

# 私たちの最初のコントローラー

---

Symfony は MVC 構造を使っているので、コントローラーが必要になります。

## コントローラーの作成

Symfony では、コントローラーファイルを手動で作成することはほとんどありません。通常は CLI を使います：

```
symfony console make:controller
# またはコントローラー名を直接指定：
symfony console make:controller ControllerName
```

このコマンドはとてもわかりやすいです：Symfony にコンソールを使ってコントローラーを生成するように頼んでいます。

Symfony は次に名前を聞いてきます — ここでは **HomeController** を選びます。

すべてのコントローラー名には必ず「Controller」という単語を含める必要があります。もし忘れたりタイプミスをすると、Symfony が自動的に「Controller」を名前に追加します。

次に PHPUnit テストを生成するか聞かれます（「実験的」と表示されます）。今は「いいえ」と答えましょう。

Symfony は以下の2行を表示します：

```
created: src/Controller/HomeController.php
created: templates/home/index.html.twig
```

これはコントローラーとテンプレートの両方が作成されたことを意味します。今はテンプレートは無視して、コントローラーに集中しましょう。

## コントローラーの構造

コントローラーファイルを行ごとに分解してみましょう：

- まず名前空間（namespace）。
- 次にクラスをインポートするための `\use` 文（`require` は不要；Symfony は Composer のオートローダーを使います）。
- 次に、ファイル名と同じ名前のクラスがあり、コントローラーのヘルパーメソッドを提供する抽象クラスを継承しています。
  - このクラスの中に `index` メソッドがあります。名前はあまり重要ではありません。
  - メソッドの上には `Route` 属性があります（`#[...]` は PHP 8+ の属性で、コメントではありません）。
    - この属性がついたメソッドは、それぞれ独自のページ（ルート）になります。

- ルート名は変更できますが、一度使い始めたら後で変えないようにしてください。
- パス文字列（例：`/blog`）はいつでも変更可能で、Symfony が全て自動で更新します。
- プロジェクト内のすべての内部リンク生成にルート名を使います。
- メソッドは現在ひとつのことをしています：`AbstractController` から継承した `render()` メソッドを呼んでいます。
  - 最初のパラメーターは表示するビュー（自動的に `templates/` フォルダーを探します）。
  - 二番目のパラメーターはビューに渡す連想配列の変数（その他の PHP 変数はビュー内でアクセスできません）。

コントローラーの他の機能は後で詳しく見ていきます。今はビューに集中しましょう。

補足：ルートはメソッド上に書く代わりに、`config/routes.yaml` で定義することもできます。

最後に、いくつかの変数（配列やランダムな数字を含む）をビューに渡してみましょう：

```
[
    'controller_name' => 'HomeController',
    'fruits' => ["banana", "tomato", "cherry"],
    'pays' => ["france" => "Bonjour le monde!", "england" => "Hello World!"],
    'chiffre' => rand(0, 10),
    'vide' => [],
    'xss' => "<script>alert('Hi');</script>"
]
```

そしてルートのパスを `/` に、名前を `app_home` に変更しましょう。