

PHP - P00



Jobiz

studi

Déroulé

- 6 lives d'une heure sur la mise en place d'un projet en php P00.
- Prérequis :
 - Avoir les connaissances de php et les notions de classe/objet et du modèle MVC.
- Questions/Réponse à la moitié (25min) et fin du live (55min).
- Sources du projet sur github :
 - https://github.com/arirangz/studi_jobiz



Programme des 6 lives

- Présentation du projet
- Conception de la base de données
- Création de la base de données
- Mise en place de la structure MVC
- Prise en main de la structure MVC
- Création des classes Entity
- Création des classes Repository
- Mise en place des contrôleurs
- Gestion des vues
- Gestion des formulaires



Environnement technique

- PHP \geq 8.1 et MySQL \geq 5.7 (WAMP)
- Composer
- Tailwind
- VSCode



Besoin

- L'entreprise Jobiz vous contacte pour développer son application web.
- Vous devrez développer une plateforme dédiée aux chercheurs d'emploi et aux employeurs spécialisés dans l'informatique.
- Les utilisateurs peuvent parcourir les offres et postuler
- Une page liste des offres
 - Pouvoir filtrer par catégorie
 - Pouvoir filtrer les emplois par fourchette de salaire
 - Pouvoir filtrer par pays
- Page du détail d'une offre (show)
 - Afficher toutes les informations sur le poste
 - Ajouter un formulaire pour postuler à l'offre (utilisateurs connectés)
 - Champs textarea pour la lettre de motivation



Charte graphique

- Font: system-ui

- Colors:



