

Des Unit Tests pour vos Plugins



Hein ?

Pourquoi ?

Comment ?

T'es sûr ?

C'est utile ?

C'est fastoche ?

🌀 Foreword 🌀

Not French ? No sweat !

Just read the code

and

wait for the blog post

(no ETA)



Bonjour ! je m'appelle

Ozh

Ça se prononce « **ose** »
comme dans « **rose** »

Mes fils écoutent
de la musique douce

I ♥ **WordPress** depuis **11** ans

Premier plugin en août 2004 !!1!!1\$!

des **plugins**, des articles

<http://planetozh.com/blog/>

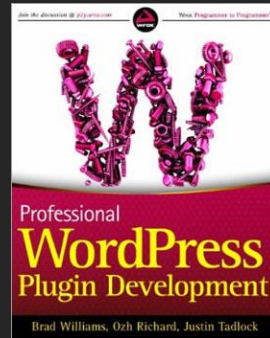
un (super) **livre**

sur les plugins

<http://amzn.to/plugindevbook>

mon **1er** WordCamp

^_(ツ)_/



Ozh

et sinon, **YOURLS**



<https://yourls.org/>



ozh@ozh.org



Inuit Test ?

(wtf)



Menu

- WTF « **Unit Test** » ?
- Installer les **outils**
- Bases de **PHPUnit**
- Cas pratique : **plugin + tests**
- Bonnes pratiques & devoirs à la maison

**oops #fail ...
on est vendredi soir vous dites?**



Sans Unit Test

```
define( 'WP_DEBUG', true );
```

```
var_dump( $wp_truc );
```

```
die( 'HERE IT WORKS' );
```

```
debug_backtrace( );
```

```
Chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
rhaaââ mais fuuuuuuu là,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
Ah, ayet, trouvé.
```

Sans Unit Test :- (

```
define( 'WP_DEBUG', true );  
  
var_dump( $wp_truc );  
  
die( 'HERE IT WORKS' );  
  
debug_backtrace( );
```

```
Chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
rhaaââ mais fuuuuuuuu là,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
chercher, modifier, tester,  
Ah, ayet, trouvé.
```



Avec Unit Test

```
$ phpunit
```

```
Installing...
```

```
Running as single site... To run multisite, use -c tests/phpunit/multisite.xml
```

```
Not running ajax tests. To execute these, use --group ajax.
```

```
Not running ms-files tests. To execute these, use --group ms-files.
```

```
Not running external-http tests. To execute these, use --group external-http.
```

```
PHPUnit 4.4.1 by Sebastian Bergmann.
```

```
Configuration read from D:\home\wordpress\wp-content\plugins\ozh-demo\phpunit.xml
```

```
.....F..... 61 / 124 ( 49%)  
..... 119 / 124 ( 95%)  
.....
```

```
Time: 57.1 seconds, Memory: 66.6Mb
```

```
There was 1 failure:
```

```
1) This_Feature_Subfeature_Test::test_do_that_thing  
Failed asserting that true is false.
```

```
D:\home\wordpress\wp-content\plugins\ozh-demo\tests\test-mockfs.php:48
```

```
FAILURES!
```

```
Tests: 124, Assertions: 1337, Failures: 1.
```

Avec Unit Test :-)

```
$ phpunit
```

```
Installing...
```

```
Running as single site... To run multisite, use -c tests/phpunit/multisite.xml
```

```
Not running ajax tests. To execute these, use --
```

```
Not running ms-files tests. To execute these, us
```

```
Not running external-http tests. To execute thes
```

```
PHPUnit 4.4.1 by Sebastian Bergmann.
```

```
Configuration read from D:\home\wordpress\wp-con
```

```
.....F.....  
.....  
.....
```

```
Time: 57.1 seconds, Memory: 66.6Mb
```

```
There was 1 failure:
```

```
1) This_Feature_Subfeature_Test::test_do_that_th  
Failed asserting that true is false.
```

```
D:\home\wordpress\wp-content\plugins\ozh-demo\te
```

```
FAILURES!
```

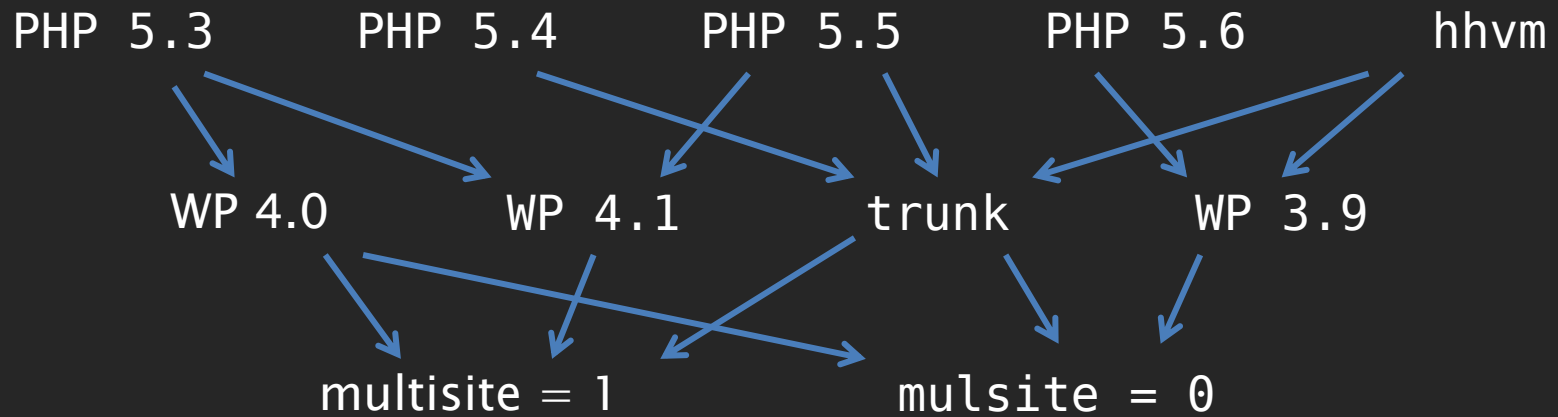
```
Tests: 124, Assertions: 1337, Failures: 1.
```



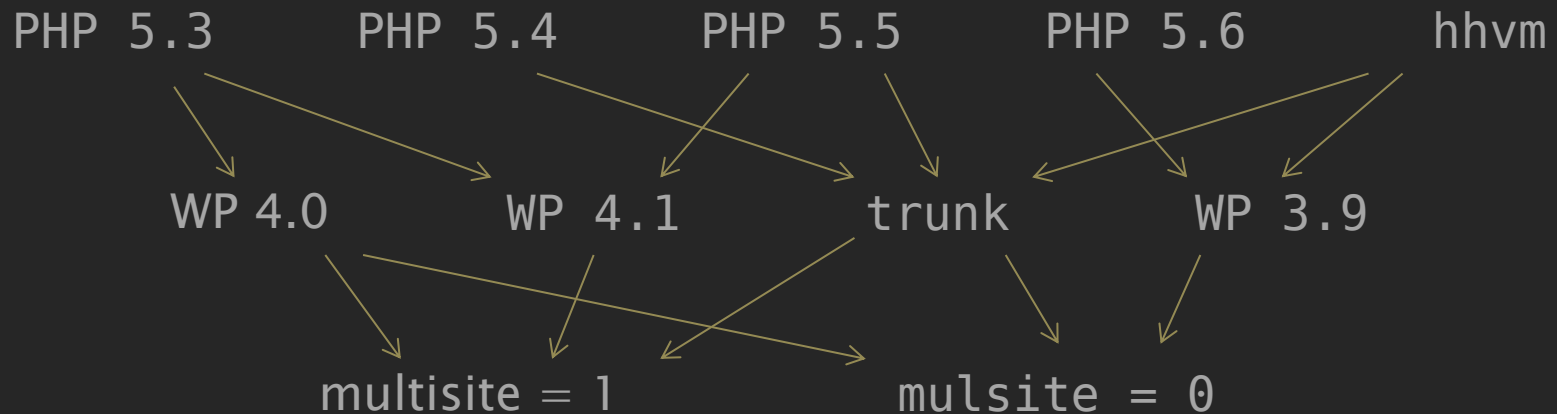
Unit Test

Des bouts de **code simples**
pour **vérifier** que
chaque **partie élémentaire**
de l'ensemble du code complexe
fonctionne comme attendu

Automatiser !



Automatiser !



