

Secteur Tertiaire Informatique

Filière « Etude et développement »

Séquence « Développer des pages Web »

**Les Tests sur les WebApp en PHP**

Apprentissage

Mise en pratique  
avec PHPUnit

Évaluation



## TABLE DES MATIÈRES

<b>1.</b>	<b>RÉALISATION DE TESTS EN UTILISANT PHPUNIT.....</b>	<b>3</b>
1.1	INSTALLATION DE PHPUNIT .....	3
1.2	RÉALISATION DES TESTS.....	3

# 1. RÉALISATION DE TESTS EN UTILISANT PHPUnit.

## 1.1 INSTALLATION DE PHPUnit

Pour consulter la documentation de PHPUnit :

<https://phpunit.readthedocs.io/fr/latest/installation.html#>

Installation de **PHPUnit** :

Lancez l'invite de commande et placez-vous sur le dossier de votre projet

Par exemple :

```
C:\wamp64\www\TP_DWWM\Back\Abicom_MVC
```

Toujours en ligne de commande, lancez la commande suivante :

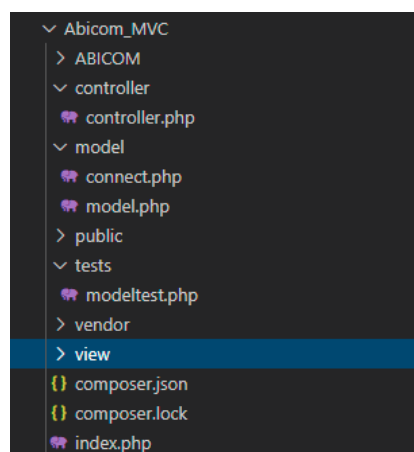
```
➤ composer require --dev phpunit/phpunit
```

Pour vérifier que PHPUnit est bien installé, avec votre logiciel de ligne de commande, placez-vous à la racine de votre projet, et lancez la commande suivante :

```
➤ vendor/bin/phpunit --help
```

## 1.2 RÉALISATION DES TESTS

Pour mémoire, voici l'arborescence du projet **AbiCom**, en MVC



Le fichier **model.php** contient toutes les méthodes de récupération des données de ce projet. C'est ce fichier que nous allons tester.

Pour ce faire, à la racine du répertoire, nous allons créer un dossier **tests**

ABICOM	04/02/2021 18:01	Dossier de fichiers	
controller	04/02/2021 18:01	Dossier de fichiers	
model	04/02/2021 18:01	Dossier de fichiers	
public	04/02/2021 18:01	Dossier de fichiers	
tests	15/02/2021 16:35	Dossier de fichiers	
vendor	15/02/2021 14:00	Dossier de fichiers	
view	04/02/2021 18:01	Dossier de fichiers	
composer.json	15/02/2021 14:09	Fichier JSON	1 Ko
composer.lock	15/02/2021 14:03	Fichier LOCK	74 Ko
index	28/01/2021 08:29	Fichier PHP	2 Ko

Dans ce répertoire, nous allons créer le fichier **modeltest.php**

Pour mémoire, le fichier **model.php** contient la fonction **get\_client\_by\_id**, qui récupère les données d'un client à partir de son **id**:

```
// Recherche d'un client
function get_client_by_id($id)
{
    $connexion = connect_db();
    $sql = $connexion->prepare("SELECT * from clients where idClient = :id");
    $sql->bindValue(":id", intval($id), PDO::PARAM_INT);
    $sql->execute();

    return $sql->fetch(PDO::FETCH_BOTH);
}
```

Nous voulons tester que cette fonction retourne bien un enregistrement si l'id existe, et retourne une erreur si l'id n'existe pas

Et nous allons donc créer nos **méthodes de test** :

```
<?php

namespace Tests\AppBundle\Entity;

include_once('model/model.php');

use PHPUnit\Framework\TestCase;

class ModelTest extends TestCase
```

```
{
    public function testgetClientbyid1()
    {
        $this->assertNotEmpty(get_client_by_id('2'));
    }
    public function testgetClientbyid2()
    {
        $this->assertNotEmpty(get_client_by_id('5'));
    }
}

?>
```

Puis nous testons le résultat de ce test. Dans votre logiciel de ligne de commande, à la racine de votre projet, lancez la commande suivante :

➤ `vendor/bin/phpunit tests/modeltest.php`

Le résultat est le suivant :

```
PS C:\wamp64\www\TP_DWMM\Back\Abicom_MVC> vendor/bin/phpunit tests/modeltest.php
Class 'modeltest' could not be found in 'C:\wamp64\www\TP_DWMM\Back\Abicom_MVC\tests\modeltest.php'.
PS C:\wamp64\www\TP_DWMM\Back\Abicom_MVC> vendor/bin/phpunit tests/modeltest.php
PHPUnit 9.5.2 by Sebastian Bergmann and contributors.

.F                                                                    2 / 2 (100%)

Time: 00:27.361, Memory: 4.00 MB

There was 1 failure:

1) Tests\AppBundle\Entity\ModelTest::testgetClientbyid2
Failed asserting that a boolean is not empty.

C:\wamp64\www\TP_DWMM\Back\Abicom_MVC\tests\modeltest.php:17

FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
PS C:\wamp64\www\TP_DWMM\Back\Abicom_MVC> █
```

Le test indique qu'il y a un échec, pour la méthode **testgetClientbyid2**, en effet, aucun client n'a l'id n°5

## Ajouter un nouveau client

ID Client	Raison sociale	Code postal client
2	AgriCorp	32 Rue des allouettes
3	AgriMoon	12 Rue des lampions
4	AgriGel	10 Rue des lapins
10	Silice	79 Rue des Maturins

PHPUnit propose toute une batterie **d'assertions** pour pouvoir mener à bien des **tests unitaires**.

Exemples d'assertion :

```
// Check 1 === 1 is true
$this->assertTrue(1 === 1);

// Check 1 === 2 is false
$this->assertFalse(1 === 2);

// Check 'Hello' equals 'Hello'
$this->assertEquals('Hello', 'Hello');

// Check array has key 'language'
$this->assertArrayHasKey('language', array('language' => 'php', 'size' => '1024'));

// Check array contains value 'php'
$this->assertContains('php', array('php', 'ruby', 'c++', 'JavaScript'));
```

La liste complète des assertions se trouve :

<https://phpunit.readthedocs.io/fr/latest/assertions.html?highlight=assertion>

Afin de vous entrainer un peu, essayer d'écrire vous-même des **Tests** en utilisant les différentes **assertions** présentées dans la documentation officielle de **PHPUnit**.

Organisation des tests :

**PHPUnit** offre une solution particulièrement **flexible** pour organiser vos tests, vous pouvez les exécuter indépendamment les uns des autres, les grouper en suites, ou rassembler les suites elles-mêmes afin d'exécuter le tout en une seule fois.

## **CREDITS**

### **ŒUVRE COLLECTIVE DE l'AFPA**

**Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services**

### **Équipe de conception (IF, formateur, mediatiseur)**

Formateur : Sacha RESTOUEIX

Sources WikiBooks

**Date de mise à jour : 15/02/2021**

## **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »