

L'architecture MVC avec Symfony (framework PHP)

Objectifs :

A la fin de cette séance, vous serez capables de définir les principes de l'architecture MVC et de les mettre en place dans un projet web

Cheminement :

Phase 1 :

Introduction à l'architecture MVC

MVC (Modèle-Vue-Contrôleur) est un modèle d'architecture logicielle introduit pour la première fois dans les années 1970 dans le cadre de la conception de l'interface utilisateur des applications Smalltalk (vous trouverez ici plus d'informations sur ce langage). Il a été popularisé dans les années 1990 avec l'essor du développement de logiciels orientés objet.

L'architecture MVC divise une application en trois parties distinctes : **le modèle, la vue et le contrôleur**.



Chacune de ces parties a un rôle spécifique à jouer dans l'application :

Le modèle - représente les données de l'application, généralement stockées dans une base de données. Le modèle gère les interactions avec la base de données, les validations de données, la logique de traitement et les règles métier.

Il constitue une couche d'abstraction entre l'application et la base de données.

La vue - représente l'interface utilisateur de l'application. Elle affiche les données et permet à l'utilisateur d'interagir avec l'application. La vue est généralement un fichier HTML, XML ou un template de framework (ex. Twig) qui est rempli avec les données du modèle.

Le contrôleur - représente la couche d'interaction entre la vue et le modèle. Il reçoit les requêtes de l'utilisateur et interagit avec le modèle pour récupérer les données nécessaires. Il passe ensuite ces données à la vue pour qu'elle les affiche à l'utilisateur.

Pourquoi utiliser l'architecture MVC ?

- ⑩ **Séparation des responsabilités** : chaque composant de l'architecture MVC a une responsabilité bien définie. Le modèle est responsable des données, la vue de la présentation et le contrôleur des interactions de l'utilisateur.

En séparant la logique de présentation, les données et les interactions de l'utilisateur, l'architecture MVC rend l'application plus facile à développer et à maintenir.

⑩ **Réutilisation de code** : l'architecture MVC encourage la réutilisation de code, ce qui peut réduire le temps de développement. Par exemple, une vue peut être réutilisée pour afficher des données de différents modèles.

⑩ **Testabilité** : l'architecture MVC facilite les tests unitaires et fonctionnels de l'application car les composants sont séparés et peuvent être testés indépendamment les uns des autres.

L'architecture MVC est largement utilisée dans le développement web car elle permet de structurer les applications de manière à séparer les responsabilités et à rendre l'application plus facile à maintenir et à évoluer.

Phase 2 :

Découverte de Symfony :

Le framework open source Symfony a été créé en 2005 par l'agence web française SensioLabs, cofondée par Fabien Potencier et Grégory Pascal. Le succès de Symfony est immédiat, surtout quand Yahoo!, à l'époque, annonce qu'il va transférer un certain nombre d'outils autour de ce framework PHP. Depuis, Symfony s'est imposé parmi les frameworks PHP de référence au niveau mondial, aux côtés de Zend.

De nombreuses entreprises l'utilisent - Dailymotion, Joomla, Drupal, Magento ou encore PrestaShop ou alors le journal Le Parisien, le site francetv, le site de la BBC, Auchan.fr etc.

Symfony fournit une architecture de base pour le développement de projets web en PHP, en suivant les principes de l'architecture MVC.

Dans un projet Symfony on retrouve le Modèle (appelé Entité), les Vues (que l'on peut développer en HTML/TWIG), les Contrôleurs.

En tant que framework dit orienté objet, Symfony est composé de classes mères qui seront dérivées et étendues par **héritage** en fonction des besoins spécifiques à chaque application.

En utilisant Symfony, les développeurs peuvent se concentrer sur la logique métier de l'application plutôt que sur des détails de mise en œuvre. Ils peuvent ajouter des modules supplémentaires à mesure que le produit se développe.

Actuellement, la version LTS (Long Term Support) est la version 5.4.4, et la dernière version stable est la 6.2, mais ça change tout le temps (voir les infos sur la page [Symfony releases](#))

NOTE : Pour pouvoir utiliser Symfony sereinement, vous devez disposer des outils suivants:

une version récente de php (et les dépendances requises)

```
frc@earth:~$ php --version
PHP 7.4.3 (cli) (built: Nov 25 2021 23:16:22) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
    with Xdebug v2.9.2, Copyright (c) 2002-2020, by Derick Rethan
```

Vous devriez obtenir un message vous indiquant la version de php.

composer

Même chose pour la version de composer

```
frc@earth:~$ composer --version
Composer version 2.2.6 2022-02-04 17:00:38
```

symfony-cli

et pour symfony-cli

The message should be displayed. The file has been successfully installed. It should be possible to use the command 'symfony'.

Si l'un des composants n'est pas installé, suivez ce guide d'installation

Commandes

Création d'un projet avec composer

```
composer create-project symfony/website-skeleton monprojet1 5.4.*
```

Démarrer le serveur de développement (Ctrl C pour arrêter)

```
symfony serve
```

Créer la base de données

```
php bin/console doctrine:database:create
```

Mettre à jour le schéma de la base de données

```
php bin/console doctrine:schema:update --force
```

Assistant de génération d'entité

```
php bin/console make:entity client
```

Générer un contrôleur

```
php bin/console make:controller catalogue
```

Générer des entités à partir de la base de données (existante)

```
php bin/console doctrine:mapping:import "App%Entity" annotation --path=src/Entity
```

Générer ou régénérer les accesseurs (get/set)

```
php bin/console make:entity --regenerate --overwrite "App%Entity"
```

Générer les scripts SQL de création de la base

```
php bin/console make:migration
```

Exécuter les scripts de création de la base

```
php bin/console doctrine:migrations:migrate
```



Commandes

Création d'un projet avec composer

Démarrer le serveur de développement (Ctrl C pour arrêter)

Créer la base de données

Mettre à jour le schéma de la base de données

Assistant de génération d'entité

Générer un contrôleur

Générer des entités à partir de la base de données (existante)

Générer ou régénérer les accesseurs (get/set)

Générer les scripts SQL de création de la base

Exécuter les scripts de création de la base