LP PRISM
DÉVELOPPEMENT WEB

Programmer en PHP côté serveur

ARCHITECTURE MVC

Restructuration du code selon le design pattern MVC

Le design pattern MVC

Les fichiers PHP codés jusqu'à maintenant sont un mélange de beaucoup de choses…

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Le design pattern MVC

Les fichiers PHP codés jusqu'à maintenant sont un mélange de beaucoup de choses…

- des langages :
  - du code PHP,
  - du code html / css.

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Le design pattern MVC

Les fichiers PHP codés jusqu'à maintenant sont un mélange de beaucoup de choses…

- des langages :
  - du code PHP,
  - du code html / css.

- des instructions de types divers :
  - insertions de fichiers PHP (`auteur.php`, etc)
  - requêtes SQL écrites en PHP,
  - « calculs » divers sur des variables PHP,
  - affichages divers par des echo PHP.

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Le design pattern MVC

Plus votre code est volumineux, et plus ce mélange est illisible.
Il arrive forcément un point critique où vous devrez ORGANISER votre code.

C'est là qu'interviennent les design patterns.
Ils proposent une organisation précise du code.

Le design pattern classique en PHP est le design **MVC**.

M = MODELE ou MODEL
V = VUE ou VIEW
C = CONTRÔLEUR ou CONTROLLER

Le design pattern MVC

- Les fichiers de type **MODELE** (MODEL) sont traditionnellement **ceux qui définissent les classes**, comme le fichier `auteur.php`, et au sein desquels on va trouver le codage des méthodes **requêtes** sur la base de données.

auteur.php

```php
<?php
class Auteur {

// attributs d'un auteur
  private $numAuteur;
  private $nom;
  private $prenom;
  private $nationalite;
  private $anneeNaissance;
```

```php
// méthode static qui retourne les auteurs en un tableau d'objets
public static function getAllAuteurs() {
  // écriture de la requête
  $requete = "SELECT * FROM auteur;";
  // envoi de la requête et stockage de la réponse
  $resultat = Connexion::pdo()->query($requete);
  // traitement de la réponse
  $resultat->setFetchmode(PDO::FETCH_CLASS,'Auteur');
  $tableau = $resultat->fetchAll();
  return $tableau;
}
```

Le design pattern MVC

- Les fichiers de type **VUE** (VIEW) ont pour mission **d'afficher du contenu html**. Le contenu affiché est souvent transporté dans des variables PHP, crées par le CONTROLEUR, et qui ne servent qu'à l'affichage. Ce qui est dans les fichiers **VUE** sera affiché sur le **navigateur** de l'internaute.

vue.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
  // affichage du lien pour créer un nouvel auteur
  echo $lienC;
?>
</body>
</html>
```

Le design pattern MVC

- Les fichiers de type **CONTROLEUR** (CONTROLLER) dirigent la manœuvre. Par exemple :
  - ✓ Ils **récupèrent des informations** dans l'url, en provenance du client navigateur,
  - ✓ Ils **convoquent** la classe MODEL nécessaire (Connexion, Auteur, etc),
  - ✓ Ils **demandent** à la classe en question **de récupérer des données** de la base,
  - ✓ Ils **traitent** ces données (calculs divers),
  - ✓ Ils **demandent** aux fichiers VUE

    d'afficher de bonne façon ce traitement.

controleur.php

```php
<?php
  // insertion des classes
  require_once("modele.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $lesAuteurs = Auteur::getAllAuteurs();
  // affichage de tous les auteurs
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  require_once("vue.php");
?>
```

Réorganisation d'un fichier PHP

Ce fichier est construit naïvement. Tout y est mélangé :

- du code html,
- des requêtes écrites en PHP,
- des calculs sur des variables PHP,...

Il est important de tout réorganiser !

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Réorganisation d'un fichier PHP

Toutes les lignes sélectionnées sont des lignes de requêtes sur la base de données.

Ces lignes sont donc à incorporer dans un fichier **MODELE**

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Réorganisation d'un fichier PHP

Toutes les lignes sélectionnées sont :

- soit des lignes de code html/css
- soit des echo PHP qui affichent un résultat visible dans le navigateur.

Ces lignes sont donc à incorporer dans un fichier **VUE**

lesAuteurs.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Réorganisation d'un fichier PHP

Toutes les lignes sélectionnées sont :

- soit des lignes pour insérer des classes PHP,
- soit des lignes de calcul de variables.

Ces lignes sont donc à incorporer dans un fichier **CONTROLEUR**

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Réorganisation d'un fichier PHP

On va donc séparer le code en trois fichiers **M**, **V**, **C**:

- Un fichier **MODELE** qui se chargera de la requête,
- Un fichier **VUE** qui affichera tout ce que le contrôleur lui demandera d'afficher,
- Un fichier **CONTROLEUR** qui intégrera les fichiers MODELE, fera des calculs divers, et demandera au fichier VUE d'afficher les variables.

lesAuteurs.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // insertion des classes
  require_once("auteur.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $requete = "SELECT * FROM auteur;";
  $reponse = Connexion::pdo()->query($requete);
  $reponse->setFetchMode(PDO::FETCH_CLASS,'Auteur');
  $lesAuteurs = $reponse->fetchAll();
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
// affichage du lien pour créer un nouvel auteur
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  echo $lienC;
?>
</body>
</html>
```

