

Projet de développement tests unitaires en PHP

Philippe Collet

Licence 3 Informatique

2012-2013

❑ Framework de test unitaire pour PHP (4 et 5)

- Créé par Marcus Baker

❑ Fonctionnalités

- Principes similaires à JUnit (et PHPUnit)
- Test passé coté client ou déployé sur le serveur
- Support pour des « objets fantaisie » (*mock objects*)
- Possibilité d'automatisation de test de non-régression d'applications web
 - ◆ Client HTTP « scriptable » qui analyse des pages web, simule des actions comme des clics sur les liens et soumet des formulaires

❑ Alternatives :

- PHPUnit

SimpleTest : fonctionnement de base (extrait du tutorial)

- ❑ Nous disposons d'une classe *classes/Log.php* qui fait... du log

- C'est la classe à tester

- ❑ Etape 1 : script de test (*tests/log_test.php*)

```
<?
php require_once('simpletest/autorun.php');
require_once('../classes/log.php');
class TestOfLogging extends UnitTestCase {
}
?>
```

- ❑ Répertoire *simpletest* (dépendant de l'installation de simpleTest)

- dans le dossier courant
- ou dans les dossiers pour fichiers inclus.

- ❑ Fichier "autorun.php"

- inclut les éléments de SimpleTest
- lance aussi les tests pour nous

SimpleTest : fonctionnement de base

- Supposons que la classe Log prenne le nom du fichier à écrire au sein du constructeur, et que nous avons un répertoire temporaire dans lequel placer ce fichier

```
<?
php require_once('simpletest/autorun.php');
require_once('../classes/log.php');
class TestOfLogging extends UnitTestCase {
    function testLogCreatesNewFileOnFirstMessage() {
        @unlink('/temp/test.log');
        $log = new Log('/temp/test.log');
        $this->assertFalse(file_exists('/temp/test.log'));
        $log->message('Should write this to a file');
        $this->assertTrue(file_exists('/temp/test.log'));
    }
}
?>
```

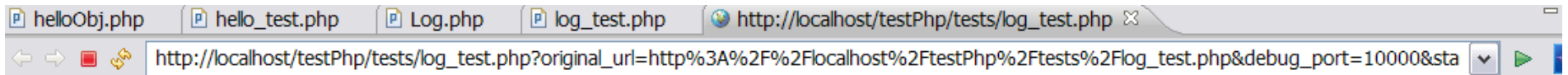
SimpleTest : exécution

❑ Au lancement du scénario de test :

- Identification des méthodes commençant par « test »
- En cas d'erreur ou d'échec, le framework affiche tout de suite le résultat

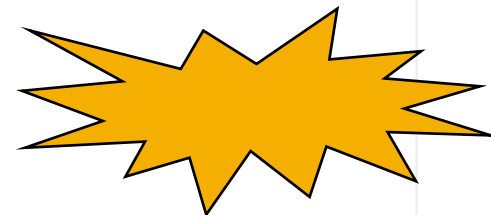
❑ Comment lancer ?

- Afficher la page dans le navigateur
- En local (si on a un interprète sur sa machine : station de travail)
- A distance (avec l'environnement www-mips de la faculté)



Warning: require_once(simpletest/autorun.php) [[function.require-once](#)]: failed to open stream: No such file or directory in C:\xampp\htdocs\testPhp\tests\log_test.php on line 2

Fatal error: require_once() [[function.require](#)]: Failed opening required 'simpletest/autorun.php' (include_path='.;C:\xampp\php\pear\') in C:\xampp\htdocs\testPhp\tests\log_test.php on line 2



SimpleTest : résultats

☐ A surveiller :

- Bonne inclusion (locale) des fichiers testés
- Bonne inclusion des fichiers de simpleTest : prendre le répertoire de simpletest dans le plugin Eclipse pour le recopier dans votre serveur de test (cf. « include_path » de votre php.ini, par exemple)

helloObj.php | hello_test.php | Log.php | log_test.php | http://localhost/testPhp/tests/log_test.php

http://localhost/testPhp/tests/log_test.php?original_url=http%3A%2F%2Flocalhost%2FtestPhp%2Ftests%2Flog_test.php&debug_port=10000&sta

log_test.php

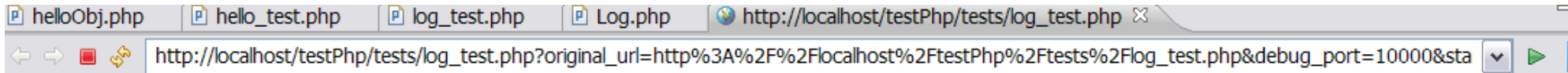
Fail: TestOfLogging -> testLogCreatesNewFileOnFirstMessage -> at [C:\xampp\htdocs\testPhp\tests\log_test.php line 10]

1/1 test cases complete: 1 passes, 1 fails and 0 exceptions.

