

## MVC とは

MVC は「Model(モデル)」「View(ビュー)」「Controller(コントローラー)」の頭文字をとった略称で、システム開発のアーキテクチャ(設計)の一つです。また、VM(ビューモデル)を追加した MVVM といったアーキテクチャもあります。

### Model

システム開発におけるビジネスロジックをさします。ビジネスロジックは「3階層システム」である「プレゼンテーション層」「ビジネスロジック層」「データアクセス層」の一つですが、ここでは説明を割愛します。

モデルは、この世に存在する実態をデータや振る舞いなどをプログラムとして設計する役割を担います。

例えば商品モデルにおいて商品名や価格などの性質や、購入処理や方法のような抽象的な振る舞いをモデリングします。

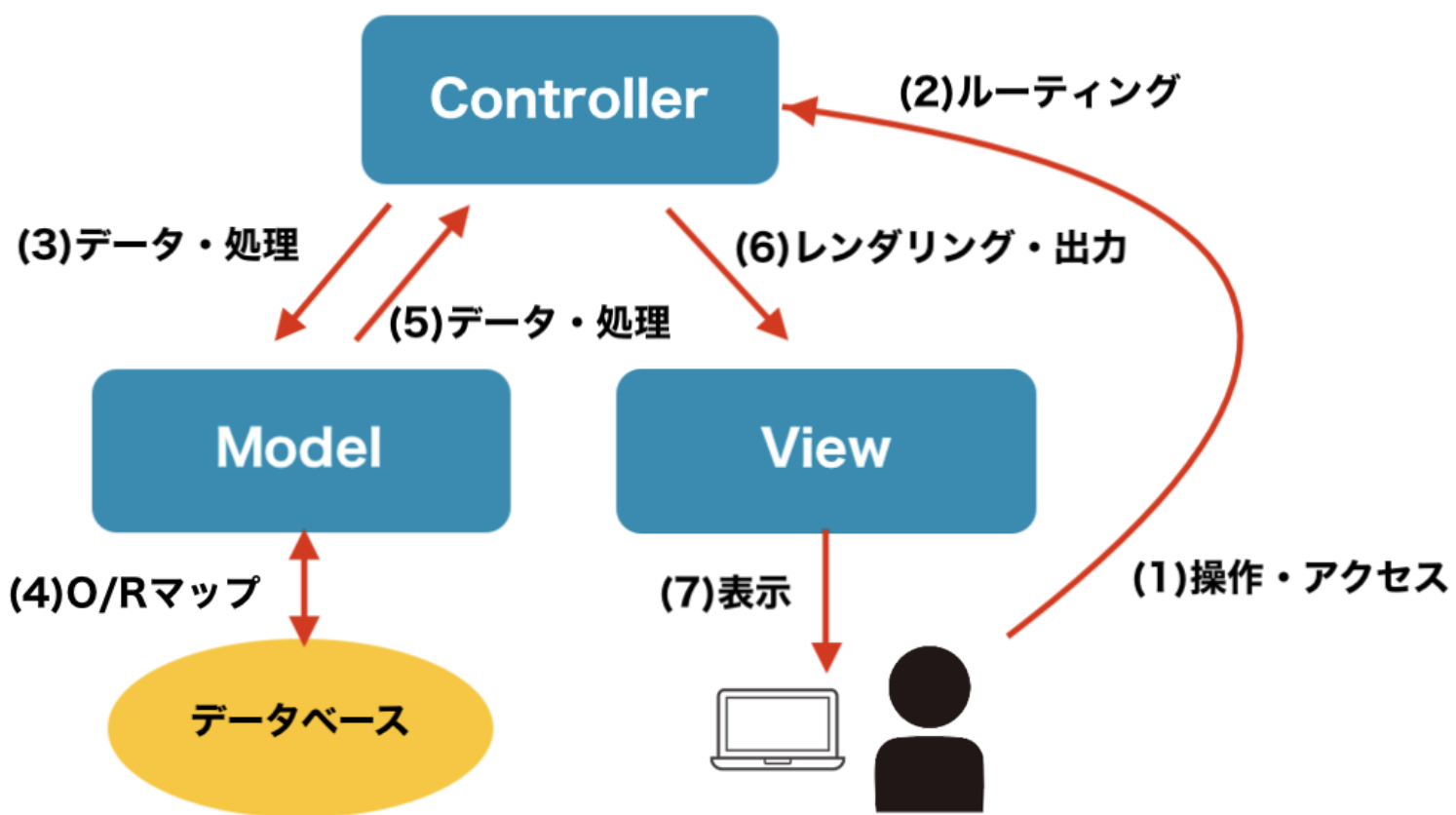
### View

View はユーザが閲覧したり操作するための画面表示・入出力の役割を担います。

### Controller

Contrller は、Model と View の中間的に位置にあり、制御するための役割を担います。

実際のシステム開発では複雑な処理は Model やその他 Serviceファイルに任せて、シンプルな操作だけを処理させることが理想となります。



## ルーティング と命名規則

Web開発における Controller は、URL を元にしたルーティングを用いるのが一般的で、Controller(コントローラー)とAction(アクション)を利用します。

例えば、以下の URL があったとします。

```
http://example.com/item/edit/1
```

example.com 以降が Webシステム開発において意味を持たせる必要があり、URL を分解すると URLは「item」「edit」「1」と分かります。これを一般化すると以下ようになります。

```
http://example.com/コントローラ名/アクション名/ID
```

「コントローラ名」「アクション名」「ID」という設計になっていますが、設計によっては、管理者の admin やユーザの user をつけることも可能です。

```
http://example.com/admin/item/edit/1
```

これは「管理者ユーザで商品ID = 1 の編集」と解釈できます。

### Controller & Action & id

http://example.com/item/edit/1 を例にすると以下のようにまとめることができます。

URL	種別	プログラム上	役割
item	コントローラー	ItemController クラス	商品の操作
edit	アクション	edit() メソッド	編集
1	id	id = 1	データベースの商品ID が 1

### プログラムとの関係

上記の URL の例では、ItemController クラスの edit() メソッドを実行して View を表示します。

```
// (1) ItemController クラスを特定
class ItemController {
    //(2) edit() メソッドを実行
    public function edit(Request $request) {
        //(3) id を取得(id = 1)
        $id = $request->id;

        //(4) テンプレートエンジンで View のレンダリング処理(HTML表示)