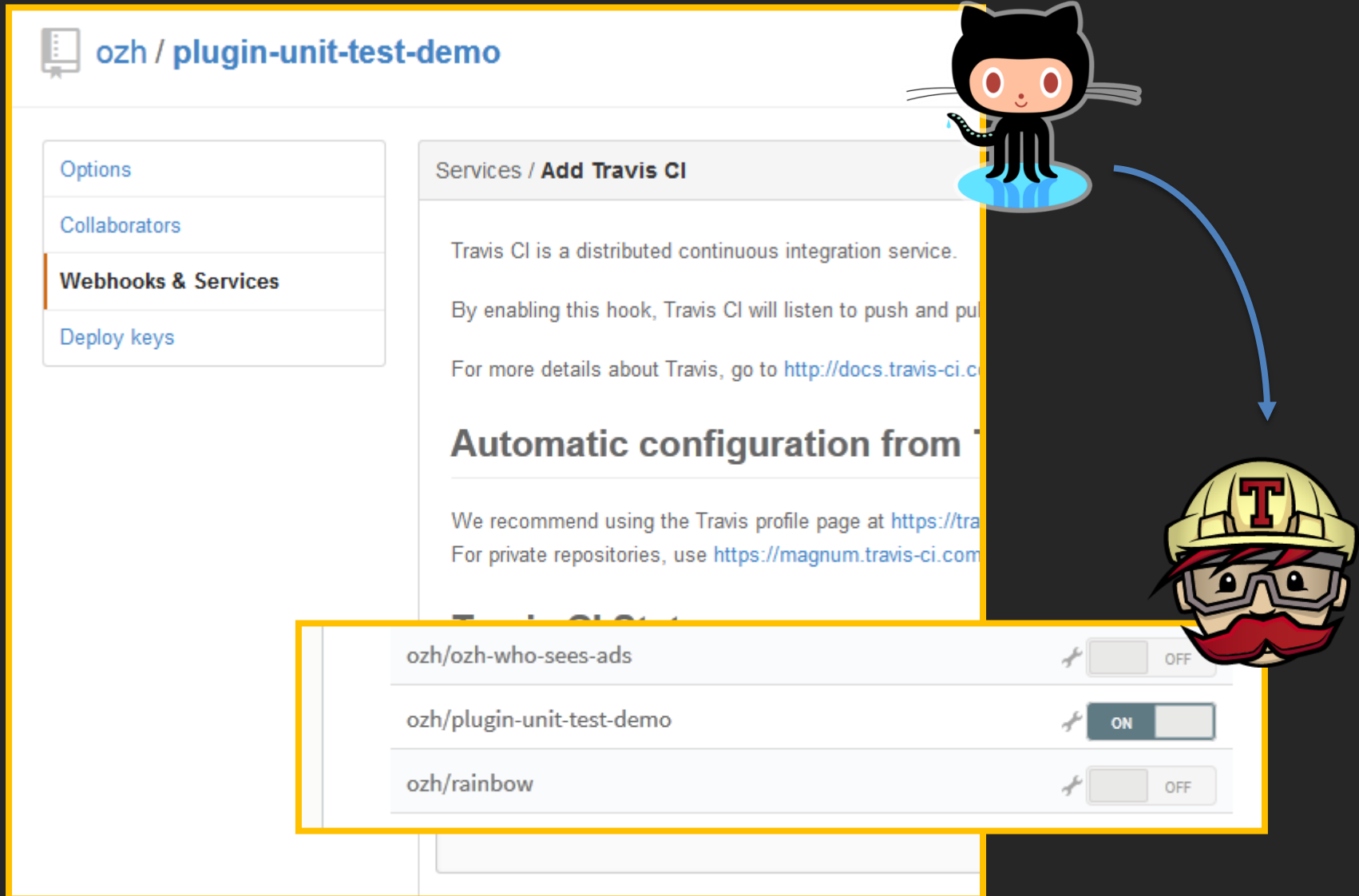
Easy Digital Download :

21 configurations

350 tests avec 1100 "assertions"

= 23 000 vérifications simples
à chaque commit ou pull request

github.com → travis-ci.org



The image shows a screenshot of the GitHub repository settings for 'ozh / plugin-unit-test-demo'. The left sidebar contains navigation links: 'Options', 'Collaborators', 'Webhooks & Services' (which is selected), and 'Deploy keys'. The main content area is titled 'Services / Add Travis CI'. It contains the following text:

Travis CI is a distributed continuous integration service.

By enabling this hook, Travis CI will listen to push and pull events.

For more details about Travis, go to <http://docs.travis-ci.com>

Automatic configuration from Travis

We recommend using the Travis profile page at <https://travis-ci.org>

For private repositories, use <https://magnum.travis-ci.com>

Below this text is a table showing the status of Travis CI integration for different repositories:

Repository	Status
ozh/ozh-who-sees-ads	OFF
ozh/plugin-unit-test-demo	ON
ozh/rainbow	OFF

Two cartoon characters are overlaid on the image. A black cat character (GitHub Octocat) is positioned above the 'Services / Add Travis CI' section. A blue arrow points from the Octocat to a character wearing a yellow hard hat with a red 'T' and a red mustache, who is positioned below the table.

github.com → travis-ci.org

ozh/plugin-unit-test-demo

build failing

Settings

Current

Build History

Pull Requests

Branch Summary



master - Commit all the stuff



ozh authored and committed

#1 failed

ran for 1 min 13 sec

less than a minute ago

Commit dc81708

Compare dc817085736d

Build Matrix

Job	Duration	Finished	PHP	ENV	OS
✓ 1.1	10 sec	2 minutes ago	5.3	WP_VERSION=latest WP_MULTISITE=0	linux
✓ 1.2	10 sec	2 minutes ago	5.3	WP_VERSION=latest WP_MULTISITE=1	linux
✗ 1.3	9 sec	2 minutes ago	5.3	WP_VERSION=3.8 WP_MULTISITE=0	linux
✗ 1.4	11 sec	about a minute ago	5.3	WP_VERSION=3.8 WP_MULTISITE=1	linux
✓ 1.5	9 sec	2 minutes ago	5.4	WP_VERSION=latest WP_MULTISITE=0	linux
✓ 1.6	9 sec	less than a minute ago	5.4	WP_VERSION=latest WP_MULTISITE=1	linux

Elle est bonne ta Pull Request ?

Include item quantities when calculating cart quantity.
#2911

 Open **stephenharris** wants to merge 1 commit into `easydigitaldownloads:master` from `stephenharris:cart-quantity`

 Conversation 2  Commits 1  Files changed 1

 **stephenharris** commented 22 days ago

This might be intended behaviour but `edd_get_cart_quantity()` doesn't take item quantities into account. So if I have a cart containing one product with quantity 2, `edd_get_cart_quantity()` returns 1 rather two.

This pull request changes that behaviour, summing the quantities of each item in the cart.

  Include item quantities when calculating cart quantity.  c02b997

“ With unit tests and continuous testing, there is always an **extra pair of eyes** looking at your code, helping you **find the bugs you didn't know existed.**

Find one bug with unit tests **before a release** is pushed out and you will **never NOT have unit test again.**”

- Pippin Williamson,
Easy Digital Downloads & autres

Bon, on
code ?

Conventions & Pictogrammes



`my_file.php`



`my_file.php (résumé)`



`$ bash run_script.sh`



`function do_stuff()`



`class Test_Do_Stuff{ }`



`Bla bla zoom`



Ingrédients

- Une **CLI** confortable

Minimum vital: bash, wget, curl, mysqladmin

- **SVN**

- **GIT**

Noob? → « Efficient Git setup on Windows » <http://ozh.in/vl>

- 
- **PHPUnit**

- **WP-CLI**

Installer PHPUnit

```
$ wget https://phar.phpunit.de/phpunit.phar
```

```
$ chmod +x phpunit.phar (si besoin)
```

```
$ mv phpunit.phar /usr/local/bin/phpunit  
ou, dans .bash_profile :  
alias phpunit="php d:/php/phpunit.phar"
```

```
$ phpunit --version  
PHPUnit 4.4.1 by Sebastian Bergmann
```

Installer WP-CLI

Même principe

<https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar>

```
$ wp --version  
WP-CLI 0.17.1
```

~/.bash_profile

...

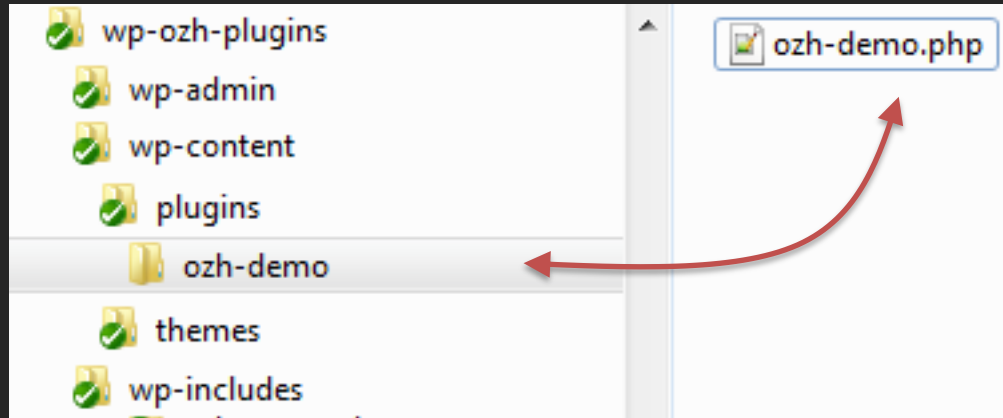
For wp-cli & phpunit tests

export WP_TESTS_DIR=/d/planetozh/wp-unittest-suite/wordpress-tests-lib

export WP_CORE_DIR=/d/planetozh/wp-unittest-suite/wordpress

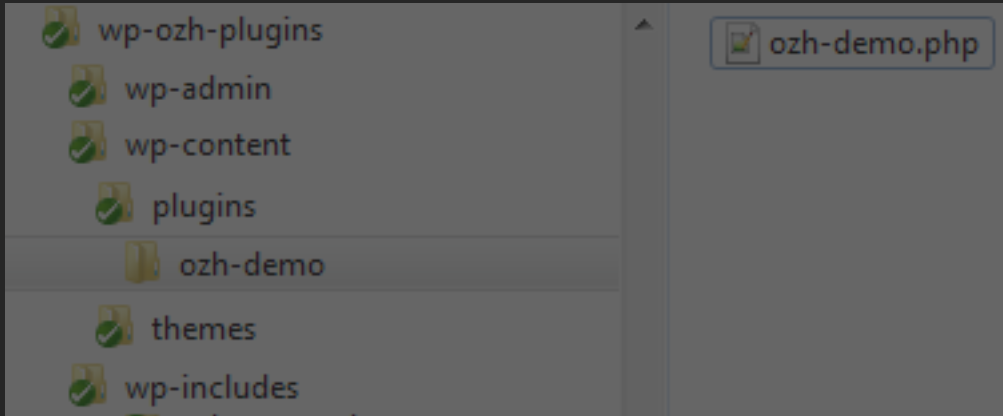
...

