

La Vue

Dans un modèle MVC, la vue est tout ce que l'utilisateur voit à l'écran. Dans notre cas, ce sera du HTML et CSS.

Un fichier php avec du HTML à l'intérieur, est nommé un « template » (gabarit). Ces « template » peuvent être affichés à l'écran en utilisant l'objet de la class « View ».

Cette classe a pour unique rôle, d'afficher un « template » php

1. Créer notre premier « template »

Nous allons tout d'abord créer un dossier « templates » à la racine de notre projet. Ce dossier va contenir tout les « templates » (pages).

Nous allons créer un premier template très simple : "hello.php"

```
<h1>Hello <?= $name ?></h1>
```

2. Créer la classe « View »

Cette classe va permettre d'afficher sur notre navigateur le contenu d'un templates. Cependant les templates sont rangés dans un dossier. Il faudra au préalable spécifier le chemin de ce dossier dans son constructeur.

Cette classe possède une seule méthode : **render**. Son fonctionnement est assez simple, nous lui donnons le nom du template à afficher ainsi qu'un tableau de variables et la méthode retournera une chaîne de caractères contenant l'intégralité du fichier php compilé.

Cette classe va être rangée dans le répertoire `src`, mais attention nous allons créer un autre répertoire dans `src` : `Core` (noyau). Ce répertoire `Core` contiendra toutes les classes nécessaires au bon fonctionnement de notre MVC.

À l'aide des fonctions suivantes :

- `file_exists` : Test si un fichier existe
- `ob_start` : Démarre un « Buffer » d'affichage. Pour résumer lorsque l'on va faire un « echo » ce dernier ne s'affichera pas à l'écran mais sera écrit dans la mémoire (dans un Buffer).
- `extract` : Permet de créer des variables en fonction des clés d'un tableau.
- `include` : Nous allons inclure le fichier template demandé.
- `ob_get_clean` : récupère tout ce qui était affiché (echo).

Exemple d'utilisation de la class View :

```
use App\Core\View;  
  
$view = new View(__DIR__ . '/../templates');  
  
echo $view->render('hello', ['name' => 'John Doe']);
```

Vous pouvez tester cette objet vue à l'intérieur du frontend controller : `public/index.php`.