システム開発 ストレートフラッシュ 共有ノート(API編)

バックエンド開発に必要な知識や情報を共有するファイルです。

目次

APIとは

APIという言葉は、Application Programming Interfaceの略称です。

ソフトウェアやWEBサービスの間をつなぐインターフェースという意味合いがあります、

例えば以下のような場面はまさにAPIの役目です。

今日の大阪の天気をアプリケーションに表示したい！

天気予報API

音声をリアルタイムで文字列に起こしたい！

Speech-to-Text-API

APIを利用することで難しい処理を自分で書かずともアプリケーションに取り入れることができます。

開発でよくある場面だと、フロントがDBを操作したい時にバックエンド側でDBを操作できるAPIを定義することで

フロントは間接的にそして簡単にDBを操作することができます。

APIの方式とRESTAPI

APIの方式

APIにも様々な方式があります。 現在よく使われている方式は以下の3つです。

RESTAPI

gRPC

GraphQL

今回の開発ではその中でも最も使われているREST APIという方式を使っていきます。

REST APIとは

RESTと呼ばれる以下の設計原則に従って実装されたAPIをREST APIといいます。

REST APIの設計原則

- アドレス可能性(Addressability) : 提供する情報がURIを通して表現できること。全ての情報はURIで表現される一意なアドレスを持っていること。
- 統一インターフェース(Uniform Interface) : 情報の操作(取得、作成、更新、削除)は全てHTTPリクエストメソッド(GET、POST、PUT、DELETE)を利用すること。
- ステートレス性(Stateless) : HTTPをベースにしたステートレスなクライアント/サーバプロトコルであること。セッション等の状態管理はせず、やり取りされる情報はそれ自体で完結して解釈できること。
- 接続性(Connectability) : 情報の内部に、別の情報や(その情報の別の)状態へのリンクを含めることができること。

※分かりにくいですが、基本的に上２つ(アドレス可能性,統一インターフェース)の考えさえ分かっていれば大丈夫です。

REST APIの例

エンドポイント(URL)

```
xxxxxx.com/users
```

上記の場合、URLで表現されるusersという一意なリソースに対して

どのようなHTTPリクエストメソッドでアクセスするかで処理を決定します。

GETでアクセス : ユーザーの参照

POSTでアクセス : ユーザーの登録

PUTでアクセス : ユーザーの更新

DELETEでアクセス : ユーザーの削除

REST APIを支える技術

HTTPリクエストメソッド

クライアントからサーバーに対しての命令の種類だと思ってもらえればいいです。

種類はいっぱいありますが、REST APIを実装するにあたって必要なメソッドは以下の4つです。

GET(参照)

POST(登録)

PUT(更新)

DELETE(削除)

HTTPリクエストメソッドの一覧

HTTPステータスコード

REST APIではAPIの処理結果に対して適切なステータスコードを返却することで、クライアントにAPIでの処理の結果を伝えます。

200 : 正常終了

400番台 : クライアントエラー

500番台 : サーバーエラー

400 401 404当たりはエラーハンドリングでよく使うと思います！

HTTPステータスコードの一覧

JSON

アプリケーション間でデータをやり取りする時に使用するデータフォーマットの事です。

PHPでいう連想配列と同様に、KEYとVALUEでデータを管理します。

ネストすることもでき、様々な表現ができます。

以下の例では likeというKEYに対して2個の要素を持つ配列をVALUEとして返しています。

```
{
    user_id:1,
    name:"yoshii",
    class:"IE4A",
    like:[
        "music",
        "game"
    ]

}
```