Créer les fichiers de test



plugins/my-cool-plugin
et my-cool-plugin.php

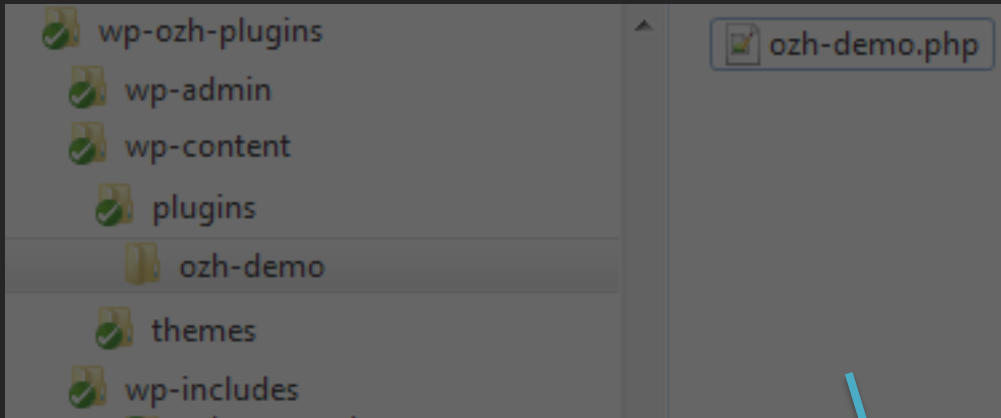
Créer les fichiers de test



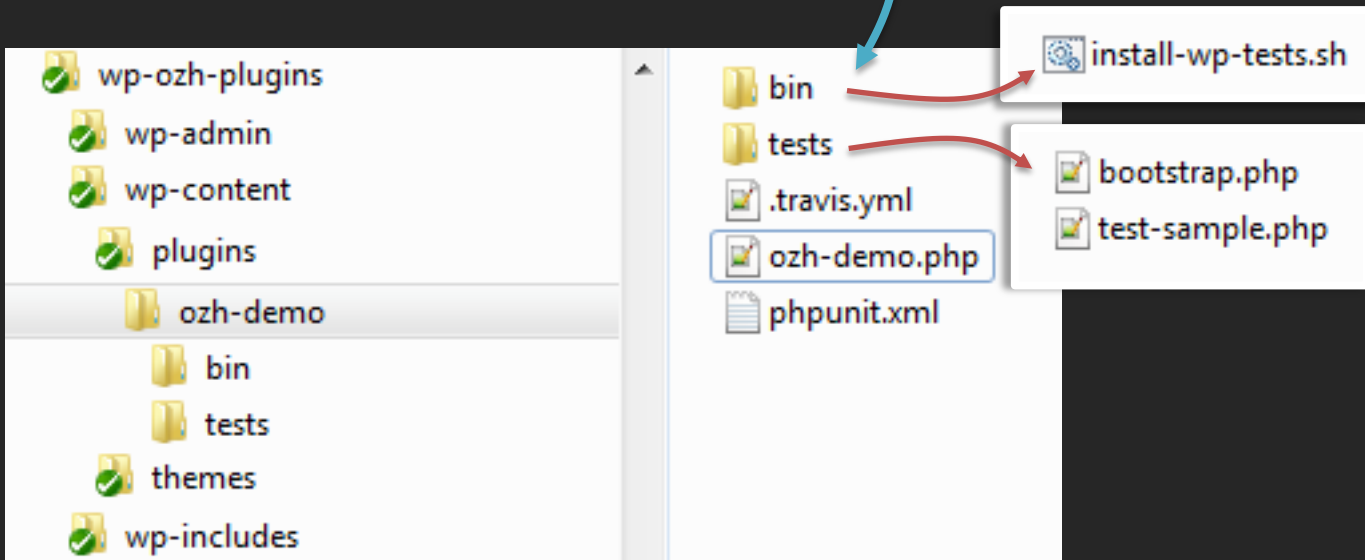
```
$ wp scaffold plugin-tests ozh-demo
```

```
Success: Created test files.
```

\$> Créer les fichiers de test



```
$ wp scaffold plugin-tests ozh-demo  
Success: Created test files.
```



.travis.yml

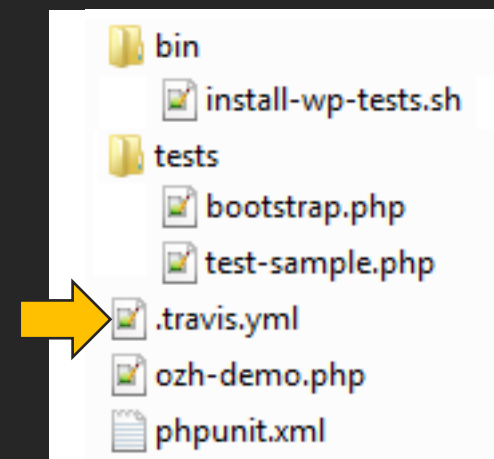
```
language: php
```

```
php:  
  - 5.3  
  - 5.4
```

```
env:  
  - WP_VERSION=latest WP_MULTISITE=0  
  - WP_VERSION=latest WP_MULTISITE=1  
  - WP_VERSION=3.8 WP_MULTISITE=0  
  - WP_VERSION=3.8 WP_MULTISITE=1
```

```
before_script:  
  - bash bin/install-wp-tests.sh wordpress_test root '' localhost $WP_VERSION
```

```
script: phpunit
```



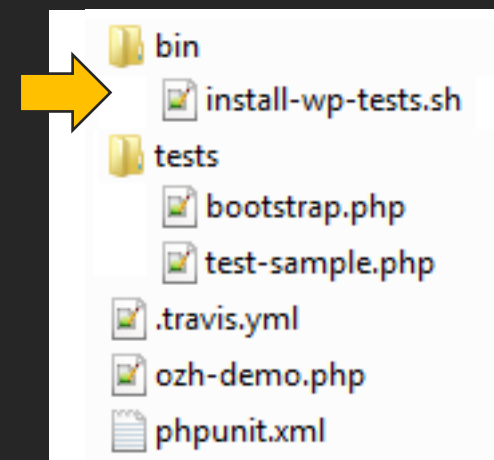
install-wp-tests.sh

```
#!/usr/bin/env bash
```

```
install_wp() {    # dans $WP_CORE_DIR
    wget https://wordpress.org/wordpress_version_XX.tar.gz
    wget https://raw.githubusercontent.com/markoheijnen/wp-mysqli/master/db.php
}

install_test_suite() {    # dans $WP_TESTS_DIR
    svn https://develop.svn.wordpress.org/trunk/tests/phpunit/includes/
    wget https://develop.svn.wordpress.org/trunk/wp-tests-config-sample.php
}

install_db() {
    mysqladmin create $DB_NAME
}
```



Installer la « WP Test Suite »

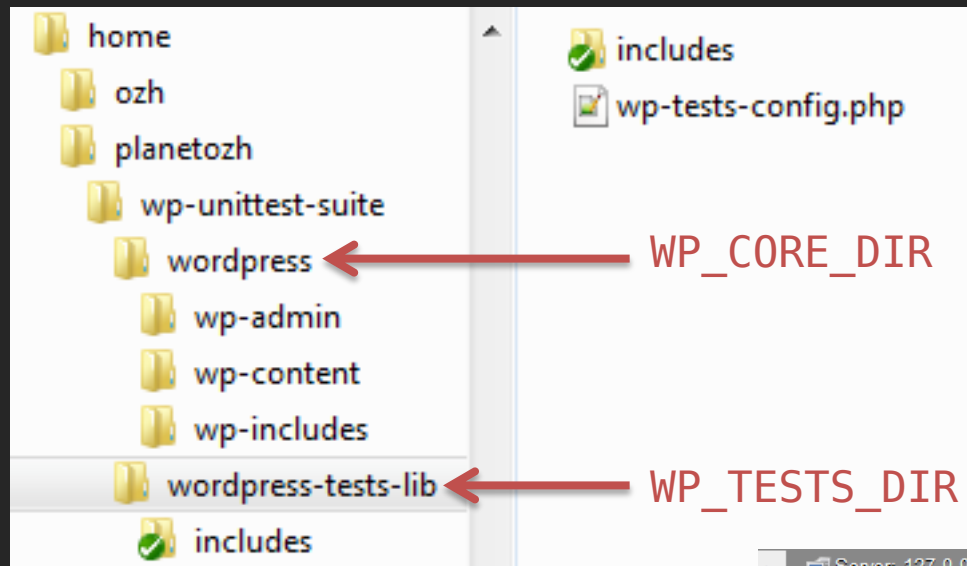
```
$ bash bin/install-wp-tests.sh wordpress_test root '' localhost latest
```

