

TD PHP MVC

Étape 5 : Actions d'un controller

Revenons sur notre HomeController. Nous commençons par récupérer les éventuels paramètres passés par le router. Le premier sera une action, le second correspondra à un éventuel id.

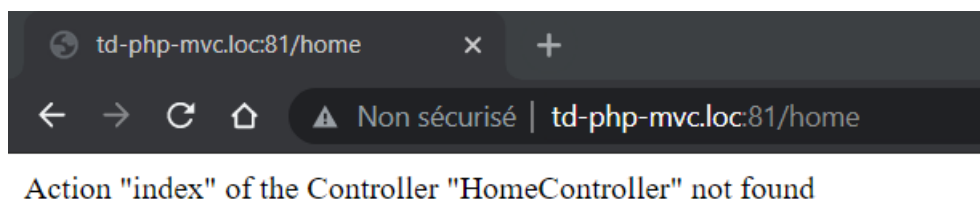
Si aucune action n'est définie dans la route nous choisissons d'exécuter par défaut la fonction index. Si aucun id n'est défini dans la route nous le mettons à null. Nous stockons ces informations dans des attributs de la classe (\$this->...)

Puis avant d'exécuter une fonction de notre classe HomeController, nous vérifions que celle-ci existe :-). Si elle n'existe pas nous arrêtons le script avec un message d'erreur.

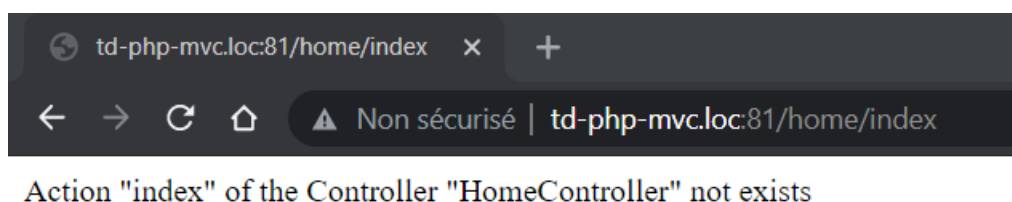
> home.controller.php

```
class HomeController{  
    function __construct($params){  
        $this->action = $params[0] ?? 'index';  
        $this->id = $params[1] ?? null;  
  
        if(!method_exists(get_called_class(), $this->action)){  
            $controllerName = get_called_class();  
            die("Action \"{$this->action}\" of the Controller \"{$controllerName}\" not exists");  
        }  
    }  
}
```

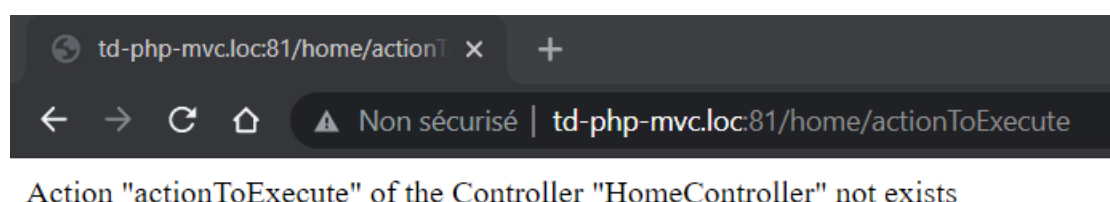
Premier test avec la route /home



avec la route /home/index



avec la route /home/actionToExecute



PHP `method_exists()` → <https://www.php.net/manual/en/function.method-exists.php>

PHP `get_called_class()` → <https://www.php.net/manual/en/function.get-called-class.php>

Aucune de ces fonctions (`index`, `actionToExecute`) n'existent dans le `HomeController`, nous allons donc créer l'action par défaut : la fonction `index()`

Si aucune action n'est spécifiée dans la route (c'est le cas pour la route `/home`), c'est cette fonction qui sera exécutée.

Puis nous appelons cette fonction dans le constructeur de la classe `$this->{$this->action}()`;

> `home.controller.php`

```
if(!method_exists(get_called_class(), $this->action)){
    $controllerName = get_called_class();
    die("Action \"{$this->action}\" of the Controller \"{$controllerName}\" not exists");
}

$this->{$this->action}();
}

function index(){
    $controllerName = get_called_class();
    $this->content = "Action \"{$this->action}\" of the Controller \"{$controllerName}\" executed";
}
```

A ce stade nous ne faisons que stocker un message dans un attribut de `HomeController` (`$this->content`)

Pour afficher ce contenu dans notre page `index.php`, nous allons créer une fonction `render` dans la classe `Router`

> `router.php`

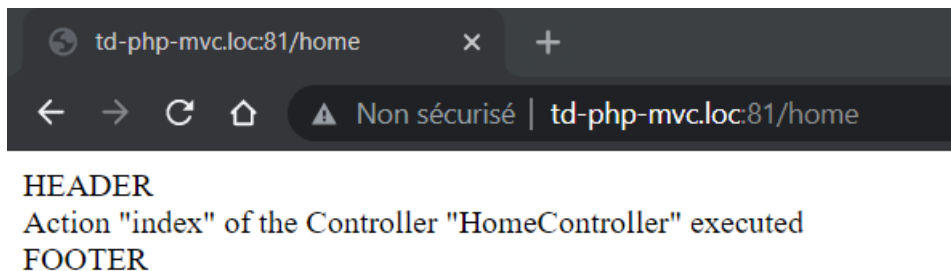
```
function render(){
    return $this->controller->content;
}
```

Puis nous appelons cette fonction `render` à l'endroit où nous voulons afficher le contenu dans la page `index.php` (voir ci-dessous)

> index.php

```
index.php
1  <?php
2
3  $route = $_SERVER["REQUEST_URI"];
4
5  require_once 'router.php';
6  $router = new Router($route);
7
8  ?>
9
10 <header>
11 |     HEADER
12 </header>
13
14 <main>
15 |     <?= $router->render() ?>
16 </main>
17
18 <footer>
19 |     FOOTER
20 </footer>
```

Résultat pour la route /home ou home/index:

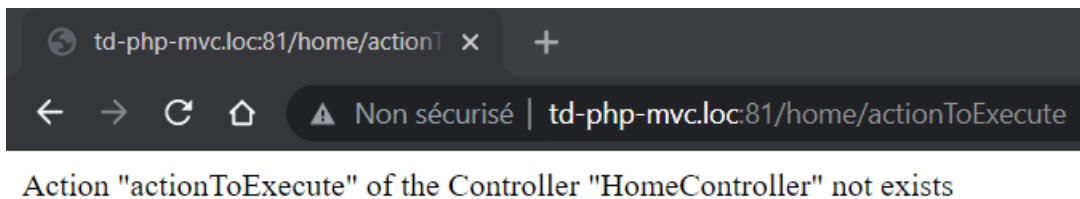


td-php-mvc.loc:81/home

Non sécurisé | td-php-mvc.loc:81/home

HEADER
Action "index" of the Controller "HomeController" executed
FOOTER

Résultat pour la route /home/actionToExecute



td-php-mvc.loc:81/home/actionToExecute

Non sécurisé | td-php-mvc.loc:81/home/actionToExecute

Action "actionToExecute" of the Controller "HomeController" not exists

Normal, la fonction actionToExecute n'existe pas dans HomeController