



ATOUM'S DOCUMENTATION

[Écrire ses tests \(chapitre2.html\)](#) >

1. Introduction 🔗 ☰

1.1. Qu'est-ce que atoum ? 🔗 ☰

atoum est un framework de test unitaire, tout comme PHPUnit ou SimpleTest, mais il présente quelques avantages par rapport à ces derniers :

- il est moderne et utilise les innovations des dernières versions de PHP (≥ 5.3) ;
- il est simple et facile à maîtriser ;
- il est intuitif, sa syntaxe se veut la plus proche du langage naturel anglais.

1.2. Téléchargement et installation 🔗 ☰

Pour le moment, atoum n'a pas de numéro de version. Il est néanmoins d'ores et déjà stable et utilisable. Si vous souhaitez l'utiliser, il vous suffit de télécharger la dernière version. Malgré les évolutions constantes d'atoum, la rétrocompatibilité est une des priorités de ses développeurs.

Vous pouvez installer atoum de plusieurs manières :

- en téléchargeant l'archive PHAR ;
- à l'aide de composer ;
- en utilisant un script d'installation ;
- en clonant le dépôt github ;
- en utilisant un plug-in symfony 1 ;
- en utilisant un bundle Symfony 2.
- en utilisant un composant Zend Framework 2.

1.2.1. Archive PHAR 🔗 ☰

Une archive PHAR (PHp ARchive) est créée automatiquement à chaque modification d'atoum.

PHAR est un format d'archive d'application PHP, disponible depuis PHP 5.3.

1.2.1.1. Installation 🔗 ☰

Vous pouvez télécharger la dernière version stable d'atoum directement depuis le site officiel : <http://downloads.atoum.org/nightly/mageekguy.atoum.phar>
(<http://downloads.atoum.org/nightly/mageekguy.atoum.phar>)

1.2.1.2. Mise à jour 🔗 ☰

La mise à jour de l'archive est très simple. Il vous suffit de lancer la commande suivante :

```
$ php -d phar.readonly=0 mageekguy.atoum.phar --update
```

La mise à jour d'atoum nécessite la modification de l'archive PHAR. Or par défaut, la configuration de PHP ne l'autorise pas. Voilà pourquoi il faut utiliser la directive - d phar.readonly=0 .

Si une version plus récente existe, elle sera alors téléchargée automatiquement et installée au sein de l'archive :

```
$ php -d phar.readonly=0 mageekguy.atoum.phar --update
Checking if a new version is available... Done !
Update to version 'nightly-1568-201210311708'... Done !
Enable version 'nightly-1568-201210311708'... Done !
Atoum was updated to version 'nightly-1568-201210311708' successfully !
```

S'il n'existe pas de version plus récente, atoum s'arrêtera immédiatement :

```
$ php -d phar.readonly=0 mageekguy.atoum.phar --update
Checking if a new version is available... Done !
There is no new version available !
```

Atoum ne demande aucune confirmation de la part de l'utilisateur pour réaliser la mise à jour, car il est très facile de revenir à une version précédente.

1.2.1.3. Lister les versions contenues dans l'archive 🔗 ☰

Pour afficher les versions contenues dans l'archive au fur et à mesure des mises à jour, il faut faire appel à l'argument `--list-available-versions`, ou `-lav` en version abrégée :

```
$ php mageekguy.atoum.phar -lav
nightly-941-201201011548
* nightly-1568-201210311708
```

La liste des versions présentes dans l'archive est alors affichée, la version actuellement active étant précédée de `*` .

1.2.1.4. Changer la version courante 🔗 ☰

Pour activer une autre version, il suffit d'utiliser l'argument `--enable-version`, ou `-ev` en version abrégée, suivi du nom de la version à utiliser :

```
$ php -d phar.readonly=0 mageekguy.atoum.phar -ev DEVELOPMENT
```

La modification de la version courante nécessite la modification de l'archive PHAR. Or par défaut, la configuration de php ne l'autorise pas. Voilà pourquoi il faut utiliser la directive `-d phar.readonly=0` .

1.2.1.5. Suppression d'anciennes versions 🔗 ☰

Au cours du temps, l'archive peut contenir plusieurs versions d'atoum qui ne sont plus utilisées.

Pour les supprimer, il suffit d'utiliser l'argument `--delete-version`, ou `-dv` dans sa version abrégée, suivi du nom de la version à supprimer :

```
$ php -d phar.readonly=0 mageekguy.atoum.phar -dv nightly-941-2012010115
```

La version est alors supprimée.

Il n'est pas possible de supprimer la version active.

*La suppression d'une version nécessite la modification de l'archive PHAR.
Or par défaut, la configuration de PHP ne l'autorise pas. Voilà pourquoi il
faut utiliser la directive `-d phar.readonly=0`.*

1.2.2. Composer

Composer (<http://getcomposer.org>) est un outil de gestion de dépendance en PHP.

Commencez par installer composer :

```
$ curl -s https://getcomposer.org/installer | php
```

Créez ensuite un fichier `composer.json` contenant le JSON (JavaScript Object Notation) suivant :

```
1. {  
2.     "require" : {  
3.         "atoum/atoum" : "dev-master"  
4.     }  
5. }
```

Enfin, exécutez la commande suivante :

```
$ php composer.phar install
```

1.2.3. Script d'installation

Vous pouvez installer atoum grâce à un script (<https://github.com/atoum/atoum-installer>) dédié à cette tâche :

```
$ curl https://raw.githubusercontent.com/atoum/atoum-installer/master/installer | p
$ php mageekguy.atoum.phar -v
atoum version nightly-xxxx-yyyymmddhhmm by Frédéric Hardy (phar:///path/
```

Vous pourrez facilement installer atoum localement (dans un projet, voir exemple précédent) ou globalement sur le système :

```
$ curl https://raw.githubusercontent.com/atoum/atoum-installer/master/installer | s
$ which atoum
/usr/local/bin/atoum
```

Des options sont disponibles pour personnaliser l'installation d'atoum : reportez-vous à la [documentation](https://github.com/atoum/atoum-installer/blob/master/README.md) (en) (<https://github.com/atoum/atoum-installer/blob/master/README.md>) de l'installeur pour obtenir plus de détails.

1.2.4. Github

Si vous souhaitez utiliser atoum directement depuis ses sources, vous pouvez cloner ou « forker » le dépôt github : [git://github.com/atoum/atoum](https://github.com/atoum/atoum) (git://github.com/atoum/atoum)

1.2.5. Plugin symfony 1

Si vous souhaitez utiliser atoum au sein d'un projet symfony 1, un plug-in existe et est disponible à l'adresse suivante : <https://github.com/atoum/sfAtoumPlugin> (<https://github.com/atoum/sfAtoumPlugin>).

Toutes les instructions pour son installation et son utilisation se trouvent dans le cookbook ([chapitre4.html#Utilisation-avec-symfony-1.4](#)) ainsi que sur son site.

