

L'ARCHITECTURE MVC

Jean-Claude AZIAHA

SOMMAIRE

- 1- Analyse de l'architecture du projet "films_repository"
- 2- La problématique
- 3- Le patron de conception MVC

1- Analyse de l'architecture du projet "films_repository"

Notre projet "films_repository" fonctionne très bien en local et en production.

La phase de codage a été simple, rapide et compréhensible.

Au final, chaque page réagit différemment en fonction des paramètres qu'elle reçoit.

1- Analyse de l'architecture du projet "repertoire_films"

Selon les cas, elle fera en totalité ou en partie :

- Récupérer les contenus de \$_GET, \$_POST, \$_SESSION
- Mettre à jour \$_SESSION
- Inclure des fichiers: head, nav, footer, connexion à la BD, etc.
- Effectuer des requêtes : INSERT, UPDATE, DELETE et SELECT
- La construction de la page HTML à afficher en prenant en compte les données à charger dynamiquement.

2- La problématique

Même si tout fonctionne bien, pouvez-vous croire que cette application présente de nombreux problèmes?

En effet, plus cette application grossit, plus elle deviendra difficile à maintenir.

Les raisons sont les suivantes :

2- La problématique

* **FAILLE DE SÉCURITÉ**

Etant donné que le point d'entrée de l'application est à la même racine que tous les autres fichiers, un utilisateur peut passer par la barre d'url et accéder aux fichiers racines auxquels il n'est pas censé accéder.

2- La problématique

*** UNE ORGANISATION LIMITÉE**

- Tout est mélangé dans le même fichier : le SQL, le PHP et le HTML
- Les pages devenant grosses, leur contenu sera difficile à comprendre
- La maintenance n'est pas facile
- Le travail à plusieurs est rendu difficile

2- La problématique

On comprend donc qu'au-delà d'avoir une application qui fonctionne, d'autres facteurs sont également à prendre en compte comme :

- L'architecture des dossiers et fichiers
- La sécurité d'accès aux dossiers et fichiers
- La maintenabilité
- La capacité à travailler à plusieurs sur un même projet

C'est pour ça que nous allons apprendre à utiliser le patron de conception **MVC**.

3- L'architecture MVC

L'architecture MVC est un patron de conception (design pattern).

Un patron de conception est un ensemble de bonnes pratiques en réponse à un problème de conception d'un logiciel ou d'une application.

Il décrit donc une **solution standard**, utilisable dans la conception de différents logiciels et applications.

3- L'architecture MVC

La **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques, très populaire pour les applications web.

Le motif est composé de trois types de modules ayant trois responsabilités différentes : les **modèles**, les **vues** et les **contrôleurs**.

Il a été théorisé pour la première fois en **1978** par l'informaticien norvégien **Trygve Reenskaug**, puis réellement introduit en programmation à partir de la fin des années 80.

3- L'architecture MVC

C'est donc une architecture qui a prouvé son efficacité sur la durée.
Elle permet de créer des sites web de meilleure qualité et facile à maintenir.

D'ailleurs, la plupart des frameworks comme LARAVEL, SYMFONY, SPRING BOOT, DJANGO, NEST.JS... implémentent tous cette architecture.

C'est donc le type d'architecture qu'il faudra absolument connaître en tant que développeur Web.

3- L'architecture MVC

L'idée, c'est de morceler notre application en 3 grands blocs :

- le **modèle** (qui correspond à l'interaction avec la base de données **SQL**)
- la **vue** (qui correspond au contenu **HTML, CSS, JAVASCRIPT**)
- le **contrôleur** (qui correspond au contenu **PHP** faisant le lien entre les deux précédents).

3- L'architecture MVC

LE MODÈLE (SQL)

On y retrouve du code PHP qui manipule du code SQL.

Il gère les données du site comme :

- Les **accès** à la base de données.
- Il propose des fonctions pour faire des **Insert, Update, Delete, Select**.

Ses fonctions peuvent renvoyer des tableaux de données.

Les résultats seront exploités ensuite par le contrôleur.

3- L'architecture MVC

LE CONTRÔLEUR (PHP)

Le contrôleur est la page appelée (le véritable index).

