



# Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ [Petite application version 2](#) ▶ [Test de l'ajout d'une personne](#)

## Test de l'ajout d'une personne

Nous allons à présent tester la fonctionnalité d'ajout d'une personne. Cette fonctionnalité mettra en jeu plusieurs composants de notre application. Certains de ces composants ont déjà été testés, comme [la dao](#) et le routing, d'autres non, comme le controller `PersonneController` et [le front controller](#). Ce test fonctionnel va donc nous permettre de vérifier que les composants non testés pour l'instant fonctionnent correctement.

Notre controller étant très lié à la couche présentation et au serveur web, nous allons utiliser Selenium RC pour effectuer des tests à partir des IHM. Selenium va nous permettre de décrire les sollicitations effectuées sur une IHM (une vue). Nous pourrions ainsi décrire, dans un cas de test, les valeurs à injecter dans les champs de formulaire, ou encore des actions à effectuer sur les boutons et autres composants graphiques. Selenium permettra, en fait, de simuler les actions d'un utilisateur, et de vérifier que ces actions amènent au bon résultat.

Pour effectuer des tests via un navigateur internet nous allons avoir besoin de Selenium RC (Remote Control). Selenium RC (ou Selenium Server) est un outil d'automatisation de test qui va nous permettre d'effectuer des actions sur les IHM via JavaScript. Selenium est écrit en java et se présente sous la forme d'une archive java exécutable (par exemple `selenium-server_standalone-2.43.1.jar`). Selenium RC est téléchargeable sur le site de [Selenium](#). Dans le cadre de ce TP, vous pourrez placer le jar dans le répertoire `test` de l'application. Pour lancer le serveur, il suffit, dans un terminal, de se placer dans le répertoire de l'archive et d'exécuter la commande `java -jar selenium-server_standalone-2.43.1.jar` :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/petite_application$ java -jar test/selenium-server-stan
dalone-2.43.1.jar
10:45:56.349 INFO - Launching a standalone server
10:45:56.394 INFO - Java: Oracle Corporation 24.65-b04
10:45:56.394 INFO - OS: Linux 3.2.0-4-686-pae i386
10:45:56.412 INFO - v2.43.1, with Core v2.43.1. Built from revision 5163bce
10:45:56.523 INFO - Default driver org.openqa.selenium.ie.InternetExplorerDriver
registration is skipped: registration capabilities Capabilities [{platform=WIND
OWS, ensureCleanSession=true, browserName=internet explorer, version=}] does not
match with current platform: LINUX
10:45:56.567 INFO - RemoteWebDriver instances should connect to: http://127.0.0.
1:4444/wd/hub
10:45:56.568 INFO - Version Jetty/5.1.x
10:45:56.569 INFO - Started HttpContext[/selenium-server/driver,/selenium-server
/driver]
10:45:56.570 INFO - Started HttpContext[/selenium-server,/selenium-server]
10:45:56.570 INFO - Started HttpContext[/,/]
10:45:56.594 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@2f40f4
10:45:56.594 INFO - Started HttpContext[/wd,/wd]
10:45:56.599 INFO - Started SocketListener on 0.0.0.0:4444
10:45:56.599 INFO - Started org.openqa.jetty.jetty.Server@ca66b4
```

Nous allons créer une classe `PersonneControllerTest` dans le répertoire `test/controller` de notre application. Cette classe de test héritera de la classe `PHPUnit_Extensions_SeleniumTestCase`.

La classe `PHPUnit_Extensions_SeleniumTestCase` possède de nombreuses méthodes qui vont nous permettre d'effectuer des actions sur les IHM, parmi elles :

- `setBrowser($url)` : prend en argument le nom du navigateur de test
- `setBrowserUrl($url)` : prend en argument l'URL qui sera ouverte dans le navigateur de test
- `open($url)` : permet d'ouvrir une URL passée en argument
- `click($locator)` : permet d'effectuer un click sur un composant, prend en argument une chaîne locator
- `type($locator, $value)` : permet d'injecter une valeur dans un champ texte, prend en argument un locator et une valeur
- `submit($locator)` : permet de soumettre un formulaire, prend en argument le locator du formulaire

Après une action qui entraîne un chargement de page (clique sur un lien ou validation d'un formulaire par

exemple), il est nécessaire d'appeler la méthode `waitForPageToLoad()`.

Il existe de nombreuses autres méthodes décrites dans [le document de référence](#).

La chaîne locator permet la sélection d'éléments et est de la forme "typeLocator=valeur". Parmi les types de locator nous trouvons les locators :

- `css` : dont la valeur sera le sélecteur css correspondant à l'élément que l'on souhaite sélectionner (ex : "`css=#nom`")
- `id` : dont la valeur sera l'id du composant (ex : "`id=nom`")
- `name` : dont la valeur sera le name du composant
- `link` : dont la valeur sera le contenu texte de l'élément a

La classe `PHPUnit_Extensions_SeleniumTestCase` possède aussi des méthodes de vérification sur le modèle `assert*****()`. Par exemple :

- `assertTitle($title)` : vérifie le titre de la page
- `assertElementContainsText($locator, $value)` : vérifie si l'élément correspondant au locator contient la valeur \$value
- `assertChecked($locator)` : vérifie si l'élément correspondant au locator est sélectionné
- `assertTable($locator, $value)` : vérifie si le tableau et la cellule correspondant au locator contient la valeur \$value

Pour tester la méthode `getAllPersonnes()` du controller nous allons créer une méthode de test dans laquelle nous ouvrirons l'URL '`http://petite_application/personnes`' dans la navigateur, puis nous vérifierons le contenu des cellules de la deuxième ligne de notre tableau (la première étant la ligne contenant les en-têtes). Pour accéder à ces cellules du tableau nous utiliserons un sélecteur du type `css=table.numCol.numLigne`, par exemple '`css=table.1.1`' pour la deuxième cellule de la deuxième ligne.

La méthode `setUp()` permettra de choisir le navigateur et son URL.