Installer la « WP Test Suite »

```
$ bash bin/install-wp-tests.sh wordpress_test root '' localhost latest
+ install_wp
+ mkdir -p C:/Users/0zh/AppData/Local/Temp/wordpress
+ '[' latest == latest ']'
+ local ARCHIVE_NAME=latest
+ wget -nv -O /tmp/wordpress.tar.gz https://wordpress.org/latest.tar.gz
+ /bin/wget --no-check-certificate -nv -O /tmp/wordpress.tar.gz
https://wordpress.org/latest.tar.gz
WARNING: cannot verify wordpress.org's certificate, issued by
`/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com,
Inc./OU=http://certs.godaddy.com/repository//CN=Go Daddy Secure Certificate
Authority - G2':
  Unable to locally verify the issuer's authority.
WARNING: certificate common name `*.wordpress.org' doesn't match requested host
name `wordpress.org'.
2015-01-02 21:46:40 URL:https://wordpress.org/latest.tar.gz [6183711/6183711] ->
C:/Users/0zh/AppData/Local/Temp/wordpress .....
...
```

et patati et patata pendant 3 minutes

\$> Installer la « WP Test Suite »

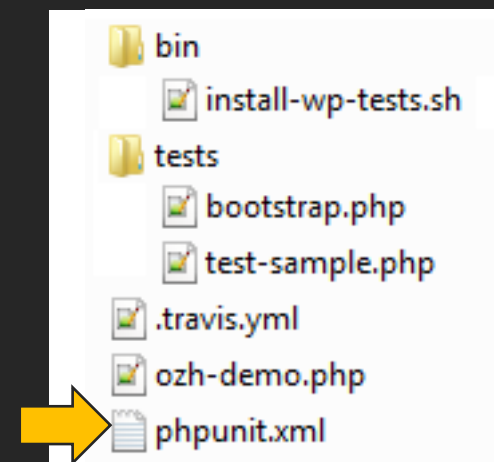


Server: 127.0.0.1 » Database: wordpress_test

Table	Action	Rows
☐ wptests_commentmeta	Browse Structure Search Insert Empty Drop	~0
☐ wptests_comments	Browse Structure Search Insert Empty Drop	~0
☐ wptests_links	Browse Structure Search Insert Empty Drop	~0
☐ wptests_options	Browse Structure Search Insert Empty Drop	~96
☐ wptests_postmeta	Browse Structure Search Insert Empty Drop	~0
☐ wptests_posts	Browse Structure Search Insert Empty Drop	~0
☐ wptests_terms	Browse Structure Search Insert Empty Drop	~1
☐ wptests_term_relationships	Browse Structure Search Insert Empty Drop	~0
☐ wptests_term_taxonomy	Browse Structure Search Insert Empty Drop	~1
☐ wptests_usermeta	Browse Structure Search Insert Empty Drop	~13
☐ wptests_users	Browse Structure Search Insert Empty Drop	~1
11 tables	Sum	112

phpunit.xml

```
<phpunit
  bootstrap="tests/bootstrap.php"
  backupGlobals="false"
  colors="true"
  convertErrorsToExceptions="true"
  convertNoticesToExceptions="true"
  convertWarningsToExceptions="true"
>
<testsuites>
  <testsuite>
    <directory prefix="test-" suffix=".php">./tests/</directory>
  </testsuite>
</testsuites>
</phpunit>
```

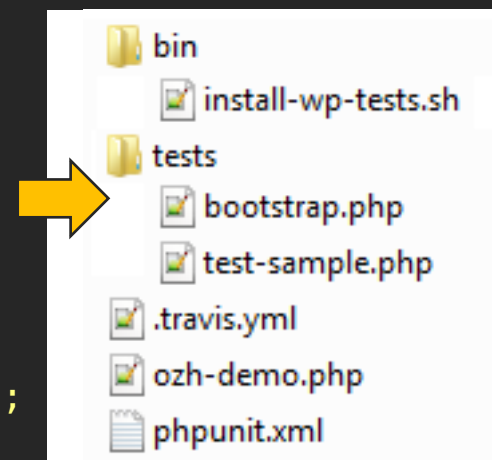


1) phpunit.xml

tests/bootstrap.php

```
<?php
```

```
$_tests_dir = getenv('WP_TESTS_DIR');  
if ( !$_tests_dir ) $_tests_dir = '/tmp/wordpress-tests-lib';  
  
require_once $_tests_dir . '/includes/functions.php';  
  
function _manually_load_plugin() {  
    require dirname( __FILE__ ) . '/../ozh-demo.php';  
}  
tests_add_filter( 'muplugins_loaded', '_manually_load_plugin' );  
  
require $_tests_dir . '/includes/bootstrap.php';
```

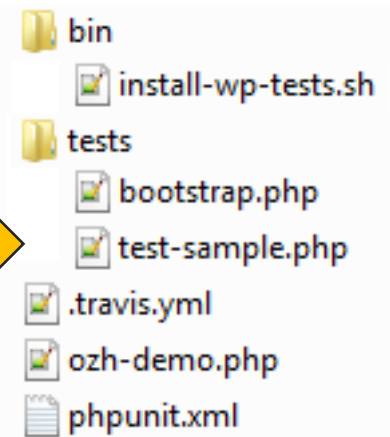


1) phpunit.xml → 2) bootstrap.php

tests/test-sample.php

```
<?php
```

```
class SampleTest extends WP_UnitTestCase {  
  
    function testSample() {  
        // replace this with some actual testing code  
        $this->assertTrue( true );  
    }  
}
```



1) **phpunit.xml** → 2) **bootstrap.php** → 3) **test-*.php**

\$> Test micro ? 1, 2 1, 2...

```
$ phpunit
```

Test micro ? 1, 2 1, 2...

```
$ phpunit
```

```
Installing...
```

```
Running as single site... To run multisite, use -c tests/phpunit/multisite.xml
```

```
Not running ajax tests. To execute these, use --group ajax.
```

```
Not running ms-files tests. To execute these, use --group ms-files.
```

```
Not running external-http tests. To execute these, use --group external-http.
```

```
PHPUnit 4.4.1 by Sebastian Bergmann.
```

```
Configuration read from D:\home\ozh\wp-ozh-plugins\wp-content\plugins\ozh-  
demo\phpunit.xml
```

```
.
```

```
Time: 6.1 seconds, Memory: 16.75Mb
```

```
OK (1 test, 1 assertion)
```

\$> Test micro ? 1, 2 1, 2...

```
$ phpunit
```

```
Installing...
```

```
Running as single site... To run multisite tests/phpunit/multisite.xml
```

```
Not running ajax tests. To execute tests/phpunit/ajax.
```

```
Not running ms-files tests/phpunit/group ms-files.
```

```
Not running external-http tests/phpunit/group external-http.
```

```
PHPUnit 4.4.1 by Sebastian Bergmann
```

```
Configuration read from  
demo/phpunit.xml
```

```
Time: 6.1 seconds, Memory: 10.5 MB
```

```
OK (1 test, 1 assertion)
```



Anatomie

d'un

Unit Test

tests/test-that-feature.php

```
<?php
```

```
class Particular_Feature_Test extends WP_UnitTestCase {
```

```
    function test_sub_feature() {
```

```
    }  
}
```

tests/test-that-feature.php

```
<?php
```

```
class Particular_Feature_Test extends WP_UnitTestCase {  
  
    function test_sub_feature() {  
        // REUNIR LES INGREDIENTS : contexte et dépendances  
  
        // CUISINER : déclencher les actions qu'on veut tester  
  
        // GOUTER : vérifier que c'est aussi bon que prévu  
  
    }  
}
```

tests/test-that-feature.php

```
<?php
```

```
class Particular_Feature_Test extends WP_UnitTestCase {

    function test_sub_feature() {
        // REUNIR LES INGREDIENTS : contexte et dépendances
        $needed_var    = 'something';
        $class_to_test = new Class_To_Test();

        // CUISINER : déclencher les actions qu'on veut tester

        // GOUTER : vérifier que c'est aussi bon que prévu

    }
}
```

tests/test-that-feature.php

```
<?php
```

```
class Particular_Feature_Test extends WP_UnitTestCase {

    function test_sub_feature() {
        // REUNIR LES INGREDIENTS : contexte et dépendances
        $needed_var = 'something';
        $class_to_test = new Class_To_Test();

        // CUISINER : déclencher les actions qu'on veut tester
        $stuff = $class_to_test->feature( $needed_var );
        $thing = do_something_with( $stuff );

        // GOUTER : vérifier que c'est aussi bon que prévu

    }
}
```

tests/test-that-feature.php

```
<?php
```

```
class Particular_Feature_Test extends WP_UnitTestCase {

    function test_sub_feature() {
        // REUNIR LES INGREDIENTS : contexte et dépendances
        $needed_var      = 'something';
        $class_to_test = new Class_To_Test();

        // CUISINER : déclencher les actions qu'on veut tester
        $stuff = $class_to_test->feature( $needed_var );
        $thing = do_something_with( $stuff );

        // GOUTER : vérifier que c'est aussi bon que prévu
        $this->assertEquals( 'résultat attendu', $thing );
        $this->assertEmpty( $class_to_test->some_var );
        $this->assertArrayHasKey( $stuff );
    }
}
```

