WebAPIの仕組みや利点

J23072　蝦名武尊

# はじめに

このドキュメントの目的はWebAPIの仕組みや利点を整理し、通信の流れをまとめることである。今回はPythonを使用するため、FastAPIのフレームワークを利用する。

# WebAPI

## 2.1 WebAPIとは

　WebAPIとはネットワークを経由してデータをやり取りする方法の一つである。例えば、手持ちのスマホを利用して目的地までの最短経路の検索を行うとする。この時、スマホはネットワーク越しのサーバに最短経路の情報を請求し、サーバから請求されたv最短経路の情報が返ってくる。この情報を使用しスマホは画面上に最短経路を表示する。この場合、情報を返したサーバをWebAPIサーバといい、スマホからの情報請求をリクエスト、その返答をレスポンスという。またリクエストを送信した存在をクライアントという。これにより、アプリケーション同士が効率的に通信し、情報を取得・処理できる。

## 2.2 メリット

　WebAPIのメリットは大きく分けて２つである。

1. インターフェースにのみ依存する

　クライアントとデータベース間にWebAPIを設置することでデータベースの種類やカラム名の変更に伴うクライアントの変更点をなくし、WebAPIの変更のみに抑えることができる。

1. セキュリティ対策

　クライアントからあるデータを取得するリクエストが送信されても、クライアントはアクセス権を有しているのか判別してからレスポンスを返すロジックをWebAPIに組み込むことができる。

## 2.3 通信の流れ

### 2.3.1リクエストの送信

　クライアントがWebAPIサーバにデータを要求する。リクエストはHTTPメソッドとともに送信される。メソッドの種類はデータを取得する”Get”、送信する”POST”、作成・更新する”PUT”、削除する”DELETE”などである。

2.3.2 サーバでの処理

　サーバでは受け取ったリクエストを解析し適切な処理を実行する。リクエストの種類に応じてデータベースサーバへアクセスを行ったり、セキュリティ確保のための認証チェック、ビジネスロジックの実行を行う。

### 2.3.3 レスポンスの返答

　リクエストに応じた処理を行った後、実行結果をレスポンスとしてクライアントに送信する。レスポンスはJSONやXML形式で返却される。その後、レスポンスデータを受け取ったクライアント側はデータを処理し、UIに表示したり次の処理に活用する。

以下は通信の流れを図に表したものである。

[クライアント] → (リクエスト) → [サーバー] → (処理) → [データベース]

↑ ↓

(レスポンス) (データ取得/更新)

# 終わりに

WebAPIを使用することでデータベースの変更による影響を最小限に抑えることができたり、アクセス権の確認の組み込みによりウイルスや不正アクセスを防ぐことができセキュリティの向上にも役立てることができる。また、アプリケーション同士での効率的なデータ通信を可能にする。

# 参考文献

・Pythonプログラミング VTuber サプー.” **【Fast API 入門】PythonでWeb APIを作ってみよう！簡単にAPIが作れるフレームワークの紹介 〜初心者向け〜.**”Youtube. 2024/01/06, <https://youtu.be/kZHdC-_yPgI?si=i5uMB6AJxsFszr_u>