Ses rôles principaux :

- Dès que le contrôleur a besoin d'interagir avec la base de données, il fait appel aux services du modèle.
- Le modèle à travers ses fonctions effectue la tâche et lui retourne des données si besoin

3- L'architecture MVC

LE CONTRÔLEUR (PHP)

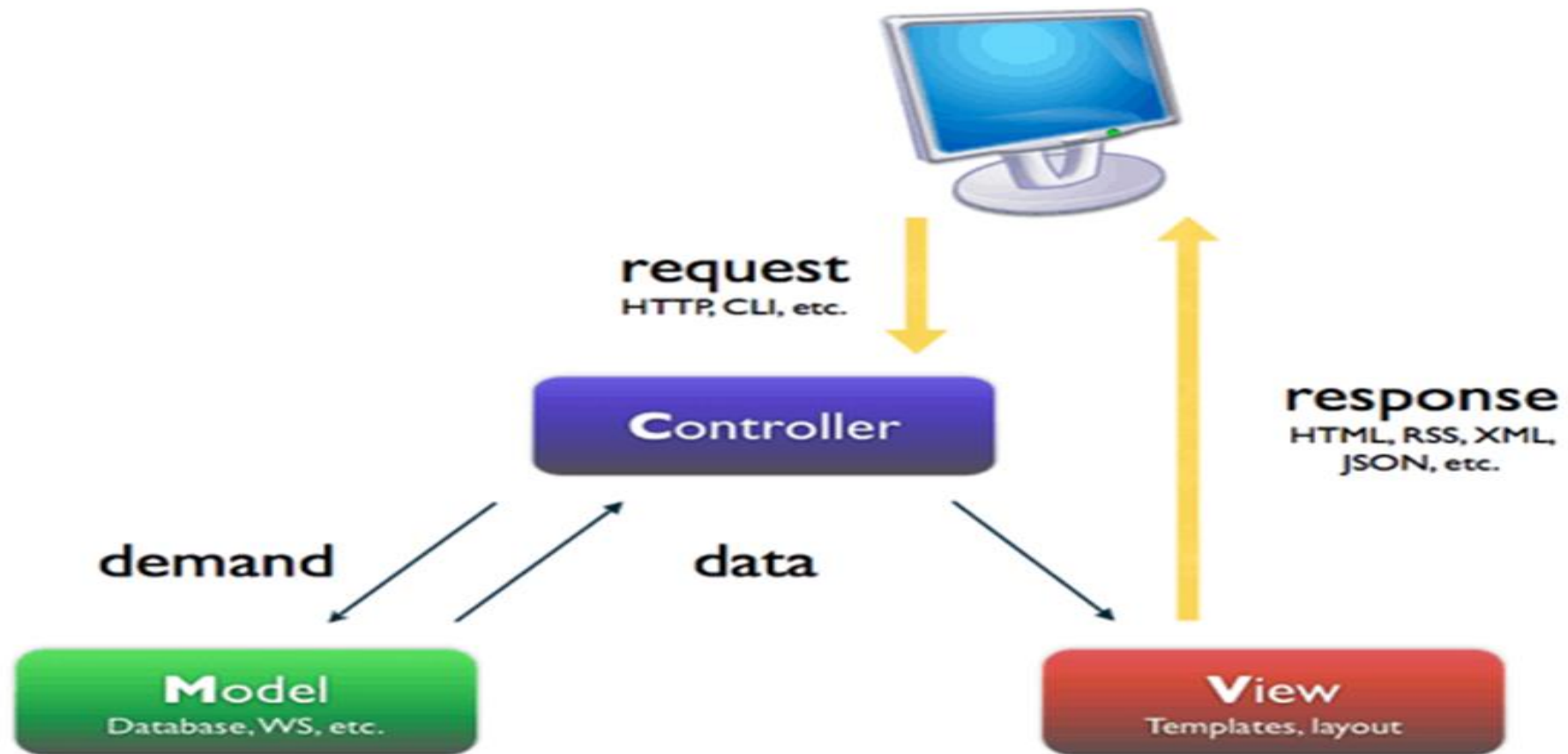
- Il analyse ensuite les données reçues du modèle
- Mais aussi celles passées en paramètre à l'appel de la page (\$_GET, \$_POST, \$_SESSION). Il détermine par exemple si le visiteur a le droit de voir la page ou non.
- En fonction de ses calculs, il appelle la bonne vue afin de construire et de retourner au client, la réponse correspondante à sa requête.

3- L'architecture MVC

LA VUE (HTML)

- La vue affiche la page HTML. Elle récupère des variables du contrôleur et/ou du Modèle pour savoir ce qu'elles doivent afficher dynamiquement.
- C'est donc une page HTML avec des boucles et conditions PHP très simples, pour afficher par exemple les tableaux de données issus du Modèle.

3- L'architecture MVC



3- L'architecture MVC