PHPUnit : assertions

```
assertClassHasAttribute()  
assertClassHasStaticAttribute()  
assertContains()  
assertContainsOnly()  
assertContainsOnlyInstancesOf()  
assertCount()  
assertEmpty()  
assertEqualXMLStructure()  
assertEquals()  
assertFalse()  
assertFileEquals()  
assertFileExists()  
assertGreaterThan()  
assertGreaterThanOrEqual()  
assertInstanceOf()  
assertInternalType()  
assertJsonFileEqualsJsonFile()  
assertJsonStringEqualsJsonFile()
```

```
assertJsonStringEqualsJsonString()  
assertLessThan()  
assertLessThanOrEqual()  
assertNull()  
assertObjectHasAttribute()  
assertRegExp()  
assertStringMatchesFormat()  
assertStringMatchesFormatFile()  
assertSame()  
assertStringEndsWith()  
assertStringEqualsFile()  
assertStringStartsWith()  
assertThat()  
assertTrue()  
assertXmlFileEqualsXmlFile()  
assertXmlStringEqualsXmlFile()  
assertXmlStringEqualsXmlString()
```

Principe : `assertQuelqueChose()` et `assertNotQuelqueChose()`

Exemple : `assertEmpty()` et `assertNotEmpty()`

<https://phpunit.de/manual/current/en/appendixes.assertions.html>

❓ PHPUnit : setUp() & tearDown()

```
class Some_Test extends PHPUnit_Framework_TestCase {
```

```
    function test_quelquechose() {  
        ...  
    }
```

```
    function test_autre_chose() {  
        ...  
    }
```

```
}
```

❓ PHPUnit : setUp() & tearDown()

```
class Some_Test extends PHPUnit_Framework_TestCase {
```

```
    function setUp() {  
        do_things();  
    }
```

```
    function tearDown() {  
        undo_the_things();  
    }
```

```
    function test_quelquechose() {  
        ...  
    }
```

```
    function test_autre_chose() {  
        ...  
    }
```

```
}
```

PHPUnit : setUp() & tearDown()

```
class Some_Test extends PHPUnit_Framework_TestCase {

    public static function setUpBeforeClass() {
        connect_to_DB();
    }

    function setUp() {
        do_things();
    }

    function tearDown() {
        undo_the_things();
    }

    function test_quelquechose() {
        ...
    }

    function test_autre_chose() {
        ...
    }

}
```

PHPUnit : debug tip

```
// bootstrap.php :
```

```
function ut_var_dump( $what ) {  
    ob_start();  
    var_dump( $what );  
    $display = ob_get_contents();  
    ob_end_clean();  
    fwrite( STDERR, $display );  
}
```

```
// test :
```

```
class My_Cool_Test extends WP_UnitTestCase {  
  
    function test_something() {  
        ...  
        ut_var_dump( $wp_something );  
        ...  
    }  
}
```

OK... Et
pour les
plugins ?

Le cobaye !

<https://github.com/ozh/plugin-unit-test-demo>

Fork me on GitHub

Test me.



ozh-demo.php

```
<?php
```

```
/*  
Plugin Name: Unit Testing Demo Plugin  
*/
```

```
class Ozh_Demo_Plugin {  
    function __construct() { }  
    ↳ function init_plugin() { }  
        ↳ function add_shortcode( $atts ) { }  
        ↳ function add_meta_if_title( $post_id ) { }  
            ↳ function get_song_title(){ }  
                ↳ function update_meta() { }  
    function activate() { }  
    ↳ function create_page() { }  
    ↳ function create_tables() { }  
}  
  
new Ozh_Demo_Plugin;  
register_activation_hook( __FILE__,  
    array( 'Ozh_Demo_Plugin', 'activate' ) );
```

__construct()

```
<?php
/*
Plugin Name: Unit Testing Demo Plugin
Plugin URI: https://github.com/ozh/plugin-unit-test-demo
Description: Demo plugin. For unit tests and the lulz
Author: Ozh
Version: 0.1
Author URI: http://ozh.org
*/

class Ozh_Demo_Plugin {

    public $post_id;
    public $song_title;

    /**
     * Class constructor
     */
    function __construct() {
        add_action( 'init', array( $this, 'init_plugin' ) );
        do_action( 'ozh_demo_plugin_loaded' );
    }

    function init_plugin() {

    }

}
```

Ozh_Demo_Construct_Test{ }

```
/**
 * Test the constructor function
 */
class Ozh_Demo_Construct_Test extends WP_UnitTestCase {

    public $demo_plugin;

    function setUp() {
        parent::setUp();
        $this->demo_plugin = new Ozh_Demo_Plugin;
    }

    function test_construct() {
        $this->assertEquals(
            10,
            has_action( 'init', array( $this->demo_plugin, 'init_plugin' ) )
        );

        $this->assertGreaterThan( 0, did_action( 'ozh_demo_plugin_loaded' ) );
    }

}
```

Ozh_Demo_Construct_Test{ }

```
/**
 * Test the constructor function
 */
class Ozh_Demo_Construct_Test extends WP_UnitTestCase {

    public $demo_plugin;

    function setUp() {
        parent::setUp();
        $this->demo_plugin = new Ozh_Demo_Plugin;
    }

    function test_construct() {
        $this->assertEquals(
            10,
            has_action( 'init', array( $this->demo_plugin, 'init_plugin' ) )
        );

        $this->assertGreaterThan( 0, did_action( 'ozh_demo_plugin_loaded' ) );
    }

}
```



init_plugin()

```
<?php

class Ozh_Demo_Plugin {

    ...

    /**
     * Plugin init: starts all that's needed
     */
    function init_plugin() {
        add_action( 'save_post', array( $this, 'add_meta_if_title' ) );
        add_shortcode( 'omglol', array( $this, 'add_short_code' ) );
    }

    function add_meta_if_title( $post_id ) {
        // do stuff
    }

    function add_short_code( $atts ) {
        return "OMG c'est trop LOL !";
    }
}
```

Ozh_Demo_Init_Test{ }

```
<?php
/**
 * Test the init function
 */
class Ozh_Demo_Init_Test extends WP_UnitTestCase {

    public $demo_plugin;
    function setUp() {

    }

    function test_init_plugin() {
        // Simulate WordPress init
        do_action( 'init' );

        $this->assertEquals(10, has_action('save_post',
            array($this->demo_plugin, 'add_meta_if_title' ) ) );

        global $shortcode_tags;
        $this->assertArrayHasKey( 'omglol', $shortcode_tags );
    }

}
```

Ozh_Demo_Init_Test{ }

```
<?php
/**
 * Test the init function
 */
class Ozh_Demo_Init_Test extends WP_UnitTestCase {

    public $demo_plugin;
    function setUp() {

    }

    function test_init_plugin() {
        // Simulate WordPress init
        do_action( 'init' );

        $this->assertEquals(10, has_action(
            array($this->demo_plugin, 'add_meta_if_title' ) ) );

        global $shortcode_tags;
        $this->assertArrayHasKey( 'omglol', $shortcode_tags );
    }
}
```



**Dans vos tests de plugin,
ne testez pas les API WordPress !**
Testez la manifestation la plus simple
de la bonne utilisation de l'API.

activate()

```
<?php
```

```
class Ozh_Demo_Plugin {
```

```
    ...
```

```
    /**
```

```
     * Things to do when the plugin is activated
```

```
     */
```

```
    function activate() {  
        $this->create_tables();  
        $this->create_page();  
    }
```

```
    function create_tables() {...}
```

```
    function create_page() {...}
```

```
}
```

```
new Ozh_Demo_Plugin;
```

```
register_activation_hook( __FILE__, array( 'Ozh_Demo_Plugin', 'activate' ) );
```

Ozh_Demo_Activate_Test{ }

```
<?php
/**
 * Test the activation function
 */
class Ozh_Demo_Activate_Test extends WP_UnitTestCase {

    function test_activate() {
        $demo_plugin = $this->getMockBuilder( 'Ozh_Demo_Plugin' )
            ->setMethods( array( 'create_tables', 'create_page' ) )
            ->getMock();

        $demo_plugin->expects( $this->once() )
            ->method( 'create_tables' );

        $demo_plugin->expects( $this->once() )
            ->method( 'create_page' );

        $demo_plugin->activate();
    }
}
```

Ozh_Demo_Activate_Test{ }

```
<?php
/**
 * Test the activation function
 */
class Ozh_Demo_Activate_Test extends WP_UnitTestCase {

    function test_activate() {
        $demo_plugin = $this->getMockBuilder( 'Ozh_De
            ->setMethods( array( 'create_tables', 'cr
            ->getMock();

        $demo_plugin->expects( $this->once()
            ->method( 'create_tables' );

        $demo_plugin->expects( $this->once() )
            ->method( 'create_page' );

        $demo_plugin->activate();
    }
}
```