1.2.6. Bundle Symfony 2

Si vous souhaitez utiliser atoum au sein d'un projet Symfony 2, un bundle existe et est disponible à l'adresse suivante : <https://github.com/atoum/AtoumBundle> (<https://github.com/atoum/AtoumBundle>).

Toutes les instructions pour son installation et son utilisation sont disponibles sur cette page.

1.2.7. Composant Zend Framework 2

Si vous souhaitez utiliser atoum au sein d'un projet Zend Framework 2, un composant existe et est disponible à l'adresse suivante : <https://github.com/blanchonvincent/zend-framework-test-atoum> (<https://github.com/blanchonvincent/zend-framework-test-atoum>).

Toutes les instructions pour son installation et son utilisation sont disponibles sur cette page.

1.3. La philosophie d'atoum

1.3.1. Exemple simple

Vous devez écrire une classe de test pour chaque classe à tester.

Imaginez que vous vouliez tester la traditionnelle classe `HelloWorld`, alors vous devez créer la classe de test `test\units\HelloWorld`.

atoum utilise les espaces de noms. Par exemple, pour tester la classe `\Vendor\Project\HelloWorld`, vous devez créer la classe `\Vendor\Project\tests\units\HelloWorld`.

Voici le code de la classe `HelloWorld` que nous allons tester.

```
1. <?php
2. # src/Vendor/Project/HelloWorld.php
3.
4. namespace Vendor\Project;
5.
6. class HelloWorld
7. {
8.     public function getHiAtoum ()
9.     {
10.         return 'Hi atoum !';
11.     }
12. }
```

Maintenant, voici le code de la classe de test que nous pourrions écrire.

```

1. <?php
2. # src/Vendor/Project/tests/units/HelloWorld.php
3.
4. // La classe de test à son propre namespace :
5. // Le namespace de la classe à tester + "tests\units"
6. namespace Vendor\Project\tests\units;
7.
8. // Vous devez inclure la classe à tester
9. require_once __DIR__ . '/../../HelloWorld.php';
10.
11. use \atoum;
12.
13. /*
14.  * Classe de test pour \HelloWorld
15.
16.  * Remarquez qu'elle porte le même nom que la classe à tester
17.  * et qu'elle dérive de la classe atoum
18.  */
19. class HelloWorld extends atoum
20. {
21.     /*
22.      * Cette méthode est dédiée à la méthode getHiAtoum()
23.      */
24.     public function testGetHiAtoum ()
25.     {
26.         // création d'une nouvelle instance de la classe à tester
27.         $helloToTest = new \Vendor\Project\HelloWorld();
28.
29.         $this
30.             // nous testons que la méthode getHiAtoum retourne bien
31.             // une chaîne de caractère...
32.             ->string($helloToTest->getHiAtoum())
33.             // ... et que la chaîne est bien celle attendue,
34.             // c'est-à-dire 'Hi atoum !'
35.             ->isEqualTo('Hi atoum !')
36.         ;
37.     }
38. }

```

Maintenant, lançons nos tests.

Vous devriez voir quelque chose comme ça :

```
$ php vendor/mageekguy.atoum.phar -f src/Project/Project/tests/units/HelloWorld.php
> PHP path: /usr/bin/php
> PHP version:
=> PHP 5.4.7 (cli) (built : Sep 13 2012 04:24:47)
=> Copyright (c) 1997-2012 The PHP Group
=> Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
=> with Xdebug v2.2.1, Copyright (c) 2002-2012, by Derick Rethans
> Vendor\Project\tests\units\HelloWorld...
[S _____][1/1]
=> Test duration : 0.02 second.
=> Memory usage : 0.00 Mb.
> Total test duration: 0.02 second.
> Total test memory usage: 0.00 Mb.
> Code coverage value: 100.00%
> Running duration: 0.34 second.
Success (1 test, 1/1 method, 2 assertions, 0 error, 0 exception) !
```

Nous venons de tester que la méthode `getHiAtoum` :

- retourne bien une chaîne de caractère ;
- que cette dernière est bien égale à "Hi atoum !".

Les tests sont passés, tout est au vert. Voilà, votre code est solide comme un roc grâce à atoum !

1.3.2. Principes de base 🔗 ☰

Lorsque vous voulez tester une valeur, vous devez :

- indiquer le type de cette valeur (entier, décimal, tableau, chaîne de caractères, etc.) ;
- indiquer les contraintes devant s'appliquer à cette valeur (égal à, nulle, contenant quelque chose, etc.).

1.4. Intégration d'atoum dans votre IDE 🔗 ☰

1.4.1. Sublime Text 2 🔗 ☰

Un plug-in pour SublimeText 2 (<https://github.com/toin0u/Sublime-atoum>) permet l'exécution des tests unitaires par atoum et la visualisation du résultat sans quitter l'éditeur.

Les informations nécessaires à son installation et à sa configuration sont disponibles sur le blog de son auteur (<http://sbin.dk/2012/05/19/atoum-sublime-text-2-plugin/>).

1.4.2. VIM

atoum est livré avec un plug-in facilitant son utilisation dans l'éditeur VIM.

Il permet d'exécuter les tests sans quitter VIM et d'obtenir le rapport correspondant dans une fenêtre de l'éditeur.

Il est alors possible de naviguer parmi les éventuelles erreurs, voire de se rendre à la ligne correspondant à une assertion ne passant pas à l'aide d'une simple combinaison de touches.

1.4.2.1. Installation du plug-in atoum pour VIM

Si vous n'utilisez pas atoum sous la forme d'une archive PHAR, vous trouverez le fichier correspondant au plug-in, nommé `atoum.vba`, dans le répertoire `ressources/vim`.

Dans le cas contraire, il faut demander à l'archive PHAR d'atoum d'extraire le fichier à l'aide de la commande suivante :

```
$ php mageekguy.atoum.phar --extractResourcesTo path/to/a/directory
```

Une fois l'extraction réalisée, le fichier `atoum.vba` correspondant au plug-in pour VIM sera dans le répertoire `path/to/a/directory/ressources/vim`.

Une fois en possession du fichier `atoum.vba`, il faut l'éditer à l'aide de VIM :

```
$ vim path/to/atoum.vba
```

Il n'y a plus ensuite qu'à demander à VIM l'installation du plug-in à l'aide de la commande :

```
:source %
```

1.4.2.2. Utilisation du plug-in atoum pour VIM

Pour utiliser le plug-in, atoum doit évidemment être installé et vous devez être en train d'éditer un fichier contenant une classe de tests unitaires basée sur atoum.

Une fois dans cette configuration, la commande suivante lancera l'exécution des tests :

```
:Atoum
```

Les tests sont alors exécutés, et une fois qu'ils sont terminés, un rapport basé sur le fichier de configuration de atoum qui se trouve dans le répertoire `ftplugin/php/atoum.vim` de votre répertoire `.vim` est généré dans une nouvelle fenêtre.

