



Programmation orientée objet en PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [POO PHP](#) ► [Petite application version 2](#) ► [Les contrôleurs](#)

Les contrôleurs

[Les contrôleurs](#) porteront le namespace controller et seront placés dans le répertoire controller.

La classe IndexController possédera une méthode indexAction() dont le rôle sera simplement d'inclure la vue d'accueil.

```
1 <?php
2
3 // controller/IndexController.php
4
5 namespace controller;
6
7
8 class IndexController {
9
10     public function indexAction() {
11         include VIEW . '/index.php';
12     }
13
14 }
```

La classe PersonneController possédera un attribut d'instance dao qui sera initialisé avec un objet de type MysqlDao dans le constructeur. Le controller possédera deux méthodes : getAllPersonnes() et addPersonne(). La méthode getAllPersonnes() récupérera l'ensemble des personnes via la méthode getAllPersonnes() du dao et inclura la vue d'affichage des personnes.

La fonction addPersonne() inclura la vue d'ajout de personne (le formulaire) si la requête est en GET, si la requête est en POST, la méthode ajoutera la personne via la méthode addPersonne() de MysqlDao et appellera la méthode getAllPersonnes() du controller.

Les exceptions qui pourraient être déclenchées dans ces méthodes seront propagées aux méthodes appelantes.

```
1 <?php
2
3 // controller/PersonneController.php
4
5 namespace controller;
6
7 use entity\Personne;
8 use dao\MysqlDao;
9
10 class PersonneController {
11
12     private $dao;
13
14     public function __construct() {
15         $this->dao = new MysqlDao();
16     }
17
18     public function addPersonne() {
19         if ($_SERVER['REQUEST_METHOD'] == 'POST') {
20             $nom = filter_input(INPUT_POST, 'nom', FILTER_VALIDATE_REGEXP,
21                               ['options' => ['regexp' => '/^[A-Za-z- ]{1,20}$/']]);
22             $prenom = filter_input(INPUT_POST, 'prenom', FILTER_VALIDATE_REGEXP,
23                                   ['options' => ['regexp' => '/^[A-Za-z0-9]{4,10}$/']]);
24         }
25     }
26 }
```

```
24         if ($nom && $prenom) {
25             $personne = new Personne($nom, $prenom);
26             $this->dao->addPersonne($personne);
27             $this->getAllPersonnes();
28         } else {
29             $msg = 'Champs incorrects';
30             include VIEW . '/formulaire.php';
31         }
32     } else {
33         include VIEW . '/formulaire.php';
34     }
35 }
36
37 public function getAllPersonnes() {
38     $result = $this->dao->getAllPersonnes();
39     include VIEW . '/display_personne.php';
40 }
41
42 }
```

[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](#)