたが、まだ処理が完了していない
- ファイル形式の変換やプッシュ通知のリクエスト、バッチ処理など、サーバ側で非同期に行う処理がある場合に返す

204 No Content

- レスポンスボディが空のときに返す
- DELETEでデータ削除を行った際に返す
- POSTでフォーム内容を送信したが画面更新がない場合
- PUTやPATCHで正しくデータ更新された場合
- DELETE以外ではあまり使うべきではないという意見もある

205 Reset Content

- ユーザーエージェントの画面をリセットする場合に返される
- 204と同じくレスポンスボディはなし



300番台
リダイレクト

300 Multiple Choices

- リクエストに対して複数のレスポンスがあることを示す
- 選択肢へリンクするHTMLが提示され、ユーザーエージェントやユーザーはそれらからひとつを選択する

301 Moved Permanently

- リクエストされたリソースのURLが恒久的に変更された
- Webサイトの移転や、HTTPからHTTPSへのリダイレクトなど
- Locationヘッダに移動先のURLが示される

302 Found

- リクエストされたリソースのURLが一時的に変更された
- Locationヘッダに移動先のURLが示される
- 実際は303の用途で使われることが多く、307として再定義されたため現在は推奨されていない

303 See Other

- リクエストされたリソースを別のURIで取得できることを示す
- Locationヘッダに移動先のURLが示される
- ブラウザのフォームからPOSTで処理を行ったレスポンスとして、結果画面にリダイレクトするときに使う

307 Temporary Redirect

- 一時的リダイレクト
- 302の規格外な使用法が横行したため再定義したもの
- Locationヘッダに移動先のURLが示される
- ユーザーエージェントは使用するHTTPメソッドを変更してはならない

308 Permanent Redirect

- 恒久的リダイレクト
- 301の規格外な使用法が横行したため再定義したもの
- Locationヘッダに移動先のURLが示される
- ユーザーエージェントは使用するHTTPメソッドを変更してはならない



400番台

クライアントエラー

400 Bad Request

- リクエストが不正
- 定義されていないメソッドを使ったり、パラメータに間違いがあるなど、クライアントのリクエストがおかしい場合
- ほかに適切なクライアントエラーを示すステータスコードがない場合にも用いる

401 Unauthorized

- 認証エラー
- ログインが必要なページでIDやパスワードを間違えた場合
- リクエストに必要なAuthorizationヘッダを含まない場合
- トークンが失効、破損しているなど不正な場合

403 Forbidden

- 認可エラー
- 許可されていないなどの理由でクライアントのアクセス権限がなく、サーバからのレスポンスが拒否された
- 特定のIPアドレスのみからアクセスできる場合などに用いる
- 401とは異なり、クライアントの識別はされている

404 Not Found

- リクエストされたリソースが存在しない
- Webで頻繁に見られる有名なエラーステータスコードのひとつ
- そもそもURI自体が存在しないのか、取得対象のリソースが存在しなかったのかなど、詳細情報を示す必要がある
- 許可されていないクライアントからリソースの存在を隠すため、403の代わりに404を返すこともある

405 Method Not Allowed

- エンドポイントは存在するが許可されていないHTTPメソッド
- GETでアクセス可能な検索APIをPOSTでしようとした場合
- APIがリソースのDELETEを禁止している場合

408 Request Timeout

- 指定時間内にリクエストが完了しなかった
- Chrome、Firefoxなど、閲覧を高速化するためのHTTP事前接続機能を使用するブラウザでよく使用される
- 回線の接続速度が低下している場合などに発生

409 Conflict

- リソースの競合が発生した場合
- すでに存在するメールアドレスや同一IDのユーザー登録を行おうとした時
- 空ではないディレクトリを削除しようとしたたり、リソースの名前をすでに他で使われているものに変更しようとした時など

410 Gone

- リクエストされたコンテンツがサーバから永久に削除され、転送先アドレスがない場合
- 404と違い「かつて存在していたが今はもう存在していない」ことを表す
- 期間限定のプロモーションサイトなどで利用することを意図している
- ユーザー情報を扱うAPIで410を返す仕様は、個人情報保護の観点から問題視される可能性もある（削除したという情報を保持している≠完全に削除されていない）

413 Payload Too Large

- リクエストヘッダ、ボディがサーバで定めている上限を超えていることを示す
- 許容サイズ以上のファイルアップロードが行われた時など
- サーバはコネクションを閉じるか、Retry-After ヘッダを返す