Évidemment, vous êtes libre de lier cette commande à la combinaison de touches de votre choix, en ajoutant par exemple la ligne suivante dans votre fichier `.vimrc` :

```
nnoremap *.php <F12> :Atoum<CR>
```

L'utilisation de la touche `F12` de votre clavier en mode normal appellera alors la commande `:Atoum`.

1.4.2.3. Gestion des fichiers de configuration de atoum

Vous pouvez indiquer un autre fichier de configuration pour atoum en ajoutant la ligne suivante à votre fichier `.vimrc` :

```
call atoum#defineConfiguration('/path/to/project/directory', '/path/to/a
```

La fonction `atoum#defineConfiguration` permet en effet de définir le fichier de configuration à utiliser en fonction du répertoire où se trouve le fichier de tests unitaires.

Elle accepte pour cela trois arguments :

- un chemin d'accès vers le répertoire contenant les tests unitaires ;
- un chemin d'accès vers le fichier de configuration de atoum devant être utilisé ;
- l'extension des fichiers de tests unitaires concernés.

Pour plus de détails sur l'utilisation du plug-in, une aide est disponible dans VIM à l'aide de la commande suivante :

```
:help atoum
```

1.4.3. Ouvrir automatiquement les tests en échec

atoum est capable d'ouvrir automatiquement les fichiers des tests en échec à la fin de l'exécution. Plusieurs éditeurs sont actuellement supportés :

- macvim (Mac OS X)
- gvim (Unix)
- PhpStorm (Mac OS X/Unix)
- gedit (Unix)

Pour utiliser cette fonctionnalité, vous devrez modifier le fichier de configuration ([chapitre3.html#Fichier-de-configuration](#)) de atoum :

1.4.3.1. macvim

```
1. <?php
2. use
3.     mageekguy\atoum,
4.     mageekguy\atoum\report\fields\runner\failures\execute\macos
5. ;
6.
7. $stdoutWriter = new atoum\writers\std\out();
8. $cliReport = new atoum\reports\realtime\cli();
9. $cliReport->addWriter($stdoutWriter);
10.
11. $cliReport->addField(new macos\macvim());
12.
13. $runner->addReport($cliReport);
```

1.4.3.2. gvim

```

1. <?php
2. use
3.     mageekguy\atoum,
4.     mageekguy\atoum\report\fields\runner\failures\execute\unix
5. ;
6.
7. $stdoutWriter = new atoum\writers\std\out();
8. $cliReport = new atoum\reports\realtime\cli();
9. $cliReport->addWriter($stdoutWriter);
10.
11. $cliReport->addField(new unix\gvim());
12.
13. $runner->addReport($cliReport);

```

1.4.3.3. PhpStorm

Si vous travaillez sous Mac OS X, utilisez la configuration suivante :

```

1. <?php
2. use
3.     mageekguy\atoum,
4.     mageekguy\atoum\report\fields\runner\failures\execute\macos
5. ;
6.
7. $stdoutWriter = new atoum\writers\std\out();
8. $cliReport = new atoum\reports\realtime\cli();
9. $cliReport->addWriter($stdoutWriter);
10.
11. $cliReport
12.     // Si PhpStorm est installé dans /Applications
13.     ->addField(new macos\phpstorm())
14.
15.     // Si vous avez installé PhpStorm
16.     // dans un dossier différent de /Applications
17.     // ->addField(
18.         //     new macos\phpstorm(
19.         //         '/path/to/PhpStorm.app/Contents/MacOS/webide'
20.         //     )
21.     // )
22. ;
23.
24. $runner->addReport($cliReport);

```

Dans un environnement Unix, utilisez la configuration suivante :

```

1. <?php
2. use
3.     mageekguy\atoum,
4.     mageekguy\atoum\report\fields\runner\failures\execute\unix
5. ;
6.
7. $stdoutWriter = new atoum\writers\std\out();
8. $cliReport = new atoum\reports\realtime\cli();
9. $cliReport->addWriter($stdoutWriter);
10.
11. $cliReport
12.     ->addField(
13.         new unix\phpstorm( '/chemin/vers/PhpStorm/bin/phpstorm.sh' )
14.     )
15. ;
16.
17. $runner->addReport($cliReport);

```

1.4.3.4. gedit 🔗 ☰

```

1. <?php
2. use
3.     mageekguy\atoum,
4.     mageekguy\atoum\report\fields\runner\failures\execute\unix
5. ;
6.
7. $stdoutWriter = new atoum\writers\std\out();
8. $cliReport = new atoum\reports\realtime\cli();
9. $cliReport->addWriter($stdoutWriter);
10.
11. $cliReport->addField(new unix\gedit());
12.
13. $runner->addReport($cliReport);

```

Écrire ses tests (chapitre2.html) ➤

1. Introduction (chapitre1.html#Introduction)

1.1. Qu'est-ce que atoum ? (chapitre1.html#Qu-est-ce-que-atoum)

1.2. Téléchargement et installation

(chapitre1.html#Telechargement-et-installation)

1.2.1. Archive PHAR (chapitre1.html#Archive-PHAR)

1.2.1.1. Installation (chapitre1.html#Installation)

1.2.1.2. Mise à jour (chapitre1.html#Mise-a-jour)

1.2.1.3. Lister les versions contenues dans l'archive (chapitre1.html#Lister-les-versions-contenues-dans-l-archive)

1.2.1.4. Changer la version courante (chapitre1.html#Changer-la-version-courante)

1.2.1.5. Suppression d'anciennes versions (chapitre1.html#Suppression-d-anciennes-

- versions)
- 1.2.2. Composer ([chapitre1.html#Composer](#))
- 1.2.3. Script d'installation ([chapitre1.html#Script-d-installation](#))
- 1.2.4. Github ([chapitre1.html#Github](#))
- 1.2.5. Plugin symfony 1 ([chapitre1.html#Plugin-symfony-1](#))
- 1.2.6. Bundle Symfony 2 ([chapitre1.html#Bundle-Symfony-2](#))
- 1.2.7. Composant Zend Framework 2 ([chapitre1.html#Composant-Zend-Framework-2](#))
- 1.3. La philosophie d'atoum ([chapitre1.html#La-philosophie-d-atoum](#))
 - 1.3.1. Exemple simple ([chapitre1.html#Exemple-simple](#))
 - 1.3.2. Principes de base ([chapitre1.html#Principes-de-base](#))
- 1.4. Intégration d'atoum dans votre IDE ([chapitre1.html#Integration-d-atoum-dans-votre-IDE](#))
 - 1.4.1. Sublime Text 2 ([chapitre1.html#Sublime-Text-2](#))
 - 1.4.2. VIM ([chapitre1.html#VIM](#))
 - 1.4.2.1. Installation du plug-in atoum pour VIM ([chapitre1.html#Installation-du-plug-in-atoum-pour-VIM](#))
 - 1.4.2.2. Utilisation du plug-in atoum pour VIM ([chapitre1.html#Utilisation-du-plug-in-atoum-pour-VIM](#))
 - 1.4.2.3. Gestion des fichiers de configuration de atoum ([chapitre1.html#Gestion-des-fichiers-de-configuration-de-atoum](#))
 - 1.4.3. Ouvrir automatiquement les tests en échec ([chapitre1.html#Ouvrir-automatiquement-les-tests-en-echec](#))
 - 1.4.3.1. macvim ([chapitre1.html#macvim](#))
 - 1.4.3.2. gvim ([chapitre1.html#gvim](#))
 - 1.4.3.3. PhpStorm ([chapitre1.html#PhpStorm](#))
 - 1.4.3.4. gedit ([chapitre1.html#gedit](#))

