



TEST UNITAIRE AVEC PHPUNIT

Qu'est-ce que les tests unitaires ?

- ▶ Le test unitaire est un moyen de vérifier qu'un extrait de code fonctionne correctement. Il consiste à isoler une partie du code et à vérifier qu'il fonctionne parfaitement. Il s'agit de petits tests qui valident l'attitude d'un objet et la logique du code.
- ▶ Les tests unitaires sont généralement effectués pendant la phase de développement des applications mobiles ou logicielles. Ces tests sont normalement effectués par les développeurs, bien qu'à toutes fins pratiques, ils puissent également être effectués par les responsables en assurance la qualité de l'application.

Le contexte où j'utilise les tests unitaires

- Il est important de s'assurer que mes codes écrits pour les projets à l'école, voire ceux réalisés en group, fonctionnent correctement car toute erreur peut entraîner l'échec de l'ensemble des efforts et du temps passé dessus. En effet, en les exécutant, les erreurs peuvent être détectées beaucoup plus tôt – des erreurs qui, sans tests unitaires, ne seraient détectées qu'à des stades plus avancés tels que le système, l'intégration des codes de l'ensemble de l'équipe.

Comment je l'installe sur mon Windows

- Les exigences avant l'installation :



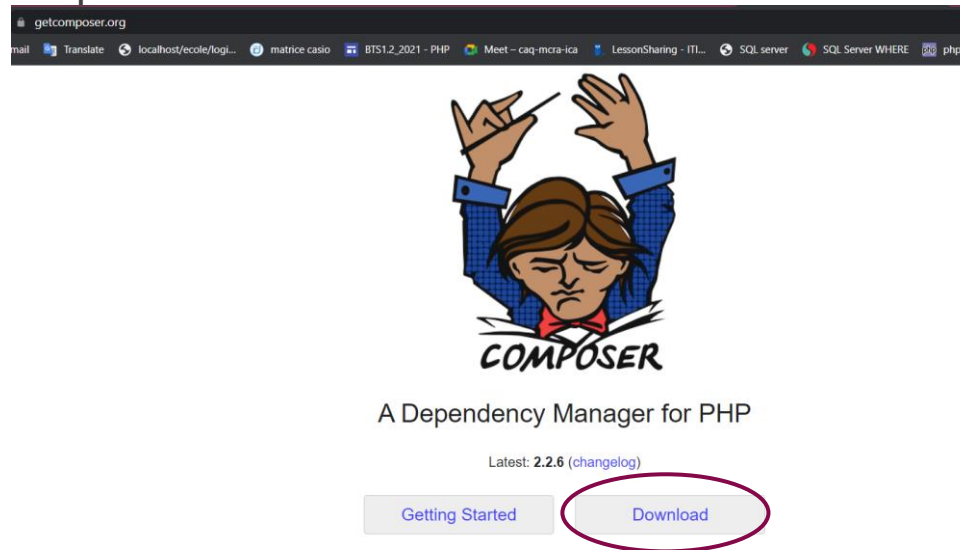
(au minimum)



Configurez PATH ENVIRONNEMENT (Propriétés système | Avancé | Variables d'environnement | Variables système | Chemin): Ajoutez le chemin de répertoire PHP aux "Variables d'environnement"

Comment je l'installe sur mon Windows

► Etape 1: installer le composer



► Etape 2: installer le PHPUnit

Aller sur l'invite des commandes et taper : **composer require --dev phpunit/phpunit [la version]**

```
C:\Users\Lise>composer require --dev phpunit/phpunit ^9.5
```

Comment je l'installe sur mon Windows

- Créer un dossier « testphpunit » et puis configurer son composer:

Commande dans Git Bash ->

```
Lise@LisePC MINGW64 /c/xampp/htdocs/testphpunit  
$ composer init
```

Continuer le processus ->

```
This command will guide you through creating your composer.json config.  
Package name (<vendor>/<name>) [Lise/testphpunit]:  
Description []:  
Author [Lise <lisetherese@yahoo.com>, n to skip]:  
Minimum Stability []:  
Package Type (e.g. library, project, metapackage, composer-plugin) []:  
License []:  
  
Define your dependencies.  
  
would you like to define your dependencies (require) interactively [yes]? no  
would you like to define your dev dependencies (require-dev) interactively [yes]  
? yes  
Search for a package: phpunit  
  
Found 15 packages matching phpunit  
  
[0] phpunit/phpunit  
[1] phpunit/php-timer  
[2] phpunit/php-text-template  
[3] phpunit/php-file-iterator  
[4] phpunit/php-code-coverage  
[5] phpunit/phpunit-mock-objects Abandoned. No replacement was suggested.  
[6] symfony/phpunit-bridge  
[7] phpunit/php-invoker  
[8] phpunit/php-token-stream Abandoned. No replacement was suggested.  
[9] johnkary/phpunit-speedtrap  
[10] jean85/pretty-package-versions  
[11] brianium/paratest  
[12] yoast/phpunit-polyfills  
[13] spatie/phpunit-snapshot-assertions  
[14] phpunit/phpunit-selenium  
  
Enter package # to add, or the complete package name if it is not listed: 0  
Enter the version constraint to require (or leave blank to use the latest version):  
Using version ^9.5 for phpunit/phpunit  
Search for a package:  
Add PSR-4 autoloader mapping? Maps namespace "Lise\Testphpunit" to the entered relative path. [src/, n to skip]:
```