☐ = Echec

☐ Sinon, sortie « verte » pour réussite des tests

SimpleTest : tests passés



log_test.php

1/1 test cases complete: 2 passes, 0 fails and 0 exceptions.

```
<?php
    class Log {
        var $_file_path;

        function Log($file_path) {
            $this->_file_path = $file_path;
        }

        function message($message) {
            $file = fopen($this->_file_path, 'a');
            fwrite($file, $message . "\n");
            fclose($file);
        }
    }
?>
```

Suite de tests

❑ Nouveau fichier : tests/all_tests.php

```
<?php
require_once('simpletest/autorun.php');

class AllTests extends TestSuite {
    function AllTests() {
        $this->TestSuite('All tests');
        $this->addFile('log_test.php');
    }
}
?>
```

❑ Autorun.php permet l'auto-exécution du script

❑ TestSuite doit chaîner son constructeur (limitation de simpletest)

Développement de l'exemple

- ❑ Supposons une classe `SessionPool` qui utilise le Log...

```
class SessionPool {  
    ...  
    function login($username) {  
        ...  
        $this->_log->message('User $username logged in.');        ...  
    }  
    ...  
}
```

Setup et Teardown (à la JUnit)

```
<?php
require_once('simpletest/autorun.php');
require_once('../classes/log.php');
require_once('../classes/session_pool.php');

class TestOfSessionLogging extends UnitTestCase {

    function setUp() {
        @unlink('/temp/test.log');
    }

    function tearDown() {
        @unlink('/temp/test.log');
    }

    function testLoggingInIsLogged() {
        $log = new Log('/temp/test.log');
        $session_pool = &new SessionPool($log);
        $session_pool->login('fred');
        $messages = file('/temp/test.log');
        $this->assertEqual($messages[0], "User fred logged in.\n");
    }
}
?>
```

Test de page web

❑ Possibilité de tester une page web

- Assez basique (pas de javascript)

❑ Fonctionnalités

- Charger une page, naviguer

```
<?php
require_once('simpletest/autorun.php');
require_once('simpletest/web_tester.php');

class TestOfAbout extends WebTestCase {
    function testOurAboutPageGivesFreeReignToOurEgo() {
        $this->get('http://test-server/index.php');
        $this->click('About');
        $this->assertTitle('About why we are so great');
        $this->assertText('We are really great');
    }
}
?>
```

❑ Fonctionnalité

- Naviguer à travers des formulaires

```
<?php
require_once('simpletest/autorun.php');
require_once('simpletest/web_tester.php');

class TestOfRankings extends WebTestCase {
    function testWeAreTopOfGoogle() {
        $this->get('http://google.com/');
        $this->setField('q', 'simpletest');
        $this->click("I'm Feeling Lucky");
        $this->assertTitle('SimpleTest - Unit Testing for PHP');
    }
}
?>
```

Suite du tutorial

- ❑ Pour débiter :
 - <http://simpletest.org/fr/start-testing.html>
- ❑ Pour créer un premier test, avoir quelques exemples :
 - http://simpletest.org/fr/first_test_tutorial.html
- ❑ Pour naviguer dans l'API :
 - <http://simpletest.org/en/overview.html>
- ❑ Les exemples de Philippe Renevier :
 - <http://deptinfo.unice.fr/twiki/bin/view/Linfo/ProjetDev2012TestsUnitaire>

Application à vos projets

- ❑ **Tout le code doit être documenté raisonnablement**
 - pages et classes PHP

- ❑ **Comment ?**
 - entête du module/classe
 - commentaire sur les fonctions/opérations les plus importantes
 - (pas forcément tous les getters/setters)