



# Programmation orientée objet en PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [POO PHP](#) ► [Petite application version 2](#) ► [Test de la DAO](#)

## Test de la DAO

Nous allons effectuer un [premier test](#) d'intégration pour vérifier le bon fonctionnement de notre DAO. En testant [la classe MysqlDao](#) nous testerons aussi sa dépendance, c'est-à-dire la classe PDO qui fournit la connexion à la base donnée. On ne peut donc pas parler ici de test unitaire, mais plutôt de test d'intégration. Pour un premier exemple, ce niveau de granularité nous suffit.

Pour effectuer le test de [la classe MysqlDao](#) nous allons donc créer une classe `test\dao\MysqlDaoTest` héritant de `PHPUnit_Extensions_Database_TestCase`.

Nous placerons dans le répertoire `test/dataset` les deux fichiers qui nous serviront de dataset. À savoir `personne_base.yml` :

```
1 # test/dataset/personne_base.yml
2
3 personne:
4   -
5     id: 1
6     nom: "Sparrow"
7     prenom: "Jack"
8   -
9     id: 2
10    nom: "Wayne"
11    prenom: "Bruce"
```

et `add_personne.yml` :

```
1 # test/dataset/add_personne.yml
2
3 personne:
4   -
5     id: 1
6     nom: "Sparrow"
7     prenom: "Jack"
8   -
9     id: 2
10    nom: "Wayne"
11    prenom: "Bruce"
12   -
13     id: 3
14     nom: "Kent"
15     prenom: "Clark"
```

L'arborescence de notre répertoire de test sera donc la suivante :

```
▼ test
  ▼ dao
    MysqlDaoTest.php
  ▼ dataset
    add_personne.yml
    personne_base.yml
    phpunit.xml
```

La classe `MysqlDaoTest` sera quasiment identique à celle du module précédent, à ceci près que cette fois la méthode `addPersonne()` de `MysqlDao` prendra en argument un objet de type `Personne` et non deux chaînes de caractères. De plus nous irons cette fois chercher la configuration de connexion dans le fichier `conf.ini`. Les

"include" et les namespaces utilisés seront aussi différents. Notez que nous incluons le fichier setup.php car il contient la définition des constantes utilisées dans l'application.

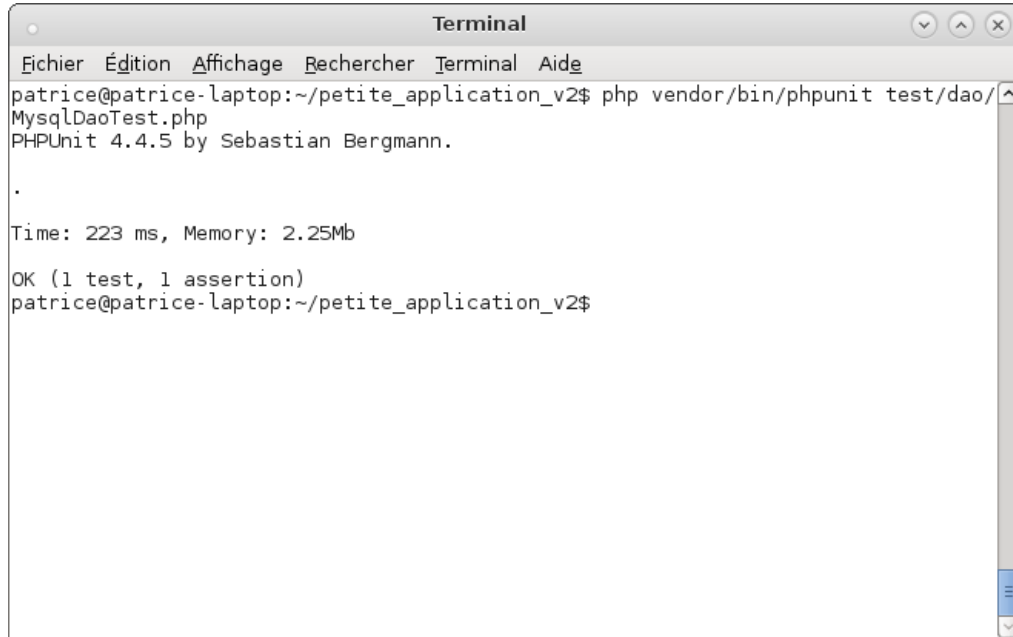
```

1  <?php
2
3  // test/dao/MysqlDaoTest.php
4
5  include_once './setup.php';
6  include_once './entity/Personne.php';
7  include_once './dao/MysqlDao.php';
8
9  use entity\Personne;
10 use dao\MysqlDao;
11
12 class MysqlDaoTest extends PHPUnit_Extensions_Database_TestCase {
13
14     protected $connection;
15     protected $dao;
16
17     protected function getConnection() {
18         if ($this->connection == null) {
19             $conf = parse_ini_file(INI_FILE, true);
20             $settings = $conf['settings']['env'];
21             $host = $conf[$settings]['host'];
22             $db_name = $conf[$settings]['db_name'];
23             $this->user = $conf[$settings]['user'];
24             $this->password = $conf[$settings]['password'];
25             $this->datasource = "mysql:host=$host;dbname=$db_name";
26             $conn = new PDO($this->datasource, $this->user, $this->password
27                 , [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
28             $this->connection = $this->createDefaultDBConnection($conn);
29         }
30         return $this->connection;
31     }
32
33     protected function getDataSet() {
34         return new PHPUnit_Extensions_Database_DataSet_YamlDataSet(
35             './test/dataset/personne_base.yml');
36     }
37
38     protected function setUp() {
39         $conn = $this->getConnection();
40         // désactivation des contraintes de clés étrangères pour permettre
41         // le chargement des données via le dataset
42         $conn->getConnection()->query('set foreign_key_checks=0');
43         parent::setUp();
44         // activation des contraintes de clés étrangères
45         $conn->getConnection()->query('set foreign_key_checks=1');
46         $this->dao = new MysqlDao();
47     }
48
49     public function testAddPersonne() {
50         $pers = new Personne("Kent", "Clark");
51         $this->dao->addPersonne($pers);
52         // création d'un objet dataset à partir de la connexion
53         $actualDataset = new PHPUnit_Extensions_Database_DataSet_QueryDataSet(
54             $this->getConnection());
55         // ajout à ce dataset des enregistrements de la table personne
56         // de notre base de données de test
57         $actualDataset->addTable('personne');
58         // récupération d'un dataset à partir de notre fichier
59         $expectedDataset = new PHPUnit_Extensions_Database_DataSet_YamlDataSet(
60             './test/dataset/add_personne.yml');
61         // comparaison des deux datasets
62         $this->assertDataSetsEqual($expectedDataset, $actualDataset);
63     }
64

```

```
65 }
}
```

Nous effectuons le test :



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/petite_application_v2$ php vendor/bin/phpunit test/dao/
MysqlDaoTest.php
PHPUnit 4.4.5 by Sebastian Bergmann.

.

Time: 223 ms, Memory: 2.25Mb

OK (1 test, 1 assertion)
patrice@patrice-laptop:~/petite_application_v2$
```

Il nous faudra ainsi tester toutes les méthodes de [la classe MysqlDao](#).

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](#)