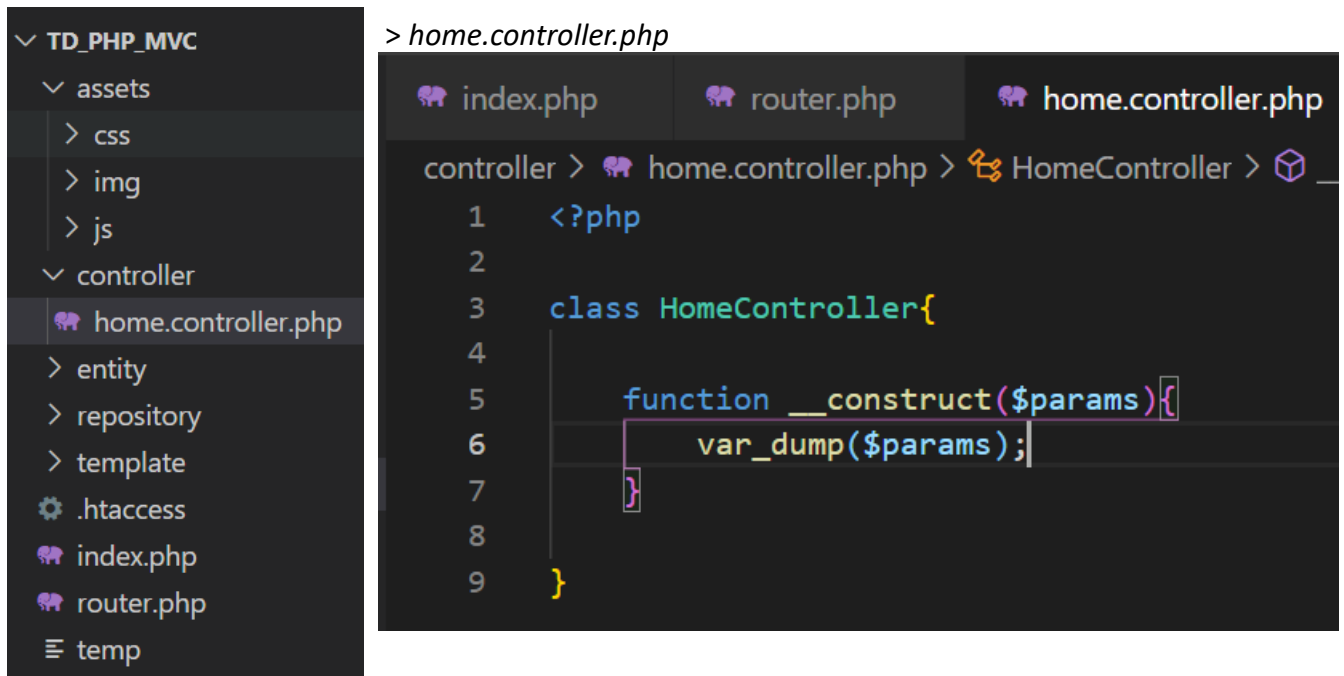


## TD PHP MVC

### Étape 4 : Un premier controller

Nous allons créer le premier controller de notre application, celui correspondant à la page d'accueil. La route sera /home, la classe doit donc s'appeler HomeController et le fichier contenant cette classe home.controller.php