414 URI Too Long

- クライアントがリクエストしたURIが、サーバで扱える長さを超えている
- クエリパラメータに長過ぎるデータが指定された時など

415 Unsupported Media Type

- リクエストされたデータのメディア形式（Content-Type）にサーバーが対応しておらず、サーバがリクエストを拒否した
- JSONリクエストしか受け取れないAPIにXMLが送られた、サポートする画像形式以外の画像をアップロードしようとした時など

429 Too Many Requests

- アクセス回数が許容範囲の限界を超えた場合に返す
- 2012年にRFC6585で定義された新しいステータスコード
- 一定の時間内にレートリミットを超える大量のリクエストを送信した（例：1分間に60回などのAPIリクエスト制限）



500番台
サーバエラー

500 Internal Server Error

- サーバ側に何らかの異常が発生し正常なレスポンスが返せない
- 「何らかの異常が発生しました」 的なエラーメッセージが返ることが多く、クライアント側では解決不能
- ほかに適切なエラーコードがない場合にも用いる
- サーバーのエラーログを見れば原因がわかるかも

502 Bad Gateway

- ゲートウェイまたはプロキシに問題が発生している
- ゲートウェイとして動作するサーバが無効なレスポンスを受け取った

503 Service Unavailable

- サーバがリクエストを処理する準備ができていない状態
- 一時的なアクセス集中やメンテナンスによりサーバがダウンしている
- メンテナンスの場合は、Retry-Afterヘッダに再開予定時期（およそ何秒後か）を含めることもできる

504 Gateway Timeout

- ゲートウェイとして動作するサーバが指定時間内にレスポンスを得られなかった
- サイト移行などによる一時的なDNS異常の場合も

505 HTTP Version Not Supported

- リクエストしたHTTPプロトコルのバージョンにサーバが対応していない



まとめ

👉 ステータスコードを意識した設計を行うことで
サーバとクライアント間の開発がスムーズに進む

😅 不適切なステータスコードを割り当ててしまうと
クライアント側が混乱し、システム全体の挙動に支障をきたすことも...

😊 ステータスコードを理解することで
問題の切り分けや判断がしやすくなる

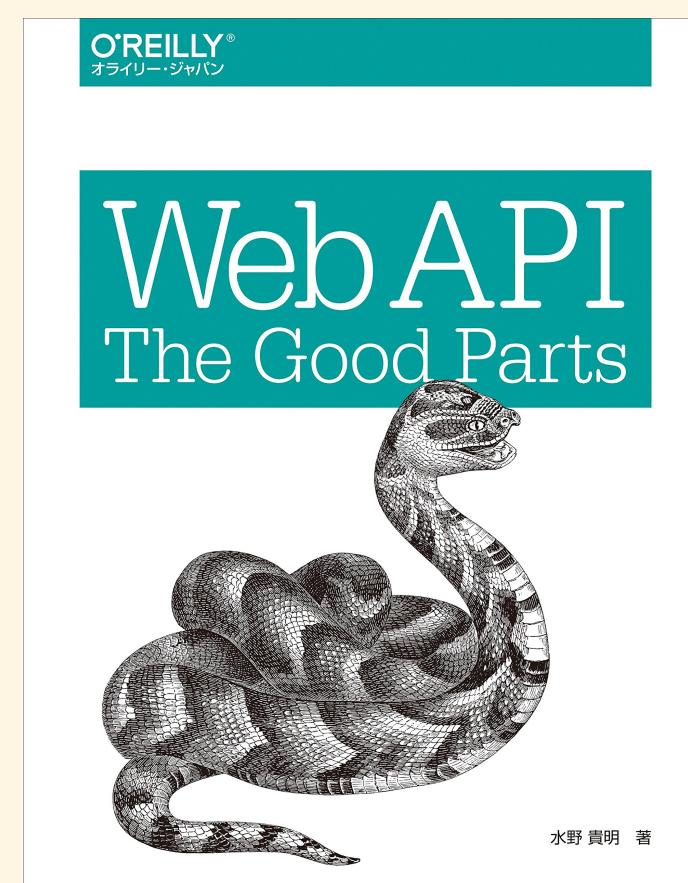
参考書籍



Webを支える技術 — HTTP、URI、HTML、そしてREST

山本陽平 著 (WEB+DB PRESS plus)

2010年4月 発行



Web API: The Good Parts

水野 貴明 著 (オライリージャパン)

2014年11月 発行

参考URL

- HTTP レスポンスステータスコード - HTTP | MDN
<https://developer.mozilla.org/ja/docs/Web/HTTP/Status>
- IETF Documents
<https://tools.ietf.org/html/>
- HTTPステータスコード - Wikipedia
<https://ja.wikipedia.org/wiki/HTTPステータスコード>



HTTP ステータスコード

完全に理解した？