```

1  <?php
2
3  // test/controller/PersonneControllerTest.php
4
5  class PersonneControllerTest extends PHPUnit_Extensions_SeleniumTestCase {
6
7      protected function setUp() {
8          $this->setBrowser('firefox');
9          $this->setBrowserUrl('http://localhost:8080/');
10     }
11
12     public function testGetAllPersonnes() {
13         $this->open('http://localhost:8080/personnes');
14         $this->waitForPageToLoad();
15         $this->assertTable('css=table.1.0', '1');
16         $this->assertTable('css=table.1.1', 'Sparrow');
17         $this->assertTable('css=table.1.2', 'Jack');
18     }
19
20 }
```

Pour tester l'ajout de la personne nous ouvrirons l'URL "`http://petite_application/personnes/ajouter`", puis nous "injecterons" de la donnée dans les champs nom et prénom du formulaire, puis nous soumettrons le formulaire. Nous vérifierons dans la page suivante si le champ désiré a bien été ajouté à la fin de notre tableau. Avant de lancer ce test il faut bien sûr avoir une base de données "propres" (vous pouvez pour cela importer le dump [personne.sql](#)).

```

1  <?php
2
3  // test/controller/PersonneControllerTest.php
4
5  class PersonneControllerTest extends PHPUnit_Extensions_SeleniumTestCase {
6
7      protected function setUp() {
8          $this->setBrowser('firefox');
9          $this->setBrowserUrl('http://localhost:8080/');
```

```
10     }
11
12     public function testGetAllPersonnes () {
13         $this->open('http://localhost:8080/personnes');
14         $this->waitForPageToLoad();
15         $this->assertTable('css=table.1.0', '1');
16         $this->assertTable('css=table.1.1', 'Sparrow');
17         $this->assertTable('css=table.1.2', 'Jack');
18     }
19
20     public function testAddPersonne () {
21         $nom = 'Stark';
22         $prenom = 'Tony';
23         $this->open('http://localhost:8080/personnes/ajouter');
24         $this->waitForPageToLoad();
25         $this->type('id=nom', $nom);
26         $this->type('id=prenom', $prenom);
27         $this->submit('css=form');
28         $this->waitForPageToLoad();
29         $this->assertTable('css=table.5.0', '5');
30         $this->assertTable('css=table.5.1', $nom);
31         $this->assertTable('css=table.5.2', $prenom);
32     }
33
34 }
```

[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](#)