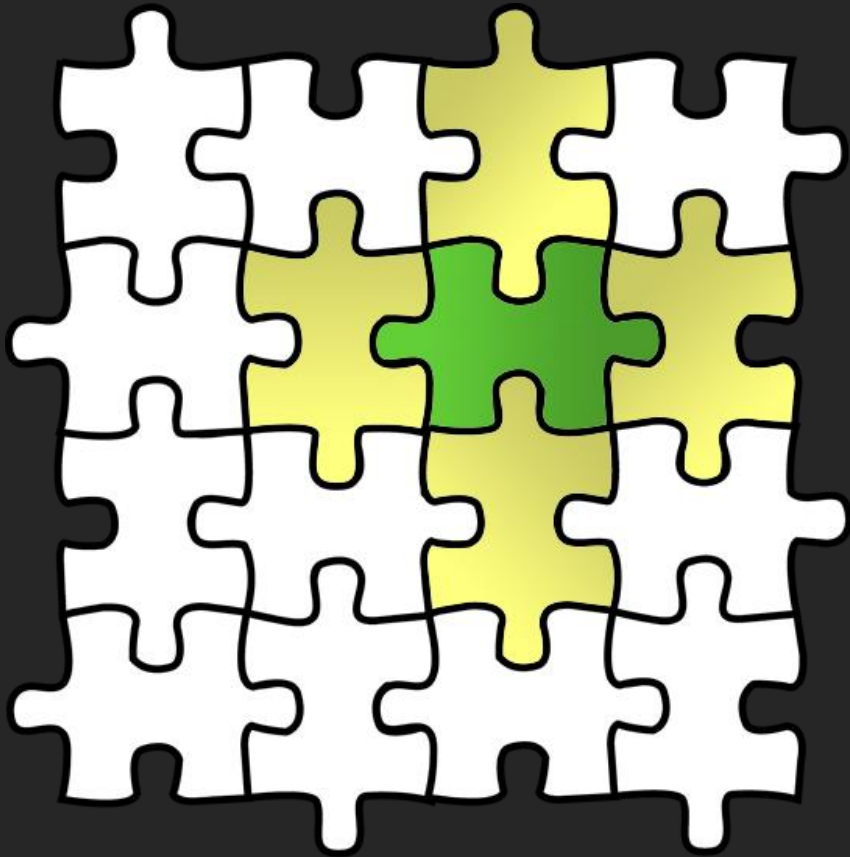


Les Mocks

Production

● my_feature()

● dépendances

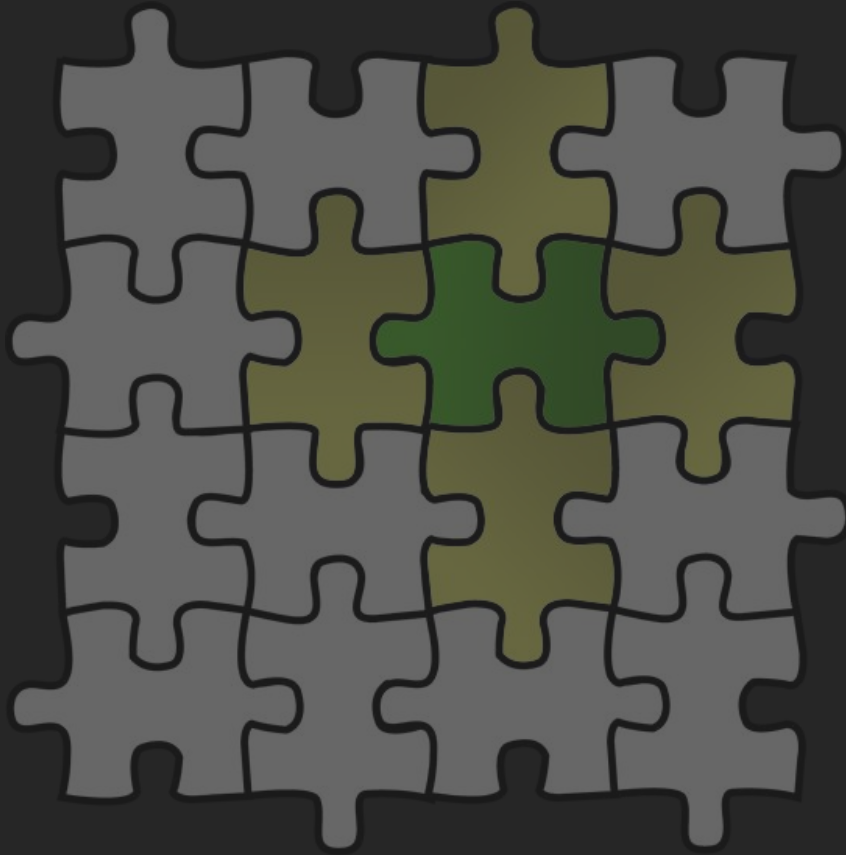


[?] Les Mocks

Production

● my_feature()

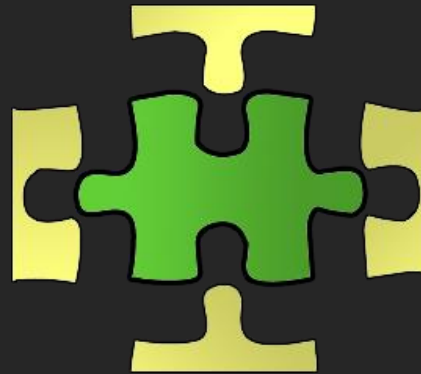
● dépendances



Tests

● my_feature()

● mocks



Mock:

objets **simulés** qui reproduisent le comportement d'objets réels de **manière contrôlée**

WP Mocks : Mock Image Editor

```
<?php
```

```
if (class_exists( 'WP_Image_Editor' ) ) :  
    class WP_Image_Editor_Mock extends WP_Image_Editor {  
        * * *  
        public function resize( $max_w, $max_h, $crop = false ) {  
        }  
        public function multi_resize( $sizes ) {  
        }  
        * * *  
    }  
endif;
```

WP Mocks : Mock Mailer

```
<?php
```

```
class Plugin_With_Mail_Test extends WP_UnitTestCase {

    function setUp(){
        parent::setUp();
        $_SERVER['SERVER_NAME'] = 'example.com'; // IMPORTANT
    }

    function test_something_that_sends_mail() {
        * * *
        $sent = wp_mail( 'ozh@ozh.org', 'OMG', 'WTF & KTHXBYE' );
        $this->assertTrue( $sent );
        * * *
    }

}
```

WP Mocks : Mock FS

WP_FileSystem :

```
$wp_filesystem->mkdir( $plugin_path . '/config/' );  
$wp_filesystem->put_contents( $filename, $content, FS_CHMOD_FILE);
```

WordPress classes :

- WP_Filesystem_Direct
- WP_Filesystem_FTPext
- WP_Filesystem_ftpsocket
- WP_Filesystem_SSH2

http://codex.wordpress.org/FileSystem_API

WordPress Tests classes :

- WP_Filesystem_MockFS

<https://develop.svn.wordpress.org/trunk/tests/phpunit/tests/filesystem/base.php>

create_tables()

```
class Ozh_Demo_Plugin {
    ...

    /**
     * Create a custom table
     */
    function create_tables() {
        global $wpdb;

        $table_name = $wpdb->prefix . "ozh_demo";
        $charset_collate = $wpdb->get_charset_collate();

        $sql = "CREATE TABLE $table_name (
            `id` mediumint(9) NOT NULL AUTO_INCREMENT,
            `time` datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
            `name` tinytext NOT NULL,
            UNIQUE KEY id (id)
        ) $charset_collate;";

        require_once( ABSPATH . 'wp-admin/includes/upgrade.php' );
        return dbDelta( $sql );
    }
    ...
}
```

Ozh_Demo_Table_Test{ }

```
/**
 * Test the DB function
 */
class Ozh_Demo_DB_Test extends WP_UnitTestCase {

    function test_create_DB() {
        global $wpdb;

        $demo_plugin = new Ozh_Demo_Plugin;
        $created = $demo_plugin->create_tables();

        // ut_var_dump( current( $created ) );
        // string(30) "Created table wptests_ozh_demo"

        $this->assertStringStartsWith( "Created table", current( $created ) );
    }
}
```

create_page()

```
class Ozh_Demo_Plugin {
    ...

    /**
     * Create a custom page
     */
    function create_page() {
        global $user_ID;
        $page = array(
            'post_type'      => 'page',
            'post_title'     => 'Code is Poterie',
            'post_content'   => 'Opération Template du Dessert',
            'post_status'    => 'publish',
            'post_author'    => $user_ID,
        );

        return wp_insert_post( $page );
    }
    ...
}
```

Ozh_Demo_Page_Create_Test{ }

```
<?php
/**
 * Test page creation
 */
class Ozh_Demo_Create_Page_Test extends WP_UnitTestCase {

    public $demo_plugin;

    public static function setUpBeforeClass() {
        global $wp_rewrite;
        $wp_rewrite->init();
        $wp_rewrite->set_permalink_structure( '/%year%/%postid%/' );
        $wp_rewrite->flush_rules();
    }

    public static function tearDownAfterClass() {
        global $wp_rewrite;
        $wp_rewrite->init();
    }

    function setUp(){
        parent::setUp();
        $this->demo_plugin = new Ozh_Demo_Plugin;
    }

    . . .
}
```