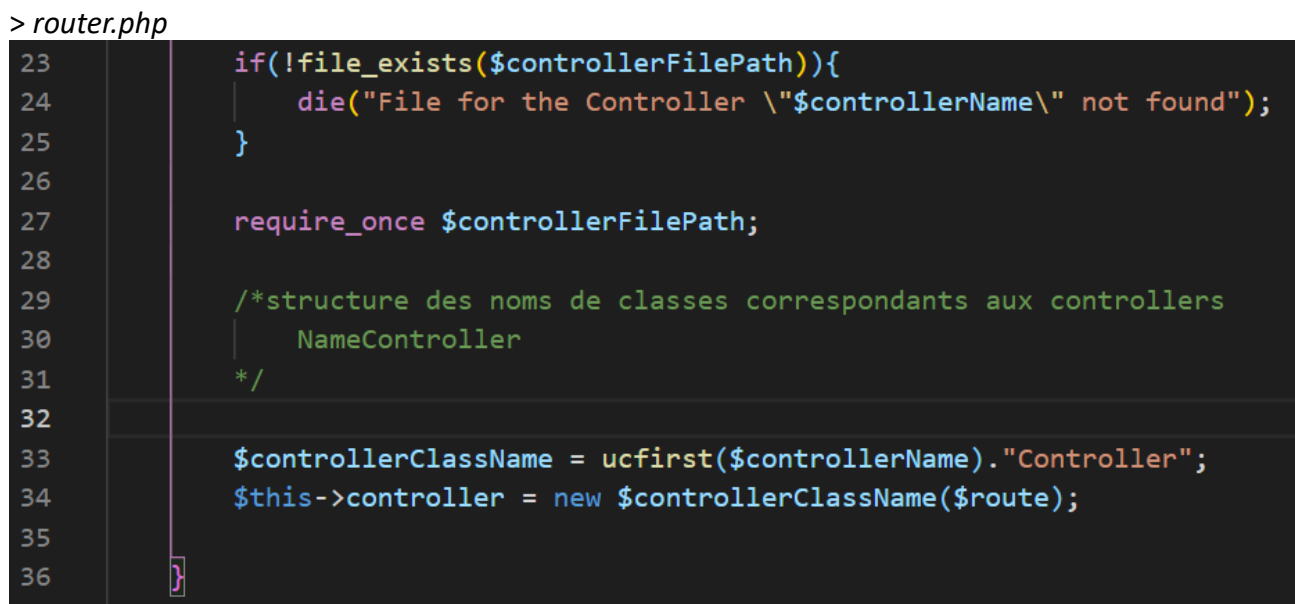


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'TD\_PHP\_MVC' with subdirectories 'assets' (containing 'css', 'img', 'js') and 'controller' (containing 'home.controller.php'). Other files include 'entity', 'repository', 'template', '.htaccess', 'index.php', 'router.php', and 'temp'. The code editor on the right shows the content of 'home.controller.php' with the following code:

```
> home.controller.php

index.php router.php home.controller.php
controller > home.controller.php > HomeController > _
1  <?php
2
3  class HomeController{
4
5      function __construct($params){
6          var_dump($params);
7      }
8
9  }
```

Nous pouvons maintenant instancier notre classe HomeController dans le router (Il faut au préalable l'importer avec require\_once) et lui passer en paramètre le tableau \$route



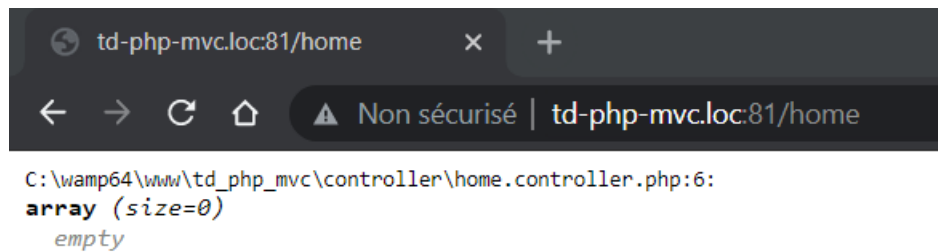
The screenshot shows a code editor with the code for 'router.php'. The code is as follows:

```
> router.php
23  if(!file_exists($controllerFilePath)){
24      die("File for the Controller \"$controllerName\" not found");
25  }
26
27  require_once $controllerFilePath;
28
29  /*structure des noms de classes correspondants aux controllers
30      NameController
31  */
32
33  $controllerClassName = ucfirst($controllerName). "Controller";
34  $this->controller = new $controllerClassName($route);
35
36  }
```

PHP `file_exists()` → [https://www.w3schools.com/php/func\\_filesystem\\_file\\_exists.asp](https://www.w3schools.com/php/func_filesystem_file_exists.asp)

PHP `ucfirst()` → [https://www.w3schools.com/php/func\\_string\\_ucfirst.asp](https://www.w3schools.com/php/func_string_ucfirst.asp)

