Introduction au Modèle MVC en PHP

Une architecture pour des applications web efficaces

Christophe BLANC

IUT Clermont-Auvergne

February 11, 2024

Pourquoi les Patrons de Conception ?

- Simplification du code et réduction de la redondance.
- Facilitation de la maintenance et de l'évolution du logiciel.
- Encouragement des bonnes pratiques de programmation.
- Fourniture d'une terminologie commune pour les développeurs.

Catégories de Patrons de Conception

Les patrons de conception se divisent en trois catégories principales :

1. **Créationnels** : concernent la création d'objets. Exemples : Singleton, Factory, Builder.
2. **Structuraux** : se rapportent à la composition des classes ou des objets. Exemples : Adapter, Decorator, Proxy.
3. **Comportementaux** : se concentrent sur la communication entre les objets. Exemples : Observer, Strategy, Command.
4. **Architecturaux** : se concentrent sur des stratégies pour l'organisation globale, la structure et la communication au sein des systèmes logiciels, en facilitant la définition de l'architecture globale des applications.. Exemples : MVC

Avantages des Patrons de Conception

- **Réutilisation du code** : promouvoir la réutilisation des solutions éprouvées.
- **Clarté architecturale** : clarifier la structure et l'intention du code.
- **Efficacité du développement** : accélérer le développement grâce à des approches standardisées.

Inconvénients Potentiels

- **Complexité supplémentaire** : risque d'ajouter une complexité inutile si mal utilisés.
- **Sur-ingénierie** : tentation de sur-utiliser les patrons dans des situations simples.
- **Apprentissage** : courbe d'apprentissage pour comprendre et appliquer correctement.

Comment choisir le bon Patron ?

- Identifier clairement le problème à résoudre.
- Considérer la flexibilité et l'évolutivité requises du système.
- Évaluer la complexité ajoutée par le patron.
- S'appuyer sur l'expérience et les retours de la communauté.

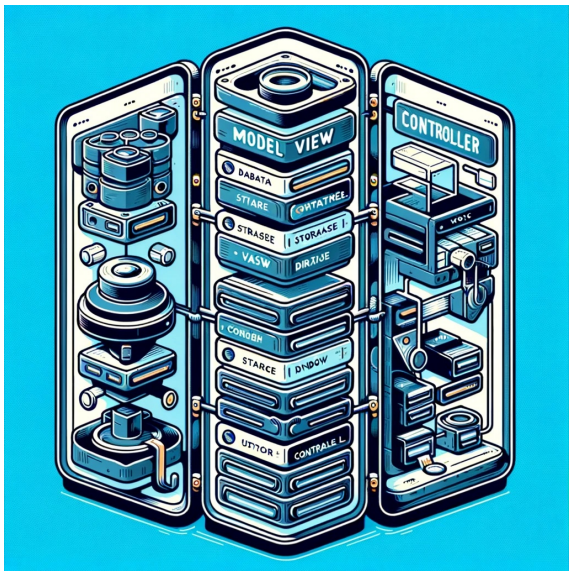
Conclusion

Les patrons de conception sont des outils puissants pour les développeurs, offrant des solutions structurées à des problèmes courants en développement logiciel. Bien choisis et correctement appliqués, ils peuvent grandement améliorer la qualité et la maintenabilité des projets.

Qu'est-ce que le MVC ?

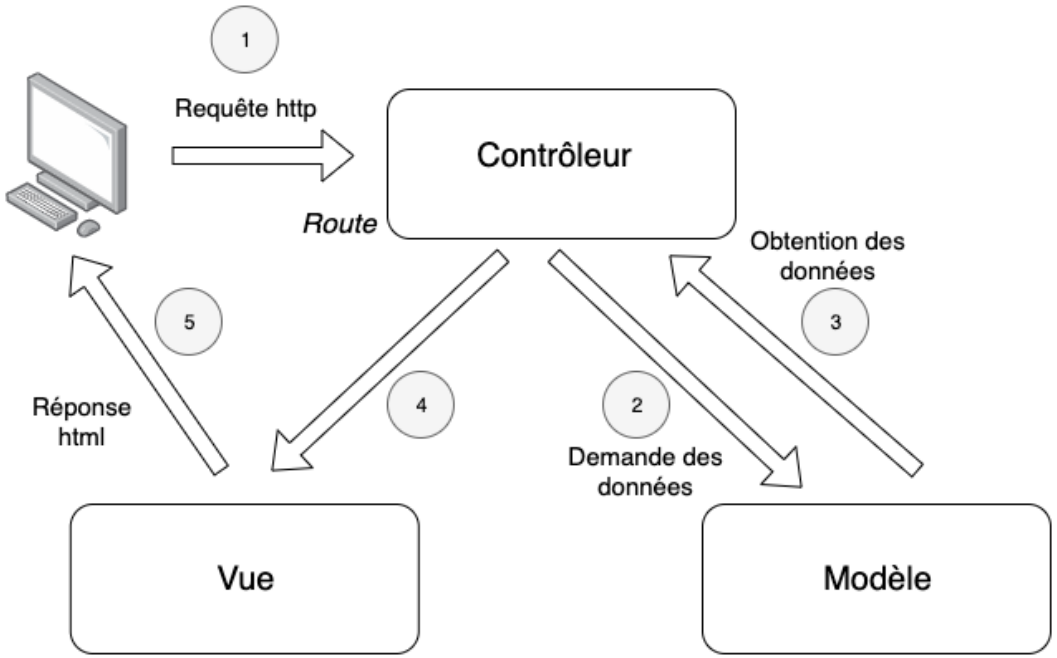
Le MVC est un patron de conception qui sépare une application en trois composants principaux: le modèle, la vue, et le contrôleur.