Ozh_Demo_Page_Create_Test{ }

```
<?php
/**
 * Test page creation
 */
class Ozh_Demo_Create_Page_Test extends WP_UnitTestCase {
    ""
    function test_create_page_one_way() {
        $this->go_to( '/code-is-poterie/' );
        $this->assertQueryTrue( 'is_404' );

        $this->demo_plugin->create_page();
        $this->go_to( '/code-is-poterie/' );
        $this->assertQueryTrue( 'is_page', 'is_singular' );
    }
}
```

update_meta()

```
class Ozh_Demo_Plugin {  
    ...  
  
    /**  
     * Update post meta data 'with_song' with song title  
     */  
    function update_meta() {  
        update_post_meta( $this->post_id, 'with_song', $this->song_title );  
    }  
    ...  
}
```

Ozh_Demo_Update_Meta_Test{ }

```
<?php
/**
 * Test the update meta function
 */
class Ozh_Demo_Update_Meta_Test extends WP_UnitTestCase {

    public $demo_plugin;
    function setUp() { }

    function test_update_meta() {
        $this->demo_plugin->post_id      = ???;
        $this->demo_plugin->song_title = ???;

        $this->assertEmpty( get_post_meta( $post_id, 'with_song' ) );
        $this->demo_plugin->update_meta();
        $this->assertNotEmpty( get_post_meta( $post_id, 'with_song' ) );
    }
}
```

Ozh_Demo_Update_Meta_Test{ }

```
<?php
/**
 * Test the update meta function
 */
class Ozh_Demo_Update_Meta_Test extends WP_UnitTestCase {

    public $demo_plugin;
    function setUp() { }

    function test_update_meta() {
        $this->demo_plugin->post_id      = $this->factory->post->create();
        $this->demo_plugin->song_title = rand_str( 32 );

        $this->assertEmpty( get_post_meta( $post_id, 'with_song' ) );
        $this->demo_plugin->update_meta();
        $this->assertNotEmpty( get_post_meta( $post_id, 'with_song' ) );
    }
}
```

L'objet factory

```
<?php
// Create posts
$page_id = $this->factory->post->create( array( 'post_title' => 'Test Page' ) );
$this->factory->post->create_many(5, array('post_date' => '2015-01-24 09:45:00'));

// Create users
$user = $this->factory->user->create();

// Comments, Terms, Categories, Tags, Attachments, Blogs, Networks
$this->factory->comment->create();
$this->factory->term->create();
$this->factory->category->create()
$this->factory->attachment->create_object('image.jpg', $post_id, $args);
$this->factory->blog->create();
$this->factory->network->create();

// https://core.trac.wordpress.org/browser/trunk/tests/phpunit/includes/factory.php
```

Ozh_Demo_Update_Meta_Test{ }

```
<?php
/**
 * Test the update meta function
 */
class Ozh_Demo_Update_Meta_Test extends WP_UnitTestCase {

    public $demo_plugin;

    function setUp() { }

    function test_update_meta() {
        $this->demo_plugin->post_id      = $this->factory->post->create();
        $this->demo_plugin->song_title = rand_str( 32 );

        $this->assertEmpty( get_post_meta( $post_id, 'with_song' );
        $this->demo_plugin->update_meta();
        $this->assertNotEmpty( get_post_meta( $post_id, 'with_song' );
    }
}
```



get_song_title()

```
class Ozh_Demo_Plugin {
    ...

    /**
     * Get current song title played on that cool web radio
     */
    function get_song_title(){
        $remote = 'http://www.radiometal.com/player/song_infos_player.txt';
        $body    = wp_remote_retrieve_body( wp_remote_get( $remote ) );
        /* sample return:
        <div id='playerartistname'>Metallica</div>
        <div id='playersongname'>Master of Puppets</div>
        */
        preg_match_all( "!<div id='[^']*'>([<]*)</div>!", $body, $matches );
        // $matches[1][0]: artist - $matches[1][1]: song name

        if( $matches[1] ) {
            return $matches[1][0] . ' - ' . $matches[1][1];
        } else {
            return false;
        }
    }
}
```

Ozh_Demo_Remote_Test{ }

```
<?php
/**
 * Test the remote request functions
 */
class Ozh_Demo_Remote_Test extends WP_UnitTestCase {

    public $demo_plugin; function setUp(){ }

    function return_valid_html() {
        $html = "BLAH <div id='playerartistname'>Slayer</div>
                <div id='playersongname'>Raining Blood</div> BLAH";
        return array( 'body' => $html );
    }

    function test_remote_request_valid_HTML() {
        add_filter( 'pre_http_request', array( $this, 'return_valid_html' ) );

        $this->assertSame(
            'Slayer - Raining Blood',
            $this->demo_plugin->get_song_title() );
    }
}
```

Ozh_Demo_Remote_Test{ }

```
<?php
/**
 * Test the remote request functions
 */
class Ozh_Demo_Remote_Test extends WP_UnitTestCase {

    function return_crap_html() {
        return array( 'body' => "Error 500 - come back later" );
    }

    function return_no_html() {
        return array( );
    }

    function test_remote_request_invalid_HTML() {
        add_filter( 'pre_http_request', array( $this, 'return_crap_html' ) );
        $this->assertFalse( $this->demo_plugin->get_song_title() );
        add_filter( 'pre_http_request', array( $this, 'return_no_html' ) );
        $this->assertFalse( $this->demo_plugin->get_song_title() );
    }
}
```

Ozh_Demo_Remote_Test{ }

```
<?php
```

```
/**
```

```
 * Test the remote request functions
```

```
 */
```

```
class Ozh_Demo_Remote_Test extends WP_UnitTestCase {
```

```
    function return_crap_html() {
```

```
        return array( 'body' => "Error 500 - come back later" );
```

```
    }
```

```
    function return_no_html() {
```

```
        return array( );
```

```
    }
```

```
    function return_valid_html() {
```

```
    }
```



Trucs divers
à **retenir** ou à **faire**

Testez aussi le trivial !

```
function do_something( )
```

→ Janvier 2015 :

2 variables
+ 1 test

= 1 valeur attendue

→ Janvier 2018 :

6 variables
+ 3 tests
+ 5 fonctions internes
+ 2 librairies externes

= la même valeur attendue !

Testez vos fonctions de thèmes !

```
<?php
require_once '/path/to/your-theme/functions.php';
function _manually_load_theme() {
    switch_theme( 'your-theme' );
}
tests_add_filter( 'muplugins_loaded', '_manually_load_theme' );
```

❓ Testable ou dé-testable ?

```
class Ozh_Demo_Plugin {  
  
    function add_meta_if_title( ) {  
        if ( 'DOING_AUTOSAVE' ) ...  
        if ( ! current_user_can( ) ) ...  
  
        $this->get_song_title();  
  
        if( $this->song_title ) {  
            $this->update_meta();  
        }  
    }  
  
    function update_meta() { ... }  
  
    function get_song_title(){ ... }  
  
}
```

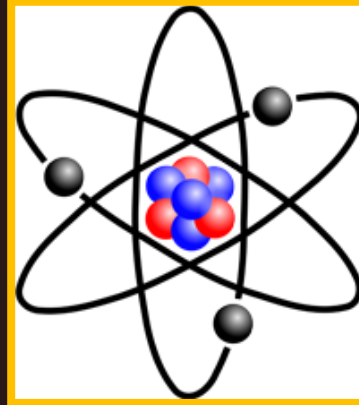
❓ Testable ou dé-testable ?

```
class Ozh_Demo_Plugin {  
  
    function add_meta_if_title( ) {  
        if ( 'DOING_AUTOSAVE' ) ...  
        if ( ! current_user_can( ) ) ...  
  
        $this->get_song_title();  
  
        if( $this->song_title ) {  
            $this->update_meta();  
        }  
    }  
  
    function update_meta() { ... }  
  
    function get_song_title(){ ... }  
  
}
```

```
class Ozh_Demo_Plugin {  
  
    function add_meta_if_title() {  
        if ( 'DOING_AUTOSAVE' ) ...  
        if ( ! current_user_can( ) ) ...  
  
        $remote = ...;  
        $body = wp_remote_retrieve_body( );  
        preg_match_all( ... ) ;  
        $this->song_title = ...;  
  
        if( $this->song_title ) {  
            update_post_meta( ... );  
        }  
    }  
  
}
```

Testable = atomique

```
class Ozh_Demo_Plugin {  
  
    function add_meta_if_title( ) {  
        if ( 'DOING_AUTOSAVE' ) ...  
        if ( ! current_user_can( ) ) ...  
  
        $this->get_song_title();  
  
        if( $this->song_title ) {  
            $this->update_meta();  
        }  
    }  
  
    function update_meta() { ... }  
  
    function get_song_title(){ ... }  
  
}
```



```
class Ozh_Demo_Plugin {  
  
    function add_meta_if_title() {  
        if ( 'DOING_AUTOSAVE' ) ...  
        if ( ! current_user_can( ) ) ...  
  
        $remote = ...;  
        $body = wp_remote_retrieve_body( );  
        match_all( ... ) ;  
        ->song_title = ...;  
  
        if( $this->song_title ) {  
            update_post_meta( ... );  
        }  
    }  
  
}
```

❓ Pas de tests dans vos archive.zip ?