    }
}
```

詳しいプログラムの内容は初心者には難しいかもしれませんが、URL に意味やルールをつけることで、プログラムが自動化され作業効率があがると同時に、システム設計もわかりやすくなります。

## View ファイル

View は Webシステムの場合は最終的に HTML を出力してブラウザで表示します。

```
class ItemController {
    public function edit() {
        // 処理
        // View ファイルにデータを渡して HTMLを出力
    }
}
```

## テンプレートファイル

プログラム処理の最後に HTML を出力してブラウザで表示します。その際にテンプレートファイルという HTML と データを同時に記述するファイルを用意します。

## テンプレートファイルの命名の重要性

テンプレートファイル名は、なるべくアクション名にあわせた方がわかりやすいでしょう。HTML のプロトタイプを作成する場合も、View ファイルを作成することになるので、ファイルの命名はとても重要な作業といえます。

今回の例では、edit アクションを検知すると ItemController の edit() メソッドを実行しますが、テンプレートファイル名もあわせて edit という名称をつけた方がわかりやすいでしょう。

### Laravel の例

例えば、フレームワークの Laravel では、ItemController クラスの edit() メソッドで処理し、テンプレートエンジンを利用して HTML 出力します。

```
class ItemController extends Controller
{
    public function edit(Request $request)
    {
        $item = Item::find($request->id);
        $data = compact('item');
        return view('item.edit', $data);
    }
}
```

## Laravel でのテンプレートファイル

Laravel では resources/views/ にテンプレートファイル blade.php を配置します。よって、ItemController の edit ファイルは以下のパスになります。

```
resources/views/item/edit.blade.php
```

## 静的 HTML で表現

プログラムが難しい場合は、まず HTML でプロトタイプを作成してみるのもよいでしょう。例えば、ItemController の edit アクションを意識する以下のファイルパスで表現できます。

### ItemController->edit()

```
./item/edit.html
```

### ItemController->create()

商品の新規追加であれば、アクション名を create とした場合、以下のファイルパスになります。

```
./item/create.html
```

### Admin/ItemController->index()

管理者画面の商品一覧表示であれば、管理者ディレクトリ admin に item/index.html を作成します。

```
./admin/item/index.html
```

このように、URL や ディレクトリ、ファイル名を意識することで わかりやすく、管理しやすい開発ができます。