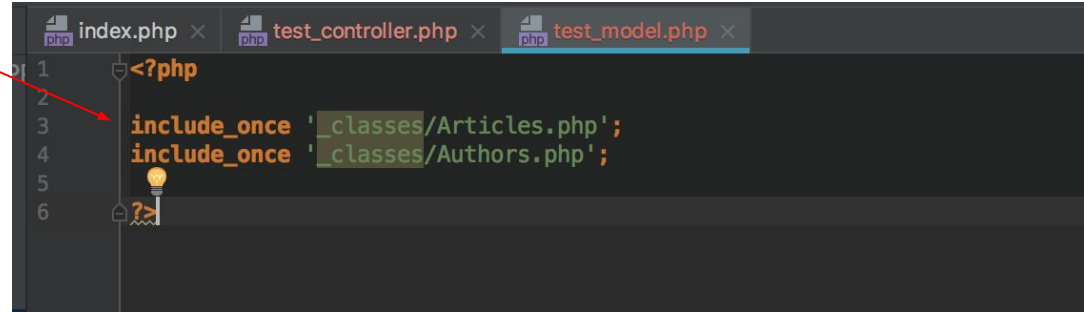


Pour utiliser des fonctions
statiques dans le contrôleur d'un
fichier.php il faut au préalable les
inclure dans le model.php



```
1 <?php
2
3 $allArticle = Articles::getAllArticles();
4 $allAuthors = Authors::getAllAuthors();
5
6
7
8 ?>
```

This screenshot shows the 'test_controller.php' file in a code editor. It contains two lines of PHP code that call static methods from the 'Articles' and 'Authors' classes. A red arrow points from the text 'il faut au préalable les inclure dans le model.php' to the first line of code.

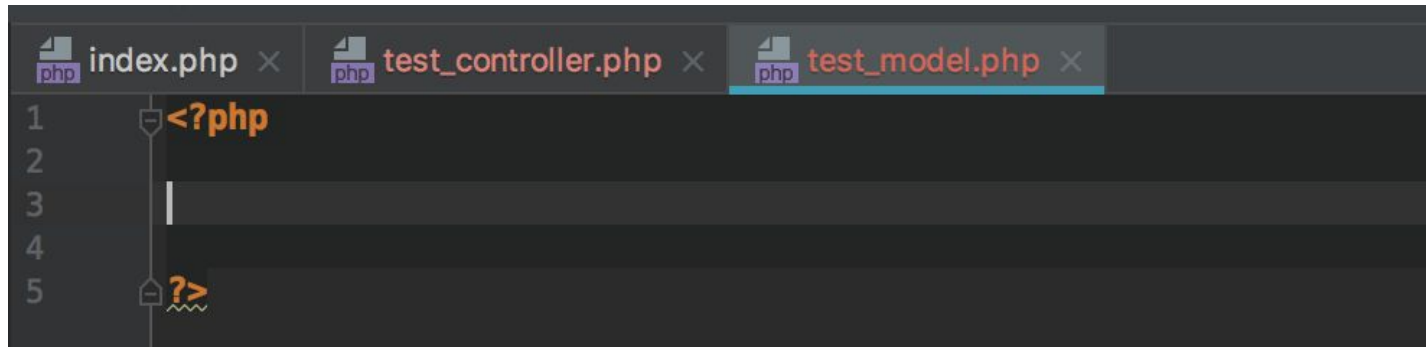


```
1 <?php
2
3 include_once '_classes/Articles.php';
4 include_once '_classes/Authors.php';
5
6 ?>
```

This screenshot shows the 'test_model.php' file in a code editor. It contains two lines of PHP code that use 'include_once' to load the 'Articles.php' and 'Authors.php' files. A red arrow points from the text 'il faut au préalable les inclure dans le model.php' to the first 'include_once' line.