2. Écrire ses tests ([chapitre2.html#Ecrire-ses-tests](#))

- 2.1. Assertions ([chapitre2.html#Assertions](#))
 - 2.1.1. variable ([chapitre2.html#variable](#))
 - 2.1.1.1. isCallable ([chapitre2.html#variableIsCallable](#))
 - 2.1.1.2. isEqualTo ([chapitre2.html#variableIsEqualTo](#))
 - 2.1.1.3. isIdenticalTo ([chapitre2.html#variableIsIdenticalTo](#))
 - 2.1.1.4. isNotCallable ([chapitre2.html#variableIsNotCallable](#))
 - 2.1.1.5. isNotEqualTo ([chapitre2.html#variableIsNotEqualTo](#))
 - 2.1.1.6. isNotIdenticalTo ([chapitre2.html#variableIsNotIdenticalTo](#))
 - 2.1.1.7. isNull ([chapitre2.html#isNull](#))
 - 2.1.1.8. isNotNull ([chapitre2.html#isNotNull](#))
 - 2.1.2. boolean ([chapitre2.html#boolean](#))
 - 2.1.2.1. isEqualTo ([chapitre2.html#booleanIsEqualTo](#))
 - 2.1.2.2. isFalse ([chapitre2.html#isFalse](#))

- 2.1.2.3. `isIdenticalTo`
([chapitre2.html#booleanIsIdenticalTo](#))
- 2.1.2.4. `isNotEqualTo`
([chapitre2.html#booleanIsNotEqualTo](#))
- 2.1.2.5. `isNotIdenticalTo`
([chapitre2.html#booleanIsNotIdenticalTo](#))
- 2.1.2.6. `isTrue` ([chapitre2.html#isTrue](#))
- 2.1.3. `integer` ([chapitre2.html#integer](#))
 - 2.1.3.1. `isEqualTo`
([chapitre2.html#integerIsEqualTo](#))
 - 2.1.3.2. `isGreaterThan`
([chapitre2.html#integerIsGreaterThan](#))
 - 2.1.3.3. `isGreaterThanOrEqualTo`
([chapitre2.html#integerIsGreaterThanOrEqualTo](#))
 - 2.1.3.4. `isIdenticalTo`
([chapitre2.html#integerIsIdenticalTo](#))
 - 2.1.3.5. `isLessThan`
([chapitre2.html#integerIsLessThan](#))
 - 2.1.3.6. `isLessThanOrEqualTo`
([chapitre2.html#integerIsLessThanOrEqualTo](#))
 - 2.1.3.7. `isNotEqualTo`
([chapitre2.html#integerIsNotEqualTo](#))
 - 2.1.3.8. `isNotIdenticalTo`
([chapitre2.html#integerIsNotIdenticalTo](#))
 - 2.1.3.9. `isZero` ([chapitre2.html#integerIsZero](#))
- 2.1.4. `float` ([chapitre2.html#float](#))
 - 2.1.4.1. `isEqualTo` ([chapitre2.html#floatIsEqualTo](#))
 - 2.1.4.2. `isGreaterThan`
([chapitre2.html#floatIsGreaterThan](#))
 - 2.1.4.3. `isGreaterThanOrEqualTo`
([chapitre2.html#floatIsGreaterThanOrEqualTo](#))
 - 2.1.4.4. `isIdenticalTo`
([chapitre2.html#floatIsIdenticalTo](#))
 - 2.1.4.5. `isLessThan`
([chapitre2.html#floatIsLessThan](#))
 - 2.1.4.6. `isLessThanOrEqualTo`
([chapitre2.html#floatIsLessThanOrEqualTo](#))
 - 2.1.4.7. `isNearlyEqualTo`
([chapitre2.html#isNearlyEqualTo](#))
 - 2.1.4.8. `isNotEqualTo`
([chapitre2.html#floatIsNotEqualTo](#))
 - 2.1.4.9. `isNotIdenticalTo`
([chapitre2.html#floatIsNotIdenticalTo](#))
 - 2.1.4.10. `isZero` ([chapitre2.html#floatIsZero](#))
- 2.1.5. `sizeof` ([chapitre2.html#sizeof](#))
 - 2.1.5.1. `isEqualTo`
([chapitre2.html#sizeofIsEqualTo](#))
 - 2.1.5.2. `isGreaterThan`
([chapitre2.html#sizeofIsGreaterThan](#))
 - 2.1.5.3. `isGreaterThanOrEqualTo`
([chapitre2.html#sizeofIsGreaterThanOrEqualTo](#))
 - 2.1.5.4. `isIdenticalTo`
([chapitre2.html#sizeofIsIdenticalTo](#))
 - 2.1.5.5. `isLessThan`
([chapitre2.html#sizeofIsLessThan](#))
 - 2.1.5.6. `isLessThanOrEqualTo`
([chapitre2.html#sizeofIsLessThanOrEqualTo](#))
 - 2.1.5.7. `isNotEqualTo`
([chapitre2.html#sizeofIsNotEqualTo](#))
 - 2.1.5.8. `isNotIdenticalTo`
([chapitre2.html#sizeofIsNotIdenticalTo](#))