Comment je fais des tests unitaires

- Créer les dossier « src » et « tests » dans le dossier « testphpunit », le résultat sera comme suivant:

Répertoire vue dans Visual Studio->

```
▼ testphpunit
  > src
  > tests
  > vendor
  ≡ .phpunit.result.cache
  {} composer.json
  {} composer.lock
  📄 phpunit.xml
```

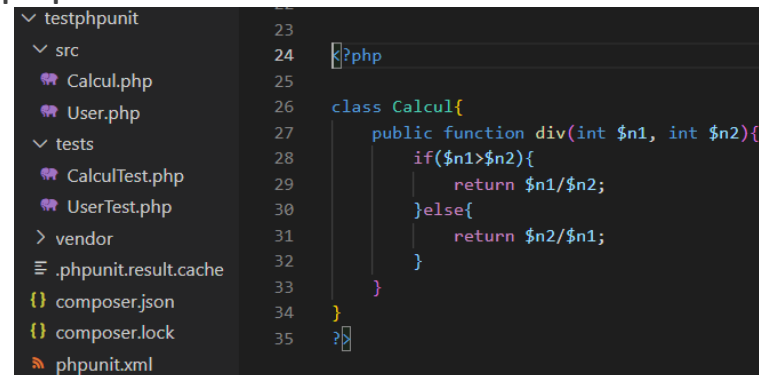
*Le fichier composer.json
s'apparaît comme ça ->*

```
{
  "name": "lise/testphpunit",
  "require-dev": {
    "phpunit/phpunit": "^9.5"
  },
  "autoload": {
    "psr-4": {
      "Lise\\Testphpunit\\": "src/"
    }
  },
  "authors": [
    {
      "name": "Lise",
      "email": "lisetherese@yahoo.com"
    }
  ],
  "require": {}
}
```

Comment je fais des tests unitaires

- Créer des fichiers php contiennent des classes avec ses fonctions pour tester dans le dossier « src »

Code ->

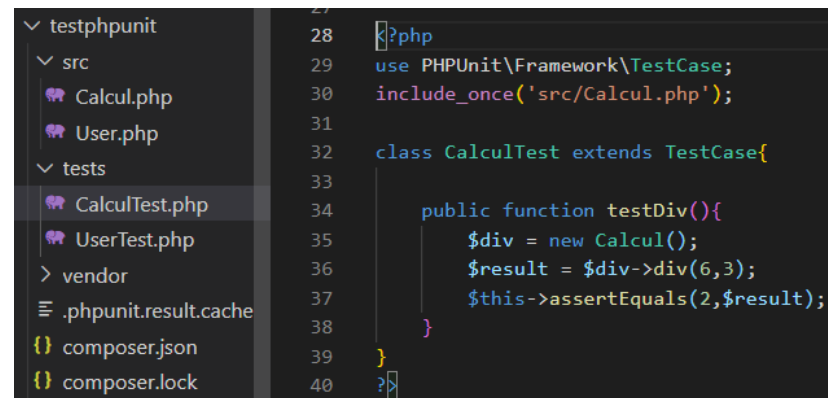


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'testphpunit', 'src', and 'tests'. The 'src' folder contains 'Calcul.php' and 'User.php'. The 'tests' folder contains 'CalculTest.php' and 'UserTest.php'. The 'testphpunit' folder also contains '.phpunit.result.cache', 'composer.json', 'composer.lock', and 'phpunit.xml'. The code editor shows the content of 'Calcul.php' with line numbers 23 to 35. The code is as follows:

```
23  
24 <?php  
25  
26 class Calcul{  
27     public function div(int $n1, int $n2){  
28         if($n1>$n2){  
29             return $n1/$n2;  
30         }else{  
31             return $n2/$n1;  
32         }  
33     }  
34 }  
35
```

- Créer des fichiers php contiennent des classes de test dans le dossier “tests”

Code ->



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the same project structure as the previous screenshot. The 'tests' folder is selected, and 'CalculTest.php' is highlighted. The code editor shows the content of 'CalculTest.php' with line numbers 28 to 40. The code is as follows:

```
28 <?php  
29 use PHPUnit\Framework\TestCase;  
30 include_once('src/Calcul.php');  
31  
32 class CalculTest extends TestCase{  
33  
34     public function testDiv(){  
35         $div = new Calcul();  
36         $result = $div->div(6,3);  
37         $this->assertEquals(2,$result);  
38     }  
39 }  
40
```


Comment je fais des tests unitaires

- ▶ Ouvrir le terminal dans le dossier « testphpunit » puis taper la commande « ./vendor/bin/phpunit » pour activer le phpunit qui se localise dans ./vendor/bin
- ▶ Si le résultat est bon, la réponse sera en vert OK, sinon en rouge FAILURE

Résultat ->

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\xampp\htdocs\testphpunit> ./vendor/bin/phpunit

PHPUnit 9.5.14 by Sebastian Bergmann and contributors.

.                                                                    1 / 1 (100%)

Time: 00:00.012, Memory: 6.00 MB

OK (1 test, 1 assertion)
PS C:\xampp\htdocs\testphpunit> 
```