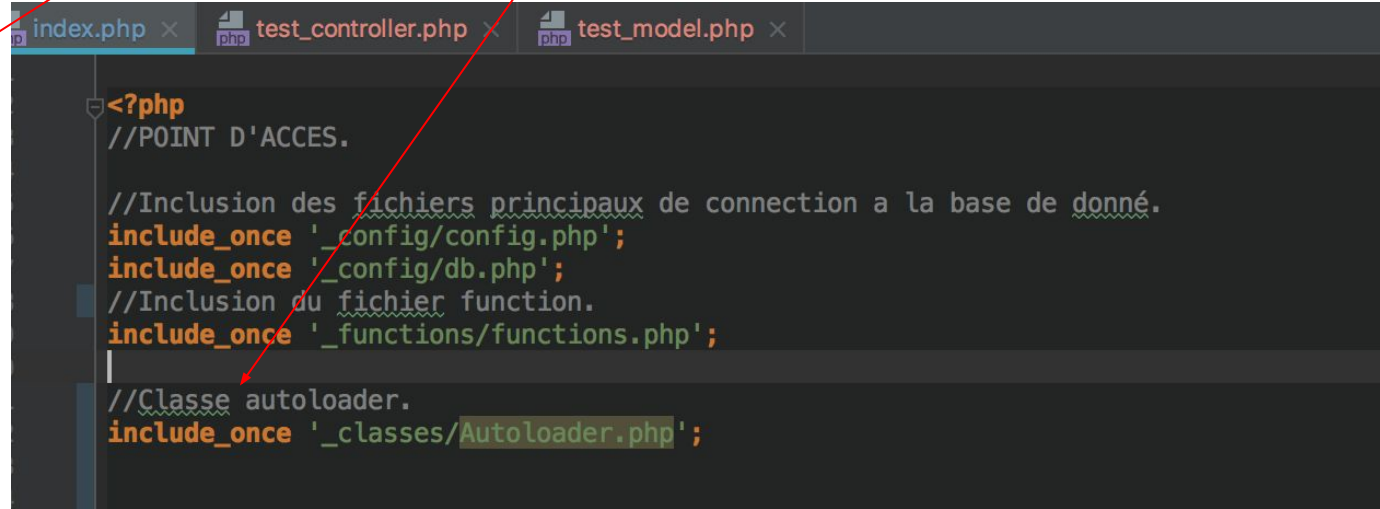
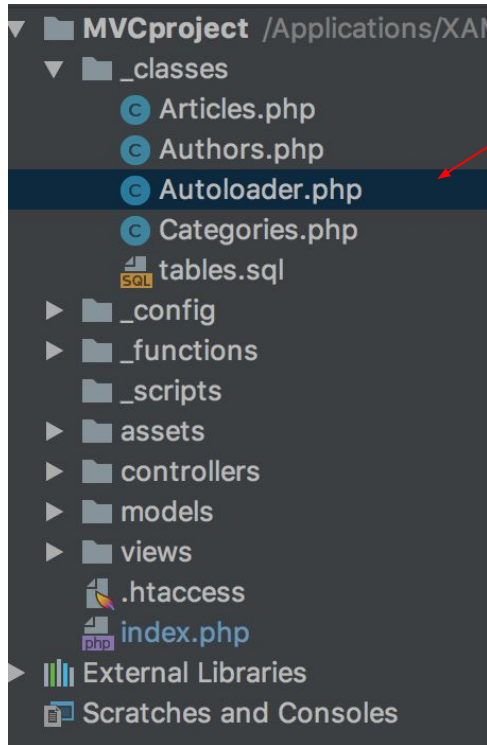
SAUF SI ON UTILISE UN AUTOLOADER CE QUE L'ON VA VOIR PAR LA SUITE.

On retire les include du modèle



```
1  <?php
2
3
4
5  ?>
```

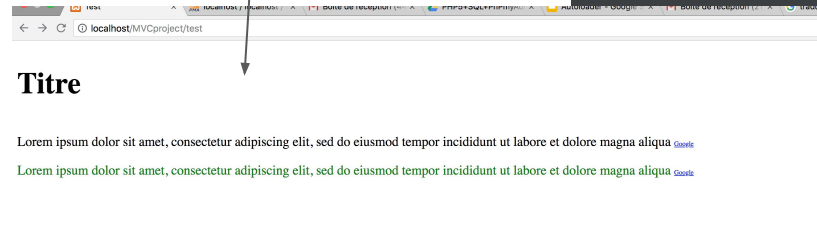
On va créer un nouveau fichier Autoloader.php et l'inclure a index.php



la fonction register va simplement exécuter la fonction spl_autoload_register() pour ajouter notre autoloader.

```
index.php x Autoloader.php x test_controller.php x test_model.php x
1 <?php
2
3
4 class Autoloader{
5
6
7 /**
8  * Ajout de l'autoloader
9  */
10 static function register(){
11     //Fonction d'autoload qui charge les classes sans avoir à les inclure dans le model.
12     spl_autoload_register(function ($class){
13         include_once '_classes/'.$class.'.php';
14     });
15 }
16
17 }
```

On appelle ensuite notre fonction static, qui va charger nos classes de façon automatique, et cela fonctionne.



The screenshot shows an IDE with the MVCproject file structure on the left and the index.php file open in the editor. The file structure includes folders for _classes, _config, _functions, _scripts, assets, controllers, models, and views, along with files like Articles.php, Authors.php, Autoloader.php, Categories.php, tables.sql, .htaccess, and index.php. The index.php file contains PHP code for a web application, including database connections, autoloader registration, and page routing logic.

```
1 <?php
2 //POINT D'ACCES.
3
4 //Inclusion des fichiers principaux de connection a la base de données.
5 include_once '_config/config.php';
6 include_once '_config/db.php';
7 //Inclusion du fichier function.
8 include_once '_functions/functions.php';
9 //Classe autoloader.
10 include_once '_classes/Autoloader.php';
11 Autoloader::register();
12
13
14
15
16
17 //Définition de la page courante.
18 if(isset($_GET['page']) and !empty($_GET['page'])){
19     $page = trim(strtolower($_GET['page']));
20 }else{
21     $page = 'home';
22 }
23
24 //Array contenant toutes les pages.
25 $allPages = scandir( directory: 'controllers/');
26 //vérification de l'existence de la page
27 if (in_array( needle: $page.'_controller.php', $allPages)){
28     //inclusion de la page.
29     include_once 'models/'.$page.'_model.php';
30     include_once 'controllers/'.$page.'_controller.php';
31     include_once 'views/'.$page.'_view.php';
32 } else{
33     include_once 'models/error_model.php';
34     include_once 'controllers/error_controller.php';
35     include_once 'views/error_view.php';
36 }
37
38
```