- 2.1.5.9. `isZero` ([chapitre2.html#sizeOfIsZero](#))
- 2.1.6. `object` ([chapitre2.html#object](#))
 - 2.1.6.1. `hasSize` ([chapitre2.html#objectHasSize](#))
 - 2.1.6.2. `isCallable`
([chapitre2.html#objectIsCallable](#))
 - 2.1.6.3. `isCloneOf`
([chapitre2.html#objectIsCloneOf](#))
 - 2.1.6.4. `isEmpty` ([chapitre2.html#objectIsEmpty](#))
 - 2.1.6.5. `isEqualTo`
([chapitre2.html#objectIsEqualTo](#))
 - 2.1.6.6. `isIdenticalTo`
([chapitre2.html#objectIsIdenticalTo](#))
 - 2.1.6.7. `isInstanceOf`
([chapitre2.html#objectIsInstanceOf](#))
 - 2.1.6.8. `isNotCallable`
([chapitre2.html#objectIsNotCallable](#))
 - 2.1.6.9. `isNotEqualTo`
([chapitre2.html#objectIsNotEqualTo](#))
 - 2.1.6.10. `isNotIdenticalTo`
([chapitre2.html#objectIsNotIdenticalTo](#))
- 2.1.7. `dateInterval` ([chapitre2.html#dateInterval](#))
 - 2.1.7.1. `isCloneOf`
([chapitre2.html#dateIntervallsCloneOf](#))
 - 2.1.7.2. `isEqualTo`
([chapitre2.html#dateIntervallsEqualTo](#))
 - 2.1.7.3. `isGreaterThan`
([chapitre2.html#dateIntervallsGreaterThan](#))
 - 2.1.7.4. `isGreaterThanOrEqualTo`
([chapitre2.html#dateIntervallsGreaterThanOrEqualTo](#))
 - 2.1.7.5. `isIdenticalTo`
([chapitre2.html#dateIntervallsIdenticalTo](#))
 - 2.1.7.6. `isInstanceOf`
([chapitre2.html#dateIntervallsInstanceOf](#))
 - 2.1.7.7. `isLessThan`
([chapitre2.html#dateIntervallsLessThan](#))
 - 2.1.7.8. `isLessThanOrEqualTo`
([chapitre2.html#dateIntervallsLessThanOrEqualTo](#))
 - 2.1.7.9. `isNotEqualTo`
([chapitre2.html#dateIntervallsNotEqualTo](#))
 - 2.1.7.10. `isNotIdenticalTo`
([chapitre2.html#dateIntervallsNotIdenticalTo](#))
 - 2.1.7.11. `isZero` ([chapitre2.html#dateIntervallsZero](#))
- 2.1.8. `dateTime` ([chapitre2.html#dateTime](#))
 - 2.1.8.1. `hasDate`
([chapitre2.html#dateTimeHasDate](#))
 - 2.1.8.2. `hasDateAndTime`
([chapitre2.html#dateTimeHasDateAndTime](#))
 - 2.1.8.3. `hasDay` ([chapitre2.html#dateTimeHasDay](#))
 - 2.1.8.4. `hasHours`
([chapitre2.html#dateTimeHasHours](#))
 - 2.1.8.5. `hasMinutes`
([chapitre2.html#dateTimeHasMinutes](#))
 - 2.1.8.6. `hasMonth`
([chapitre2.html#dateTimeHasMonth](#))
 - 2.1.8.7. `hasSeconds`
([chapitre2.html#dateTimeHasSeconds](#))
 - 2.1.8.8. `hasTime`
([chapitre2.html#dateTimeHasTime](#))
 - 2.1.8.9. `hasTimezone`
([chapitre2.html#dateTimeHasTimezone](#))
 - 2.1.8.10. `hasYear`

- (chapitre2.html#dateTimeHasYear)
- 2.1.8.11. isCloneOf
(chapitre2.html#dateTimeIsCloneOf)
- 2.1.8.12. isEqualTo
(chapitre2.html#dateTimeIsEqualTo)
- 2.1.8.13. isIdenticalTo
(chapitre2.html#dateTimeIsIdenticalTo)
- 2.1.8.14. isInstanceOf
(chapitre2.html#dateTimeIsInstanceOf)
- 2.1.8.15. isNotEqualTo
(chapitre2.html#dateTimeIsNotEqualTo)
- 2.1.8.16. isNotIdenticalTo
(chapitre2.html#dateTimeIsNotIdenticalTo)
- 2.1.9. mysqlDateTime
(chapitre2.html#mysqlDateTime)
- 2.1.9.1. hasDate
(chapitre2.html#mysqlDateTimeHasDate)
- 2.1.9.2. hasDateAndTime
(chapitre2.html#mysqlDateTimeHasDateAndTime)
- 2.1.9.3. hasDay
(chapitre2.html#mysqlDateTimeHasDay)
- 2.1.9.4. hasHours
(chapitre2.html#mysqlDateTimeHasHours)
- 2.1.9.5. hasMinutes
(chapitre2.html#mysqlDateTimeHasMinutes)
- 2.1.9.6. hasMonth
(chapitre2.html#mysqlDateTimeHasMonth)
- 2.1.9.7. hasSeconds
(chapitre2.html#mysqlDateTimeHasSeconds)
- 2.1.9.8. hasTime
(chapitre2.html#mysqlDateTimeHasTime)
- 2.1.9.9. hasTimezone
(chapitre2.html#mysqlDateTimeHasTimezone)
- 2.1.9.10. hasYear
(chapitre2.html#mysqlDateTimeHasYear)
- 2.1.9.11. isCloneOf
(chapitre2.html#mysqlDateTimeIsCloneOf)
- 2.1.9.12. isEqualTo
(chapitre2.html#mysqlDateTimeIsEqualTo)
- 2.1.9.13. isIdenticalTo
(chapitre2.html#mysqlDateTimeIsIdenticalTo)
- 2.1.9.14. isInstanceOf
(chapitre2.html#mysqlDateTimeIsInstanceOf)
- 2.1.9.15. isNotEqualTo
(chapitre2.html#mysqlDateTimeIsNotEqualTo)
- 2.1.9.16. isNotIdenticalTo
(chapitre2.html#mysqlDateTimeIsNotIdenticalTo)
- 2.1.10. exception (chapitre2.html#exception)
- 2.1.10.1. hasCode (chapitre2.html#hasCode)
- 2.1.10.2. hasDefaultCode
(chapitre2.html#hasDefaultCode)
- 2.1.10.3. hasMessage
(chapitre2.html#hasMessage)
- 2.1.10.4. hasNestedException
(chapitre2.html#hasNestedException)
- 2.1.10.5. isCloneOf
(chapitre2.html#exceptionIsCloneOf)
- 2.1.10.6. isEqualTo
(chapitre2.html#exceptionIsEqualTo)
- 2.1.10.7. isIdenticalTo
(chapitre2.html#exceptionIsIdenticalTo)