slate-700

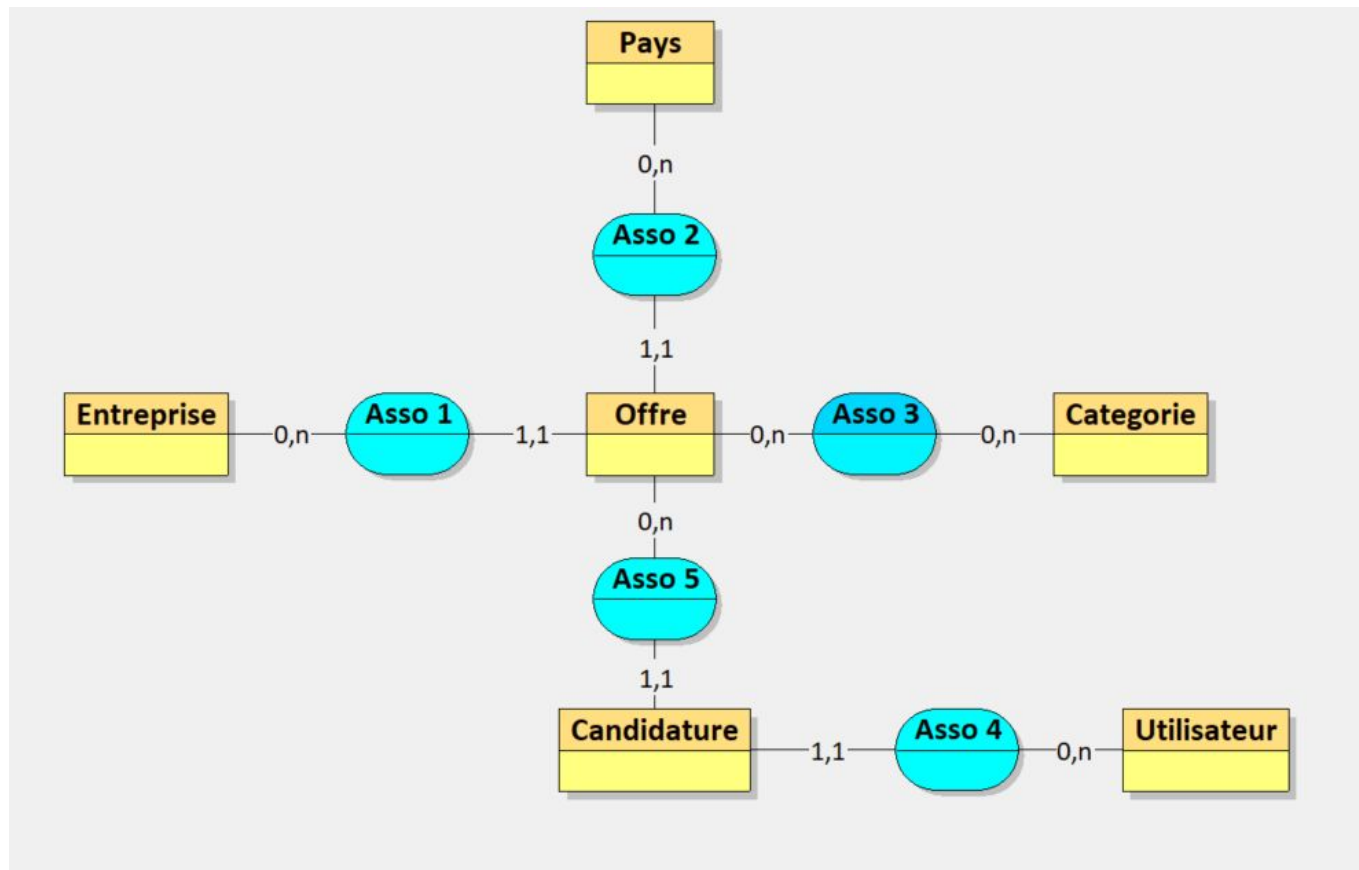


teal-600

- Logo :



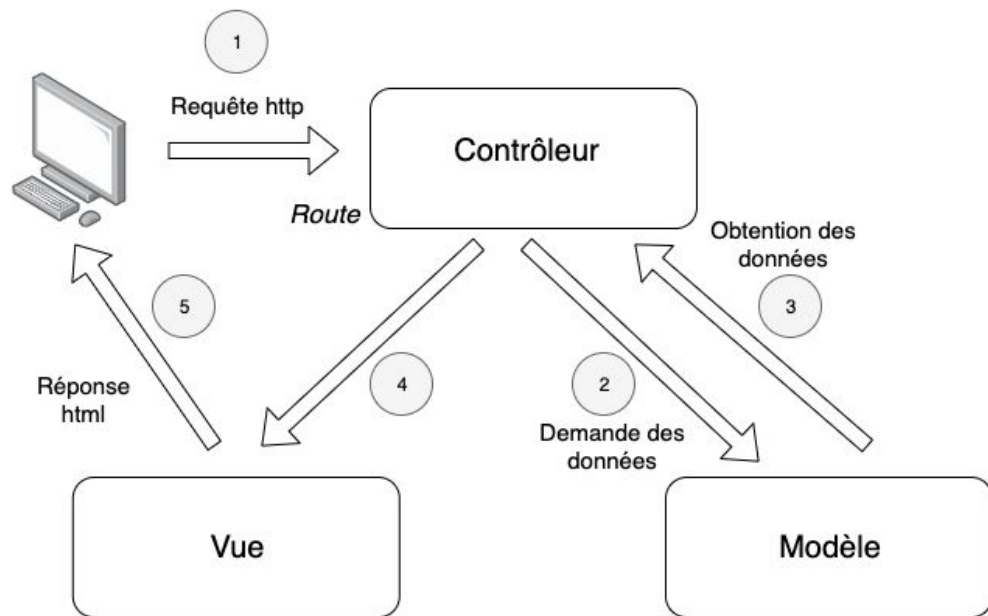
Conception de la base de données





MVC

MVC



MVC : Définition

Le modèle MVC (Modèle-Vue-Contrôleur) est un modèle de conception pour le développement de logiciels qui permet de **séparer les responsabilités** des différentes parties d'une application en trois composantes distinctes : **le modèle, la vue et le contrôleur**.



