



# Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ [Petite application version 2](#) ▶ [Le front controller](#)

## Le front controller

Le script index.php du répertoire public constituera notre front controller, c'est-à-dire le point d'entrée de notre application. Ce front controller aura pour rôle de déterminer à partir de l'URL quel controller doit être instancié et quelle méthode de ce controller doit être appelée. Si une exception survient, le front controller captera cette exception dans un bloc try/catch et inclura la vue d'erreur.

```
1 <?php
2
3 // public/index.php
4 include '../setup.php';
5
6 use util\Router;
7
8 $request_uri = $_SERVER['REQUEST_URI'];
9 $route = Router::getRoute($request_uri);
10 if ($route == null) {
11     include VIEW . '/index.php';
12 } else {
13     try {
14         $controller_name = $route->getController();
15         $action_name = $route->getAction();
16         $class_name = "controller\\"$controller_name";
17         $class = new ReflectionClass($class_name);
18         $controller = $class->newInstance();
19         $method = $class->getMethod("$action_name");
20         $method->invoke($controller);
21     } catch (Exception $ex) {
22         include VIEW . '/errorPage.html';
23     }
24 }
```

La partie de l'URL qui servira de chemin est passée en argument de la méthode de classe Router::getRoute(). Cette méthode renvoie un objet de type Route ou null. L'objet Route permet de récupérer le nom qualifié du controller à instancier. Cette instanciation se fera via des méthodes dites d'introspection ou de "reflection". La classe ReflectionClass va nous permettre de récupérer un objet représentant la classe dont le nom est passé en argument (ici ce sera le nom qualifié de notre controller). Grâce à cette instance de ReflectionClass, représentant la classe de notre controller, nous allons pouvoir instancier un controller via la méthode newInstance() de ReflectionClass. Cette méthode newInstance() va donc renvoyer une instance de la classe du controller dont le nom qualifié a été passé en argument du constructeur de ReflectionClass.

La méthode getMethod() de la classe ReflectionClass nous permet, elle, de récupérer un objet qui représente une méthode. À partir de cet objet nous allons pouvoir invoquer cette méthode sur une instance du controller.

Par exemple, si l'URL saisie par l'utilisateur est http://localhost:8080/personnes, alors la valeur de \$request\_uri sera "/personnes". La méthode Router::getRoute() nous enverra donc un objet Route correspondant au path "/personnes". Cet objet Route aura donc en attribut controller la valeur "PersonneController" et en attribut action la valeur "getAllPersonnes".

Le nom qualifié de la classe du controller sera donc "controller\PersonneController" et sera assigné à la variable class\_name. Nous pourrons alors récupérer un objet de type ReflectionClass qui représentera la classe controller\PersonneController et instancier une classe de type controller\PersonneController grâce à la méthode newInstance() de ReflectionClass.

L'attribut action de l'objet Route et la méthode getMethod de ReflectionClass nous permettront de récupérer un

objet représentant la méthode `getAllPersonnes()`. Nous pourrions ensuite invoquer cette méthode sur l'instance du contrôleur que nous avons précédemment créée.

L'`index.php` représente notre point d'entrée, il est donc nécessaire que les requêtes soient redirigées vers ce front controller. Le script `router.php` du répertoire web se chargera de cette redirection pour nous, mais en production il sera nécessaire de créer un fichier `.htaccess` afin d'utiliser la ré-écriture d'URL et/ou de créer un hôte virtuel.

Fin

## NAVIGATION



[Accueil](#)

■ [Ma page](#)

[Pages du site](#)

[Mon profil](#)

[Cours actuel](#)

[POO PHP](#)

[Participants](#)

[Généralités](#)

[La programmation orientée objet : premiers pas](#)

[L'héritage](#)

[Les interfaces](#)

[Le typage](#)

[Les namespaces](#)

[Les exceptions](#)

[Les bases de données avec PDO](#)

[Les tests avec PHPUnit](#)

[Petite application version 2](#)

[Le projet](#)

[conf.ini et setup.php](#)

[Les entités et la dao](#)

[Test de la DAO](#)

[La gestion des routes](#)

[Test des routes](#)

[Les contrôleurs](#)

[Les vues](#)

**[Le front controller](#)**

[Test de l'ajout d'une personne](#)

[T.P. mettre à jour une personne](#)

[Petite application version 3](#)

[Mes cours](#)

## ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[POO PHP](#)