- 2.1.10.8. `isInstanceOf`
([chapitre2.html#exceptionIsInstanceOf](#))
- 2.1.10.9. `isNotEqualTo`
([chapitre2.html#exceptionIsNotEqualTo](#))
- 2.1.10.10. `isNotIdenticalTo`
([chapitre2.html#exceptionIsNotIdenticalTo](#))
- 2.1.10.11. `message` ([chapitre2.html#message](#))
- 2.1.11. `array` ([chapitre2.html#array](#))
 - 2.1.11.1. `contains` ([chapitre2.html#arrayContains](#))
 - 2.1.11.2. `containsValues`
([chapitre2.html#containsValues](#))
 - 2.1.11.3. `hasKey` ([chapitre2.html#hasKey](#))
 - 2.1.11.4. `hasKeys` ([chapitre2.html#hasKeys](#))
 - 2.1.11.5. `hasSize` ([chapitre2.html#arrayHasSize](#))
 - 2.1.11.6. `isEmpty` ([chapitre2.html#arrayIsEmpty](#))
 - 2.1.11.7. `isEqualTo`
([chapitre2.html#arrayIsEqualTo](#))
 - 2.1.11.8. `isIdenticalTo`
([chapitre2.html#arrayIsIdenticalTo](#))
 - 2.1.11.9. `isNotEmpty`
([chapitre2.html#arrayIsNotEmpty](#))
 - 2.1.11.10. `isNotEqualTo`
([chapitre2.html#arrayIsNotEqualTo](#))
 - 2.1.11.11. `isNotIdenticalTo`
([chapitre2.html#arrayIsNotIdenticalTo](#))
 - 2.1.11.12. `keys` ([chapitre2.html#keys](#))
 - 2.1.11.13. `notContains`
([chapitre2.html#arrayNotContains](#))
 - 2.1.11.14. `notContainsValues`
([chapitre2.html#notContainsValues](#))
 - 2.1.11.15. `notHasKey` ([chapitre2.html#notHasKey](#))
 - 2.1.11.16. `notHasKeys`
([chapitre2.html#notHasKeys](#))
 - 2.1.11.17. `size` ([chapitre2.html#size](#))
 - 2.1.11.18. `strictlyContains`
([chapitre2.html#strictlyContains](#))
 - 2.1.11.19. `strictlyContainsValues`
([chapitre2.html#strictlyContainsValues](#))
 - 2.1.11.20. `strictlyNotContains`
([chapitre2.html#strictlyNotContains](#))
 - 2.1.11.21. `strictlyNotContainsValues`
([chapitre2.html#strictlyNotContainsValues](#))
- 2.1.12. `string` ([chapitre2.html#string](#))
 - 2.1.12.1. `contains` ([chapitre2.html#stringContains](#))
 - 2.1.12.2. `hasLength`
([chapitre2.html#stringHasLength](#))
 - 2.1.12.3. `hasLengthGreaterThan`
([chapitre2.html#stringHasLengthGreaterThan](#))
 - 2.1.12.4. `hasLengthLessThan`
([chapitre2.html#stringHasLengthLessThan](#))
 - 2.1.12.5. `isEmpty` ([chapitre2.html#stringIsEmpty](#))
 - 2.1.12.6. `isEqualTo`
([chapitre2.html#stringIsEqualTo](#))
 - 2.1.12.7. `isEqualToContentsOfFile`
([chapitre2.html#stringIsEqualToContentsOfFile](#))
 - 2.1.12.8. `isIdenticalTo`
([chapitre2.html#stringIsIdenticalTo](#))
 - 2.1.12.9. `isNotEmpty`
([chapitre2.html#stringIsNotEmpty](#))
 - 2.1.12.10. `isNotEqualTo`
([chapitre2.html#stringIsNotEqualTo](#))

- 2.1.12.11. `isNotIdenticalTo`
(chapitre2.html#stringIsNotIdenticalTo)
- 2.1.12.12. `length` (chapitre2.html#length)
- 2.1.12.13. `match` (chapitre2.html#stringMatch)
- 2.1.12.14. `notContains`
(chapitre2.html#stringNotContains)
- 2.1.13. `castToString` (chapitre2.html#castToString)
 - 2.1.13.1. `contains`
(chapitre2.html#castToStringContains)
 - 2.1.13.2. `notContains`
(chapitre2.html#castToStringNotContains)
 - 2.1.13.3. `hasLength`
(chapitre2.html#castToStringHasLength)
 - 2.1.13.4. `hasLengthGreaterThan`
(chapitre2.html#castToStringHasLengthGreaterThan)
 - 2.1.13.5. `hasLengthLessThan`
(chapitre2.html#castToStringHasLengthLessThan)
 - 2.1.13.6. `isEmpty`
(chapitre2.html#castToStringIsEmpty)
 - 2.1.13.7. `isEqualTo`
(chapitre2.html#castToStringIsEqualTo)
 - 2.1.13.8. `isEqualToContentsOfFile`
(chapitre2.html#castToStringIsEqualToContentsOfFile)
 - 2.1.13.9. `isIdenticalTo`
(chapitre2.html#castToStringIsIdenticalTo)
 - 2.1.13.10. `isNotEmpty`
(chapitre2.html#castToStringIsNotEmpty)
 - 2.1.13.11. `isNotEqualTo`
(chapitre2.html#castToStringIsNotEqualTo)
 - 2.1.13.12. `isNotIdenticalTo`
(chapitre2.html#castToStringIsNotIdenticalTo)
 - 2.1.13.13. `match`
(chapitre2.html#castToStringMatch)
- 2.1.14. `hash` (chapitre2.html#hash)
 - 2.1.14.1. `contains` (chapitre2.html#hashContains)
 - 2.1.14.2. `isEqualTo`
(chapitre2.html#hashIsEqualTo)
 - 2.1.14.3. `isEqualToContentsOfFile`
(chapitre2.html#hashIsEqualToContentsOfFile)
 - 2.1.14.4. `isIdenticalTo`
(chapitre2.html#hashIsIdenticalTo)
 - 2.1.14.5. `isMd5` (chapitre2.html#isMd5)
 - 2.1.14.6. `isNotEqualTo`
(chapitre2.html#hashIsNotEqualTo)
 - 2.1.14.7. `isNotIdenticalTo`
(chapitre2.html#hashIsNotIdenticalTo)
 - 2.1.14.8. `isSha1` (chapitre2.html#isSha1)
 - 2.1.14.9. `isSha256` (chapitre2.html#isSha256)
 - 2.1.14.10. `isSha512` (chapitre2.html#isSha512)
 - 2.1.14.11. `notContains`
(chapitre2.html#hashNotContains)
- 2.1.15. `output` (chapitre2.html#output)
 - 2.1.15.1. `contains` (chapitre2.html#outputContains)
 - 2.1.15.2. `hasLength`
(chapitre2.html#outputHasLength)
 - 2.1.15.3. `hasLengthGreaterThan`
(chapitre2.html#outputHasLengthGreaterThan)
 - 2.1.15.4. `hasLengthLessThan`
(chapitre2.html#outputHasLengthLessThan)
 - 2.1.15.5. `isEmpty` (chapitre2.html#outputIsEmpty)
 - 2.1.15.6. `isEqualTo`