MVC

- **Modèle** : Le modèle **représente les données** de l'application et les règles métier associées. Il contient les méthodes qui permettent de manipuler ces données, ainsi que les fonctions de validation, de transformation et de persistance.
- **Vue** : La vue représente l'**interface utilisateur de l'application**, c'est-à-dire la manière dont les données sont présentées à l'utilisateur final. Elle est responsable de l'affichage des données, de leur mise en forme et de leur organisation.
- **Contrôleur** : Le contrôleur agit comme l'**intermédiaire** entre le modèle et la vue. Il gère les interactions utilisateur (requêtes), récupère les données du modèle et les transmet à la vue.

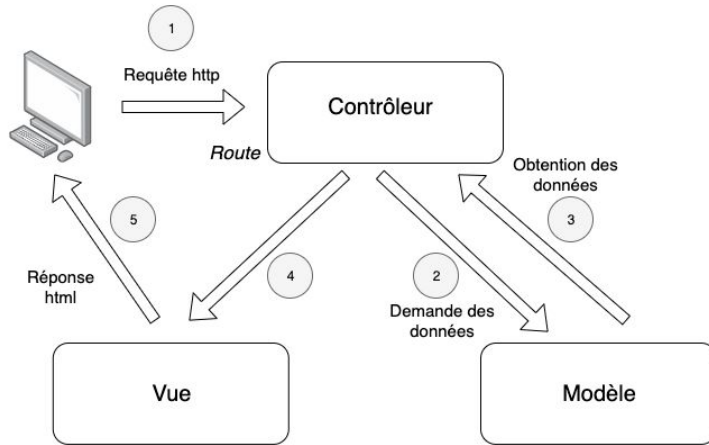


Importance de la séparation des responsabilités

- La séparation des responsabilités entre le modèle, la vue et le contrôleur est une pratique importante dans le développement web.
- Elle permet de **faciliter la conception, la maintenance et l'évolution** des applications, en favorisant la réutilisation et la modularité du code.
- Elle permet également d'**améliorer la lisibilité et la compréhensibilité du code**, en le découpant en composantes indépendantes et cohérentes.
- Enfin, elle **facilite la collaboration entre les développeurs**, en leur permettant de travailler sur des parties distinctes de l'application sans se marcher sur les pieds.



MVC



Exemple pour une page article

- 1. Un utilisateur demande un article
- 2. Le contrôleur `ArticleController.php` récupère les informations de la requête (ex: `id_article`) et fait la demande au modèle `Article.php` pour récupérer les données
- 3. `Article.php` retourne les données à `ArticleController.php`
- 4. `ArticleController.php` retourne les données à la vue et charge la vue `article.php` qui contient le html
- 5. La vue complète est retournée à l'utilisateur



DOMAINE LOCAL

Ajouter un domaine local : Etape 1

Sous windows :

Ouvrir le fichier C:\Windows\System32\drivers\etc\hosts

Ajouter une ligne vers votre domaine local.

ex:

127.0.0.1 monprojet.local

Sous mac, suivre les instructions :

<https://dev.to/crankysparrow/configuring-virtual-hosts-with-mamp-f3i>



Ajouter un domaine local : Etape 2

Nous allons ensuite devoir indiquer à apache vers quel dossier le domaine doit pointer.

Pour cela il faut modifier le fichier httpd-vhosts.conf

- Sous **Wamp**
C:\wamp64\bin\apache\apache2.4.51\conf\extra\httpd-vhosts.conf
- Sous **Xampp**
C:\xampp\apache\conf\extra\httpd-vhosts.conf
- Sous **Mamp**
/Applications/MAMP/conf/apache/extra/httpd-vhosts.conf

et ajouter la configuration en adaptant les informations

```
<VirtualHost *:80>  
    DocumentRoot "C:\wamp64\www\mon_projet\public"  
    ServerName monprojet.local  
</VirtualHost>
```

Il faut ensuite redémarrer le serveur



Composer

Pour l'autoload et les namespaces, on va utiliser composer <https://getcomposer.org/>.

On va ajouter un fichier composer.json à la racine

```
{
  "autoload": {
    "psr-4": {
      "App\\": "src/"
    }
  }
}
```

Puis on va exécuter la commande suivante :

```
composer dump-autoload
```

