# Web アプリ開発の基礎：バックエンド完全ガイド

開発学習ノート

2026 年 2 月 7 日

## 目次

## はじめに

本ドキュメントは、Python と FastAPI を用いたバックエンド（サーバーサイド）開発の仕組みと、各ファイルの役割を整理したものである。「厨房」の役割を果たすサーバーが、どのように動いているかを解説する。

## 第 1 部：バックエンド開発の流れ

### Step 1: 環境の隔離（Virtual Environment）

Python 開発では、プロジェクトごとに「専用の道具箱」を作るのが鉄則である。

**venv（ブイ・エンブ）**

「Virtual Environment（仮想環境）」の略。パソコン全体の設定を汚さずに、このアプリ専用の Python 環境を作る機能。

- コマンド: `python -m venv venv`
- 有効化: `.\venv\Scripts\Activate` （これを行うと行頭に（venv）が付く）

### Step 2: サーバー構築（FastAPI）

**FastAPI（ファスト・エーピーアイ）**

Python で Web API（アプリの窓口）を作るための最新フレームワーク。高速で、コードが書きやすく、自動でドキュメントを作ってくれる優れもの。

**Uvicorn（ユービコーン）**

書いた Python コードを、実際に Web サーバーとして動かすための実行ソフト。FastAPI とセットで使う。

### Step 3: 通信の許可（CORS）

**★ CORS（コアーズ：Cross-Origin Resource Sharing）**

Web ブラウザのセキュリティ機能。「違う場所（オリジン）からのアクセス」をブロックする仕組み。

**今回の状況:**

- フロントエンド: `localhost:5173` （React）
- バックエンド: `localhost:8000` （Python）

住所（ポート番号）が違うため、何もしないと通信がブロックされる。バックエンド側で「5173 からのアクセスは OK だよ」と許可証（Allow Origins）を発行する必要がある。

# 第 2 部：ファイル構成図解（Backend Map）

study-api フォルダ内の構成。

## 1. 実行ファイル

- **main.py**
  **【役割：厨房の司令塔】**
  バックエンドの全てのロジックが書かれたファイル。
  - **CORS 設定**: React からの接続を許可する設定。
  - **BaseModel (Pydantic)**: 「どんなデータ（型）を受け取るか」を定義した伝票。今回は「学年」「教科」「勉強時間」などを定義した。
  - **@app.post("/diagnose")**: 実際の注文受付窓口。データを受け取り、処理（今回はモックの返信）をして返す場所。

## 2. 環境・システムファイル

- **venv/ （フォルダ）**
  **【役割：専用の道具箱】**
  このフォルダの中に、インストールしたライブラリ（FastAPI など）の実体が入っている。
  - **注意**: このフォルダは Git などにアップロードしてはいけない（サイズが大きいため）。

# 第 3 部：フロントエンドとの連携の仕組み

## 1. リクエスト（注文）

React（フロントエンド）の fetch 関数が、Python サーバーの /diagnose という URL にデータを投げる。

```
// フロントエンド側のコードイメージ
fetch('http://127.0.0.1:8000/diagnose', {
    method: 'POST',
    body: JSON.stringify(formData) // 入力データをにして送る JSON
})
```

## 2. レスポンス（料理の提供）

Python（バックエンド）がデータを受け取り、辞書型（dict）で返事を返す。FastAPI が自動的にこれを JSON（Web の標準データ形式）に変換して React に届ける。

```
# バックエンド側のコードイメージ
return {
    "summary": "からのアドバイス AI...",
    "books": [...]
}
```

## 開発中の注意点

**【注意】 2 つのターミナルを維持する**

アプリを動かすためには、以下の 2 つが同時に動いている必要がある。

1. **React (Wait Staff)**: `npm run dev`
2. **FastAPI (Chef)**: `uvicorn main:app -reload`

片方でも止まっていると、画面が表示されなかったり、診断ボタンを押してもエラーになったりする。