

Tests fonctionnels automatisés

L'automatisation des tests fonctionnels consiste à créer des scénarii de tests qui pourront ensuite être reproduits à la demande (exécutés) au cours du développement.

- La réalisation d'un scénario permet de mieux identifier le besoin exprimé
- La mise en place du test relatif à une fonctionnalité permet de tester cette fonctionnalité (en cours et en fin d'implémentation)
- L'exécution d'une suite de tests permet de vérifier la non-régression d'un projet suite à une modification ou l'introduction d'une nouvelle fonctionnalité.

-- Mise en place des outils pour PHP

-- Composer

Composer est un gestionnaire de paquets compatible GIT permettant d'installer ou de mettre à jour les librairies incluses dans un projet à partir d'un fichier de configuration **composer.json**, déclarant les dépendances du projet.

Installation

Sous Windows :

- télécharger et installer [Composer-Setup.exe](#)
- Ajouter le dossier d'installation de composer dans la variable PATH de windows pour pouvoir exécuter composer directement en ligne de commande.

Vérifier l'installation :

Dans un terminal : Frapper **composer -v** puis Entrée ←



```
C:\Users\jc>composer -v  
  
Composer version 1.0-dev (c41079192f38f0fc446b17baa8f628dcb3b61e7d) 2015-09-28 09:38:16
```

-- PHPUnit et WebDriver

PHPUnit va permettre de réaliser des tests unitaires (différents des tests fonctionnels).

Pour la partie fonctionnelle, nous utiliserons **Selenium Server** + **Facebook WebDriver**, pour émuler les interactions utilisateur dans un navigateur.

```
{  
  "require-dev": {
```

```
"facebook/webdriver": "dev-master",  
"phpunit/phpunit": "~4.8"  
}  
}
```

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/slam5/testsfonctionnels/automatisees?rev=1443462095>

Last update: **2019/08/31 14:37**