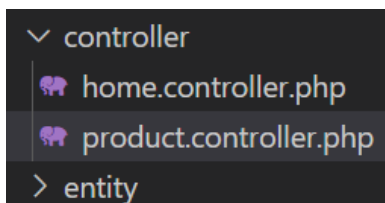
Résultat :



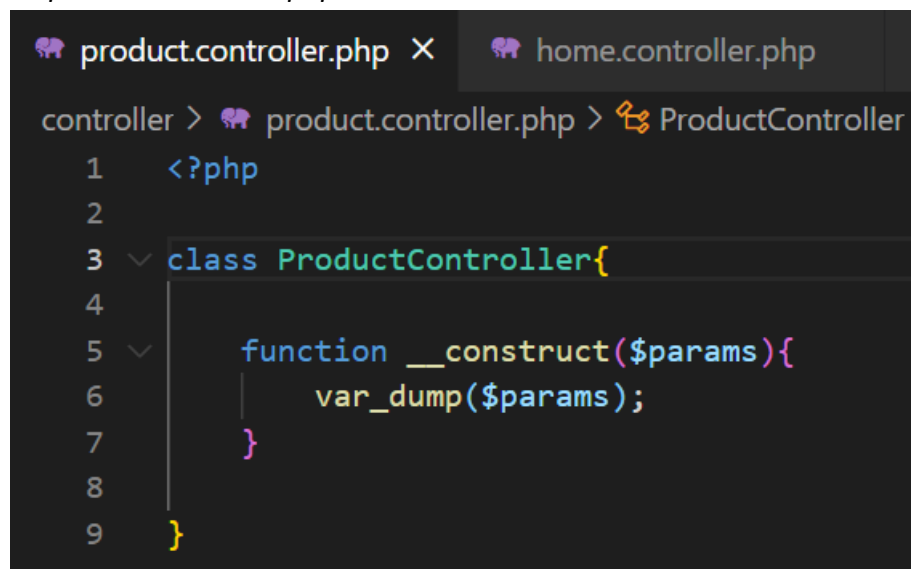
```
td-php-mvc.loc:81/home  
Non sécurisé | td-php-mvc.loc:81/home  
C:\wamp64\www\td_php_mvc\controller\home.controller.php:6:  
array (size=0)  
empty
```

ici, le chemin saisi dans l'url ne contient qu'une partie (correspondant au nom du controller), le tableau route que nous affichons ne contient donc plus rien.

Créons un 2ème controller pour la route /product

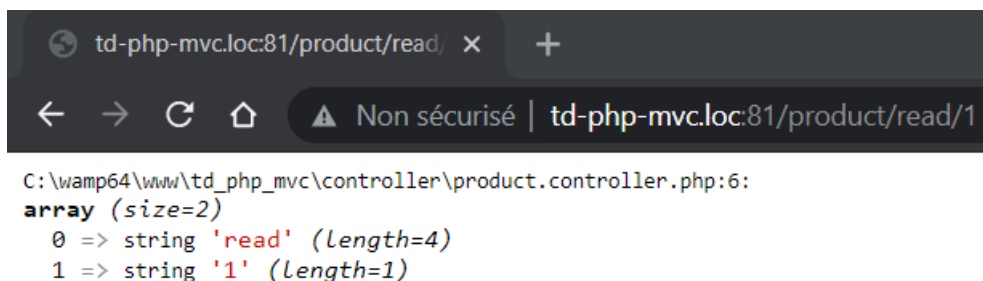


> product.controller.php



```
product.controller.php X home.controller.php  
controller > product.controller.php > ProductController  
1 <?php  
2  
3 class ProductController{  
4  
5     function __construct($params){  
6         var_dump($params);  
7     }  
8  
9 }
```

Résultat : (en saisissant comme chemin /product/read/1)



```
td-php-mvc.loc:81/product/read/1 X  
Non sécurisé | td-php-mvc.loc:81/product/read/1  
C:\wamp64\www\td_php_mvc\controller\product.controller.php:6:  
array (size=2)  
0 => string 'read' (length=4)  
1 => string '1' (length=1)
```

Nous affichons \$params (\$route du router) qui contient les 2ème et 3ème parties du chemin saisi. Ces paramètres vont nous permettre de définir une action (fonction) à exécuter dans le controller et d'avoir éventuellement un id