- (chapitre2.html#outputIsEqualTo)
- 2.1.15.7. isEqualToContentsOfFile
(chapitre2.html#outputIsEqualToContentsOfFile)
- 2.1.15.8. isIdenticalTo
(chapitre2.html#outputIsIdenticalTo)
- 2.1.15.9. isEmpty
(chapitre2.html#outputIsEmpty)
- 2.1.15.10. isNotEqualTo
(chapitre2.html#outputIsNotEqualTo)
- 2.1.15.11. isNotIdenticalTo
(chapitre2.html#outputIsNotIdenticalTo)
- 2.1.15.12. match (chapitre2.html#outputMatch)
- 2.1.15.13. notContains
(chapitre2.html#outputNotContains)
- 2.1.16. utf8String (chapitre2.html#utf8String)
 - 2.1.16.1. contains
(chapitre2.html#utf8StringContains)
 - 2.1.16.2. hasLength
(chapitre2.html#utf8StringHasLength)
 - 2.1.16.3. hasLengthGreaterThan
(chapitre2.html#utf8StringHasLengthGreaterThan)
 - 2.1.16.4. hasLengthLessThan
(chapitre2.html#utf8StringHasLengthLessThan)
 - 2.1.16.5. isEmpty
(chapitre2.html#utf8StringIsEmpty)
 - 2.1.16.6. isEqualTo
(chapitre2.html#utf8StringIsEqualTo)
 - 2.1.16.7. isEqualToContentsOfFile
(chapitre2.html#utf8StringIsEqualToContentsOfFile)
 - 2.1.16.8. isIdenticalTo
(chapitre2.html#utf8StringIsIdenticalTo)
 - 2.1.16.9. isEmpty
(chapitre2.html#utf8StringIsEmpty)
 - 2.1.16.10. isNotEqualTo
(chapitre2.html#utf8StringIsNotEqualTo)
 - 2.1.16.11. isNotIdenticalTo
(chapitre2.html#utf8StringIsNotIdenticalTo)
 - 2.1.16.12. match (chapitre2.html#utf8StringMatch)
 - 2.1.16.13. notContains
(chapitre2.html#utf8StringNotContains)
- 2.1.17. afterDestructionOf
(chapitre2.html#afterDestructionOf)
- 2.1.18. error (chapitre2.html#error)
 - 2.1.18.1. exists (chapitre2.html#exists)
 - 2.1.18.2. notExists (chapitre2.html#notExists)
 - 2.1.18.3. withType (chapitre2.html#withType)
- 2.1.19. class (chapitre2.html#class)
 - 2.1.19.1. hasInterface
(chapitre2.html#hasInterface)
 - 2.1.19.2. hasMethod (chapitre2.html#hasMethod)
 - 2.1.19.3. hasNoParent
(chapitre2.html#hasNoParent)
 - 2.1.19.4. hasParent (chapitre2.html#hasParent)
 - 2.1.19.5. isAbstract (chapitre2.html#isAbstract)
 - 2.1.19.6. isSubclassOf
(chapitre2.html#isSubclassOf)
- 2.1.20. mock (chapitre2.html#mock)
 - 2.1.20.1. call (chapitre2.html#call)
 - 2.1.20.1.1. atLeastOnce
(chapitre2.html#atLeastOnce)
 - 2.1.20.1.2. exactly (chapitre2.html#exactly)

- 2.1.20.1.3. never (chapitre2.html#never)
- 2.1.20.1.4. once/twice/thrice (chapitre2.html#once-twice-thrice)
- 2.1.20.1.5. withAnyArguments (chapitre2.html#withAnyArguments)
- 2.1.20.1.6. withArguments (chapitre2.html#withArguments)
- 2.1.20.1.7. withIdenticalArguments (chapitre2.html#withIdenticalArguments)
- 2.1.20.2. wasCalled (chapitre2.html#wasCalled)
- 2.1.20.3. wasNotCalled (chapitre2.html#wasNotCalled)
- 2.1.21. stream (chapitre2.html#stream)
 - 2.1.21.1. isRead (chapitre2.html#isRead)
 - 2.1.21.2. isWrite (chapitre2.html#isWrite)
- 2.2. Aide à l'écriture (chapitre2.html#Aide-a-l-ecriture)
 - 2.2.1. if, and, then (chapitre2.html#if-and-then)
 - 2.2.2. when (chapitre2.html#when)
 - 2.2.3. assert (chapitre2.html#assert)
- 2.3. Le mode loop (chapitre2.html#Le-mode-loop)
- 2.4. Le mode debug (chapitre2.html#Le-mode-debug)
 - 2.4.1. dump (chapitre2.html#dump)
 - 2.4.2. stop (chapitre2.html#stop)
 - 2.4.3. executeOnFailure (chapitre2.html#executeOnFailure)
- 2.5. Les méthodes d'initialisation (chapitre2.html#Les-methodes-d-initialisation)
- 2.6. Fournisseurs de données (data provider) (chapitre2.html#Fournisseurs-de-donnees-data-provider)
- 2.7. Les bouchons (mock) (chapitre2.html#Les-bouchons-mock)
 - 2.7.1. Générer un bouchon (chapitre2.html#Generer-un-bouchon)
 - 2.7.2. Le générateur de bouchon (chapitre2.html#Le-generateur-de-bouchon)
 - 2.7.2.1. Changer le nom de la classe (chapitre2.html#Changer-le-nom-de-la-classe)
 - 2.7.2.2. Shunter les appels aux méthodes parentes (chapitre2.html#Shunter-les-appels-aux-methodes-parentes)
 - 2.7.2.3. Rendre une méthode orpheline (chapitre2.html#Rendre-une-methode-orpheline)
 - 2.7.3. Modifier le comportement d'un bouchon (chapitre2.html#Modifier-le-comportement-d-un-bouchon)
 - 2.7.3.1. methods (chapitre2.html#methods)
 - 2.7.3.2. methodsWhichMatch (chapitre2.html#methodsWhichMatch)
 - 2.7.4. Cas particulier du constructeur (chapitre2.html#Cas-particulier-du-constructeur)
 - 2.7.5. Tester un bouchon (chapitre2.html#Tester-un-bouchon)

3. Lancement des tests (chapitre3.html#Lancement-des-tests)

- 3.1. Exécutable (chapitre3.html#Executable)
 - 3.1.1. Avec l'archive phar (chapitre3.html#Avec-l-archive-phar)
 - 3.1.1.1. linux / mac (chapitre3.html#pharLinuxMac)
 - 3.1.1.2. windows (chapitre3.html#pharWindows)