Objectif: faciliter l'organisation du code et améliorer la maintenabilité.



Qu'est-ce que le MVC ?

Le patron de conception MVC a été créé dans le but de mettre en œuvre des interfaces utilisateur.
Le cycle action→mise à jour→affichage induit par ce patron est bien adapté aux applications web.
La vue attend des modifications du modèle, puis modifie la présentation des éléments visuels correspondants.



Composant Modèle

Définition: cette partie gère ce qu'on appelle la logique métier de votre site. Elle comprend notamment la gestion des données qui sont stockées, mais aussi tout le code qui prend des décisions autour de ces données. Son objectif est de fournir une interface d'action la plus simple possible au contrôleur. On y trouve donc entre autres des algorithmes complexes, des appels aux APIs, des requêtes SQL.

Exemple: Opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) sur une base de données.

Composant Vue

Définition: cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.

Exemple: HTML, CSS, et JavaScript générant l'interface utilisateur.

Composant Contrôleur

Définition: cette partie gère les échanges avec l'utilisateur. C'est en quelque sorte l'intermédiaire entre l'utilisateur, le modèle et la vue. Le contrôleur va recevoir des requêtes de l'utilisateur. Pour chacune, il va demander au modèle d'effectuer certaines actions (lire des articles de blog depuis une base de données, supprimer un commentaire) et de lui renvoyer les résultats (la liste des articles, si la suppression est réussie). Puis il va adapter ce résultat et le donner à la vue. Enfin, il va renvoyer la nouvelle page HTML, générée par la vue, à l'utilisateur.

Exemple: Traitement des requêtes de l'utilisateur, appel du modèle approprié, et sélection de la vue à afficher.

Avantages du MVC

- Développement parallèle: les développeurs peuvent travailler sur le modèle, la vue, et le contrôleur en parallèle.
- Facilité de maintenance: séparation claire des préoccupations.
- Réutilisabilité du code: le modèle peut être réutilisé sans modification par différentes vues.

Implémentation en PHP

Exemple simple: Création d'un contrôleur, d'un modèle, et d'une vue en PHP.

Structure des Dossiers MVC

```
/mvc_example
    /controller
        MainController.php
    /model
        UserModel.php
    /view
        home.php
    index.php
    /css
        style.css
    /js
        script.js
```

Note : Cette structure est un exemple basique d'une application MVC, où chaque composant du modèle MVC est organisé dans son propre dossier.

Composant Modèle - UserModel.php

```php
<?php
// Fichier : /model/UserModel.php

class UserModel {
    private $name = 'Utilisateur Exemple';

    public function getName() {
        return $this->name;
    }
}
?>
```

Composant Vue - home.php

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemple MVC en PHP</title>
6  </head>
7  <body>
8      <h1>Bienvenue, <?php echo $name; ?>!</h1>
9  </body>
10 </html>
```

Composant Contrôleur - MainController.php

```php
<?php
// Fichier : /controller/MainController.php

require_once __DIR__ . '/../model/UserModel.php';

class MainController {
    public function home() {
        $userModel = new UserModel();
        $name = $userModel->getName();

        require __DIR__ . '/../view/home.php';
    }
}
?>
```

Routeur - index.php

```php
<?php
// Fichier : /index.php

require_once __DIR__ . '/controller/MainController.php';

$controller = new MainController();
$controller->home();
?>
```

Gestion des fichiers en php

- PHP offre plusieurs fonctions pour inclure des fichiers : require, include, require_once, et include_once.
- Choisir la bonne fonction est crucial pour la bonne gestion des dépendances et la structure du projet.
- L'utilisation de __DIR__ assure que les chemins des fichiers sont correctement résolus, améliorant ainsi la portabilité du code.

require vs include

- **require** génère une erreur fatale et arrête l'exécution du script si le fichier spécifié ne peut pas être inclus.
- **include** génère une alerte (warning) mais le script continue de s'exécuter même si le fichier ne peut pas être trouvé.

```php
require 'config.php';
include 'template.php';
```

require_once vs include_once

- **require_once** et **include_once** vérifient si le fichier a déjà été inclus et, si c'est le cas, ne l'incluent pas une deuxième fois.
- require_once est utilisé pour inclure des fichiers qui sont essentiels au script.
- include_once est préférable pour les fichiers comme les templates ou les bibliothèques facultatives.

```
1 require_once 'config.php';
2 include_once 'template.php';
```

Utilisation de __DIR__

- __DIR__ donne le chemin du dossier du fichier courant, ce qui aide à créer des chemins absolus.
- Utiliser __DIR__ avec require ou include garantit que le chemin du fichier reste correct quel que soit l'endroit d'où le script est exécuté.

```php
require __DIR__ . '/config/database.php';
include __DIR__ . '/views/header.php';
```

Conclusion gestion des fichiers

- Choisir entre require, include, require_once, et include_once dépend de la nécessité du fichier pour l'exécution du script et si le fichier doit être inclus plusieurs fois.
- L'ajout de __DIR__ améliore la fiabilité et la portabilité du code en spécifiant des chemins absolus.

Conclusion

Le modèle MVC offre une structure robuste pour le développement d'applications web en PHP. Encourager l'adoption du MVC pour des projets PHP structurés et maintenables.

Questions & Réponses

Des questions ?