

MVC構造

MVCは、アプリケーション、ソフトウェア、ウェブサイトを構築するための構造です。

この略語は次の意味を持ちます：

- モデル (Model)
- ビュー (View)
- コントローラー (Controller)

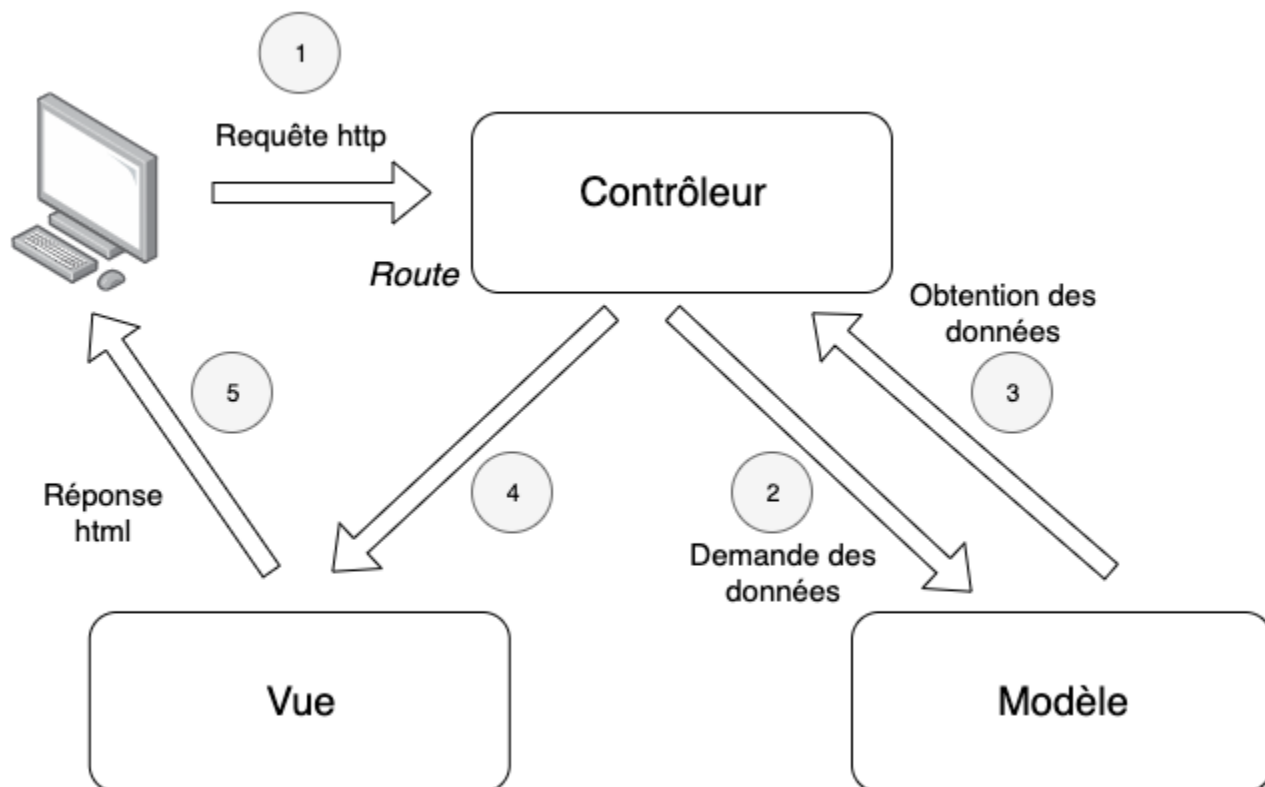
コードを3つの明確な部分に分けることで、コードの整理と管理をしやすくします。

これまでは、データベースのクエリ、アルゴリズム、HTMLがすべて同じファイルにあり、何かを変更したいときは、何十行ものコードから目的の部分を探さなければなりません。

MVCの基本的な考え方は次のとおりです：

- **コントローラー (Controllers) :**
ページごとの処理ロジック（アルゴリズム）をテーマ別にまとめたファイル。
- **モデル (Models) :**
データベースへのクエリをテーマ別にまとめたファイル。
- **ビュー (Views) :**
ページの表示部分だけを記述するファイル。

MVCでは、すべてのユーザーのリクエストはコントローラーに送られます。コントローラーは必要に応じてモデルとやり取りし、最終的にビューをユーザーに返します。



ここでは、CRUD講座で作成したブログサイトをMVC構造で再現します。

1. ルーター

ルーターを使わずにMVC構造を作成することも可能です。

しかしその場合、ページの数だけコントローラーファイルを作成する必要があり、または多くの情報をGETパラメータで渡さなければなりません。

ここでは、ページをテーマごとに整理するために、ルーターを改良してMVCを管理できるようにします。

手順：

1. `.htaccess` – ルーターの講座で使用したものと同じ。
2. `routes.php` – コントローラーと関数を追加。
3. `index.php` – 関数の呼び出し。

2. 最初のモデル

`model`フォルダーを作成し、その中に`userModel.php`というファイルを作ります。

このファイルの中で、PDOを使ってデータベース接続ファイルをインクルードします。

前述の通り、モデルはデータベースへのクエリをまとめるためのものです。

ファイル名 (`userModel.php`) が示すように、このモデルはユーザーに関するクエリを含みます。

3. 最初のビュー

次に、`view`フォルダーを作成し、その中に`user`というサブフォルダーを作ります。

この中に以下の4つのファイルを作成します：

- `list.php`
- `inscription.php`
- `delete.php`
- `update.php`

これらのファイルは、ページの表示部分のみ（HTMLとPHPの`echo`）を含みます。

4. 最初のコントローラー

`controller`フォルダーを作成し、その中に`userController.php`というファイルを作ります。

このファイルの中で、ユーザーに関連する4つのページを処理する4つの関数を作成します。

これらの関数は、モデルとのやり取り、ロジック処理、ビューの読み込みを担当します。

MongoDBバージョンのモデル

これまでに、MySQLを使ってMVC構造の完全なCRUDを作成しましたが、

今回は代わりにMongoDBを使って同じことを行います。