- 3.1.2. Avec les sources (chapitre3.html#Avec-les-sources)
 - 3.1.2.1. linux / mac (chapitre3.html#sourceLinuxMac)
 - 3.1.2.2. windows (chapitre3.html#sourceWindows)
- 3.1.3. Exemples dans le reste de la documentation (chapitre3.html#Exemples-dans-le-reste-de-la-documentation)
- 3.2. Fichiers à exécuter (chapitre3.html#Fichiers-a-executer)
 - 3.2.1. Par fichiers (chapitre3.html#Par-fichiers)
 - 3.2.2. Par répertoires (chapitre3.html#Par-repertoires)
- 3.3. Filtres (chapitre3.html#Filtres)
 - 3.3.1. Par espace de noms (chapitre3.html#Par-espace-de-noms)
 - 3.3.2. Une classe ou une méthode (chapitre3.html#Une-classe-ou-une-methode)
 - 3.3.3. Tags (chapitre3.html#Tags)
- 3.4. Fichier de configuration (chapitre3.html#Fichier-de-configuration)
 - 3.4.1. Couverture du code (chapitre3.html#Couverture-du-code)
 - 3.4.2. Notifications (chapitre3.html#Notifications)
 - 3.4.2.1. Growl (chapitre3.html#Growl)
 - 3.4.2.2. Mac OS X Notification Center (chapitre3.html#OSXNotificationCenter)
 - 3.4.2.3. Libnotify (chapitre3.html#Libnotify)
- 3.5. Fichier de bootstrap (chapitre3.html#Fichier-de-bootstrap)
- 3.6. Option de la ligne de commande (chapitre3.html#Option-de-la-ligne-de-commande)
 - 3.6.1. -bf <file> / --bootstrap-file <file> (chapitre3.html#bf-file-bootstrap-file-file)
 - 3.6.2. -c <file> / --configuration <file> (chapitre3.html#c-file-configuration-file)
 - 3.6.3. -d <directories> / --directories <directories> (chapitre3.html#d-directories-directories-directories)
 - 3.6.4. --debug (chapitre3.html#debug)
 - 3.6.5. -drt <string> / --default-report-title <string> (chapitre3.html#drt-string-default-report-title-string)
 - 3.6.6. -f <files> / --files <files> (chapitre3.html#f-files-files-files)
 - 3.6.7. -ft / --force-terminal (chapitre3.html#ft-force-terminal)
 - 3.6.8. -g <pattern> / --glob <pattern> (chapitre3.html#g-pattern-glob-pattern)
 - 3.6.9. -h / --help (chapitre3.html#h-help)
 - 3.6.10. -l / --loop (chapitre3.html#l-loop)
 - 3.6.11. -m <class::method> / --methods <class::methods> (chapitre3.html#m-class-method-methods-class-methods)
 - 3.6.12. -mcn <integer> / --max-children-number <integer> (chapitre3.html#mcn-integer-max-children-number-integer)
 - 3.6.13. -ncc / --no-code-coverage (chapitre3.html#ncc-no-code-coverage)
 - 3.6.14. -nccfc <classes> / --no-code-coverage-for-classes <classes> (chapitre3.html#nccfc-classes-no-code-coverage-for-classes-classes)
 - 3.6.15. -nccfns <namespaces> / --no-code-

coverage-for-namespaces <namespaces>
(chapitre3.html#nccfns-namespaces-no-code-coverage-for-namespaces-namespaces)
3.6.16. -nccid <directories> / --no-code-coverage-in-directories <directories> (chapitre3.html#nccid-directories-no-code-coverage-in-directories-directories)
3.6.17. -ns <namespaces> / --namespaces <namespaces> (chapitre3.html#ns-namespaces-namespaces-namespaces)
3.6.18. -p <file> / --php <file> (chapitre3.html#p-file-php-file)
3.6.19. -sf <file> / --score-file <file> (chapitre3.html#sf-file-score-file-file)
3.6.20. -t <tags> / --tags <tags> (chapitre3.html#t-tags-tags-tags)
3.6.21. --test-all (chapitre3.html#test-all)
3.6.22. --test-it (chapitre3.html#test-it)
3.6.23. -tfe <extensions> / --test-file-extensions <extensions> (chapitre3.html#tfe-extensions-test-file-extensions-extensions)
3.6.24. -ulr / --use-light-report (chapitre3.html#ulr-use-light-report)
3.6.25. -v / --version (chapitre3.html#v-version)

4. Cookbook (chapitre4.html#Cookbook)

- 4.1. Test d'un singleton (chapitre4.html#Test-d-un-singleton)
- 4.2. Utilisation dans behat (chapitre4.html#Utilisation-dans-behat)
- 4.3. Utilisation dans Jenkins (ou Hudson)
(chapitre4.html#Utilisation-dans-Jenkins-ou-Hudson)
 - 4.3.1. Étape 1 : Ajout d'un rapport xUnit à la configuration atoum (chapitre4.html#Etape-1-Ajout-d-un-rapport-xUnit-a-la-configuration-atoum)
 - 4.3.1.1. Vous n'avez pas de fichier de configuration (chapitre4.html#Vous-n-avez-pas-de-fichier-de-configuration)
 - 4.3.1.2. Vous avez déjà un fichier de configuration (chapitre4.html#Vous-avez-deja-un-fichier-de-configuration)
 - 4.3.2. Étape 2 : Tester la configuration (chapitre4.html#Etape-2-Tester-la-configuration)
 - 4.3.3. Étape 3 : Lancement des tests via Jenkins (ou Hudson) (chapitre4.html#Etape-3-Lancement-des-tests-via-Jenkins-ou-Hudson)
 - 4.3.4. Étape 4 : Publier le rapport avec Jenkins (ou Hudson) (chapitre4.html#Etape-4-Publier-le-rapport-avec-Jenkins-ou-Hudson)
- 4.4. Hook git (chapitre4.html#Hook-git)
 - 4.4.1. Étape 1 : Création du script à exécuter (chapitre4.html#Etape-1-Creation-du-script-a-executer)
 - 4.4.2. Étape 2 : Ajout des droits d'exécution (chapitre4.html#Etape-2-Ajout-des-droits-d-execution)
- 4.5. Changer l'espace de nom par défaut (chapitre4.html#Changer-l-espace-de-nom-par-defaut)
- 4.6. Utilisation avec ezPublish (chapitre4.html#Utilisation-avec-ezPublish)

4.7. Utilisation avec Symfony 2

(chapitre4.html#Utilisation-avec-Symfony-2)

4.7.1. Étape 1: installation d'atoum

(chapitre4.html#Etape-1-installation-d-atoum)

4.7.2. Étape 2: création de la classe de test

(chapitre4.html#Etape-2-creation-de-la-classe-de-test)

4.7.3. Étape 3: écriture d'un test

(chapitre4.html#Etape-3-ecriture-d-un-test)

4.7.4. Étape 4: lancement des tests

(chapitre4.html#Etape-4-lancement-des-tests)

4.8. Utilisation avec symfony 1.4

(chapitre4.html#Utilisation-avec-symfony-1-4)

4.8.1. Installation (chapitre4.html#Installation)

4.8.1.1. En utilisant composer (chapitre4.html#En-utilisant-composer)

4.8.1.2. En utilisant des submodules git

(chapitre4.html#En-utilisant-des-submodules-git)

4.8.2. Ecrire les tests (chapitre4.html#Ecrire-les-tests)

4.8.3. Lancer les tests (chapitre4.html#Lancer-les-tests)

5. Participer

(chapitre5.html#Participer)

5.1. Comment participer (chapitre5.html#Comment-participer)

5.2. Convention de codage

(chapitre5.html#Convention-de-codage)

6. Citations (chapitre6.html#Citations)

Something is wrong in the documentation ? Help us fix this by [reporting errors and problems !](https://github.com/atoum/atoum-documentation/issues) (<https://github.com/atoum/atoum-documentation/issues>).