Dans .gitattributes

Exclude certain files or directories when generating an archive

.travis.yml export-ignore

composer.* export-ignore

.git* export-ignore

bin export-ignore

tests export-ignore

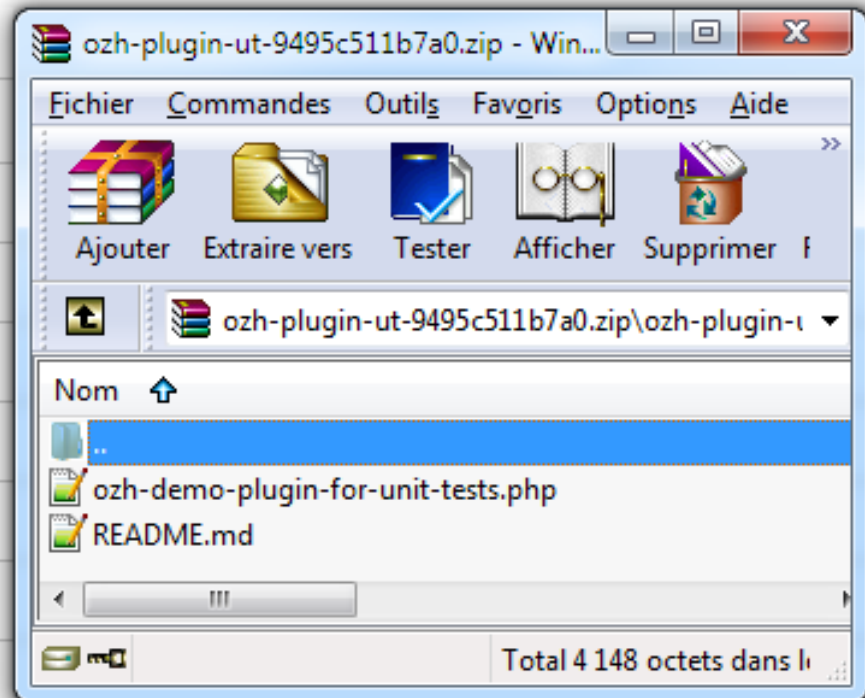
phpunit.xml export-ignore

[?] Pas de tests dans vos archive.zip ?

Dans .gitattributes

```
# Exclure les fichiers de tests du processus de compression
.travis:
compos:
.git*:
bin ex:
tests:
phpun:

bin
tests
.gitattributes
.gitignore
.travis.yml
README.md
ozh-demo-plugin-for-unit-tests.php
phpunit-coverage.xml
phpunit.xml
```



Etendez l'objet WP_UnitTestCase

1) Bootstrap.php :

```
require_once dirname( __FILE__ ) . '/My_Custom_Assertions.php';
```

🔗 Etendez l'objet WP_UnitTestCase

1) Bootstrap.php :

```
require_once dirname( __FILE__ ) . '/My_Custom_Assertions.php';
```

2) My_Custom_Assertions.php :

```
class MyPlugin_UnitTestCase extends WP_UnitTestCase {  
    /**  
     * Assert something specific to my plugin  
     */  
    function assertSpecificThing(...) {  
        ...  
        $this->assertFalse(...)  
    }  
}
```

Etendez l'objet WP_UnitTestCase

1) Bootstrap.php :

```
require_once dirname( __FILE__ ) . '/My_Custom_Assertions.php';
```

2) My_Custom_Assertions.php :

```
class MyPlugin_UnitTestCase extends WP_UnitTestCase {  
    /**  
     * Assert something specific to my plugin  
     */  
    function assertSpecificThing(...) {  
        ...  
        $this->assertFalse(...)  
    }  
}
```

3) tests/test-something.php :

```
class Something_Test extends MyPlugin_UnitTestCase {  
    ...  
}
```

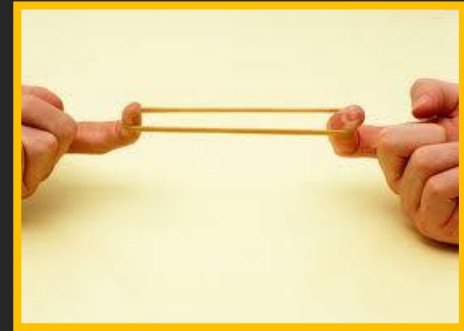
Etendez l'objet WP_UnitTestCase

1) Bootstrap.php :

```
require_once dirname( __FILE__ ) . '/My_Custom_Assertions.php';
```

2) My_Custom_Assertions.php :

```
class MyPlugin_UnitTestCase extends WP_UnitTestCase {  
    /**  
     * Assert something specific to my plugin  
     */  
    function assertSpecificThing(...) {  
        ...  
        $this->assertFalse(...)  
    }  
}
```



3) tests/test-something.php :

```
class Something_Test extends MyPlugin_UnitTestCase {  
    ...  
}
```

Exemples :

`assertArchiveEquals()`, `assertArchiveContains()`, `assertArchiveFileCount()`

<https://github.com/humanmade/backupwordpress/blob/master/tests/class-wp-test-hm-backup-testcase.php>

Vérifiez votre Code Coverage

phpunit-coverage.xml

```
<phpunit
  bootstrap="tests/bootstrap.php"
  ...
>

  <filter>
    <blacklist>
      <directory>/home/planetozh/wp-unittest-suite</directory>
      <directory>./tests</directory>
    </blacklist>
  </filter>

  <logging>
    <log type="coverage-html" target="coverage" title="PHPUnit"
      charset="UTF-8" yui="true" highlight="true"
      lowUpperBound="50" highLowerBound="90"/>
  </logging>

</phpunit>
```

Générer le rapport de code coverage

```
$ phpunit -c phpunit-coverage.xml
```

[?] Vérifiez votre Code Coverage

	Code Coverage									
	Classes and Traits			Functions and Methods				Lines		
Total		0.00%	0 / 1		88.89%	8 / 9	CRAP		86.67%	39 / 45
Ozh_Demo_Plugin		0.00%	0 / 1		90.00%	9 / 10	15.53		86.67%	39 / 45
__construct()					100.00%	1 / 1	1		100.00%	3 / 3
init_plugin()					100.00%	1 / 1	1		100.00%	3 / 3
add_meta_if_title(\$post_id)					0.00%	0 / 1	10.40		40.00%	4 / 10
add_short_code(\$atts)					100.00%	1 / 1	1			
update_meta()					100.00%	1 / 1	1		100.00%	2 / 2
save_into_flat_file()										
get_song_title()										
create_page()										
create_tables()										
activate()										

</logging>

</phpunit>



Générer le rapport de code coverage

\$ phpunit -c phpunit-coverage.xml

```

40 function add_meta_if_title( $post_id ) {
41
42     if ( defined( 'DOING_AUTOSAVE' ) && DOING_AUTOSAVE ) {
43         return;
44     }
45
46     if ( ! current_user_can( 'edit_post', $post_id ) ) {
47         return;
48     }
49
50     $this->get_song_title();
51
52     if( $this->song_title ) {
53         $this->update_meta();
54         $this->save_into_flat_file();
55     }
56 }
```

🔍 Testez vos interfaces !

Selenium - <http://www.seleniumhq.org/>

```
<?php
// Selenium - automates browsers.
class ConnectTest extends PHPUnit_Extensions_Selenium2TestCase {

    protected function setUp() {
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("https://mylocal.dev/");
    }

    public function testUserCanLogIn() {
        $this->open("/");
        $this->click("css=img[alt='Twitter']");
        $this->waitForPageToLoad("30000");
        $this->assertContains( 'dashboard', $this->title() );
    }
}
```

Qunit – <http://qunitjs.com/>

```
<script>
test("Array should contain four items", function () {
    equal( myArray.length, 4, "Pass! - array contains four items");
});
</script>
```

[?] Testez vos interfaces !

Selenium - <http://www.seleniumhq.org/>

```
<?php
// Selenium - automates browsers.
class ConnectTest extends PHPUnit_Extensions_Selenium2TestCase {

    protected function setUp() {
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("https://mylocal.dev/");
    }

    public function testUserCanLogIn() {
        $this->open("/");
        $this->click("css=img[alt='Twitter']");
        $this->waitForPageToLoad("30000");
        $this->assertContains( 'dashboard', $this->title() )
    }
}
```



Qunit – <http://qunitjs.com/>

```
<script>
test("Array should contain four items", function () {
    equal( myArray.length, 4, "Pass! - array contains four items");
});
</script>
```

Devoirs à la maison

L'excellente **documentation** de PHPUnit

→ <https://phpunit.de/documentation.html>

Des **tutoriaux** supplémentaires

→ <http://voceplatforms.com/?s=unit+testing+wordpress>

→ <https://pippinsplugins.com/tag/unit-tests/>

→ <http://planetozh.com/blog/> ??

Les **Unit Tests** de WordPress

→ <http://develop.svn.wordpress.org/trunk/>

Des **plugins** avec tests

→ <https://github.com/easydigitaldownloads/Easy-Digital-Downloads>

→ <https://github.com/humanmade/backupwordpress>

Et **coder** des tests !

→ Ecrire **10** tests la semaine prochaine

→ Ecrire **100** tests le mois suivant ?

→ 1 **bug** corrigé = 1 **nouveau test**

Merci !

  ? → <http://ozh.in/feedback>

Questions ?