Les fichiers MODELE, VUE et CONTROLEUR

vue.php

```php
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>tous les auteurs</title>
</head>
<body>
<?php
  // affichage de tous les auteurs
  echo "<h3>Liste des auteurs de la base de données</h3>";
  echo "<ul>";
  foreach ($lesAuteurs as $unAtueur) {
    $unAuteur->afficher();
  }
  echo "</ul>";
  // affichage du lien pour créer un nouvel auteur
  echo $lienC;
?>
</body>
</html>
```

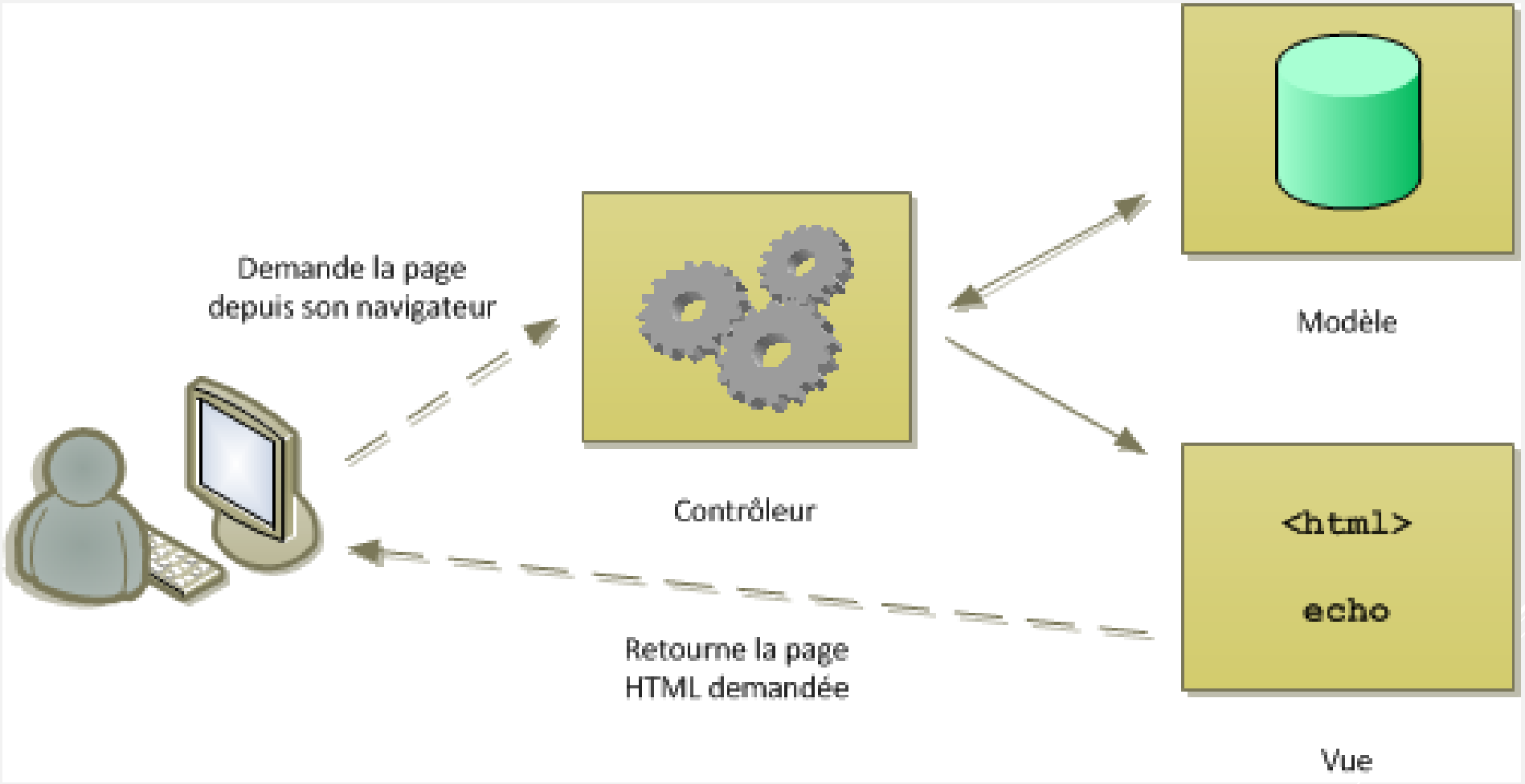
controleur.php

```php
<?php
  // insertion des classes
  require_once("modele.php");
  require_once("connexion.php");
  // connexion à la base
  Connexion::connect();
  // récupération de tous les auteurs
  $lesAuteurs = Auteur::getAllAuteurs();
  // affichage de tous les auteurs
  $lienC = "<a href='formulaireCreationAuteur.html'> créer un auteur </a>";
  require_once("vue.php");
?>
```

modele.php

```php
<?php
class Auteur {
  ...
  // méthode static qui retourne les auteurs en un tableau d'objets
  public static function getAllAuteurs() {
    // écriture de la requête
    $requete = "SELECT * FROM auteur;";
    // envoi de la requête et stockage de la réponse
    $resultat = Connexion::pdo()->query($requete);
    // traitement de la réponse
    $resultat->setFetchmode(PDO::FETCH_CLASS,'Auteur');
    $tableau = $resultat->fetchAll();
    return $tableau;
  }
}
?>
```

Schéma de l'architecture MVC



Source : https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc