
SimpleTest

Unit testy pro PHP

Tomáš Kadlec



Obsah

- Unit testy?
- xUnit
- SimpleTest
 - Schopnosti frameworku
 - Instalace
 - Základní principy
 - Sady testů
 - Mock objekty
- Příklad (EDUX ProviderManager)



Unit testy

- **Problém: změnou kódu programové jednotky mohou vzniknout chyby**
- **Unit testy**
 - verifikace, validace software
 - automatické (programové) testování jednotek
 - regresního testování
 - nástroje testování jeden z nejbližších vývojáři
 -
- **Jednotka**
 - nejmenší testovatelná část software (procedura, objekt ...)

xUnit – základní pojmy I

- struktura unit testů se neliší mezi různými technologiemi a jazyky → usnadnění pro vývojáře
 - nástroje pro podporu unit testů (JUnit, PHPUnit, SimpleTest ...)
- Test fixtures (kontext)
 - sada předpokladů nebo stavy, které jsou potřeba aby test mohl uspět
- Test case (testovaný případ)
 - sada podmínek – testovaná jednotka pracuje správně či nikoliv
- Test suites (sada testů)
 - sada testů, které sdílejí tentýž *kontext*, na pořadí nezáleží

xUnit – základní pojmy II

- Assertion (tvrzení, jednotková logická funkce)
 - funkce, metoda, makro atp.
 - ověřuje chování testované jednotky
 - chyba znamená ukončení prováděného testu a indikaci chyby.
- method stub
 - simuluje chování funkce
- mock objects
 - simulují reálné objekty, ale jsou kontrolovány programátorem → tyto objekty netestujeme, ale ony mohou testovat naše objekty
- fake objects
 - jednodušší mock objekty - „pouze“ implementují stejný interface

xUnit – vykonání testu

```
// prepare a 'world' - an isolated environment for testing
setup();

...
/* Body of test */
...

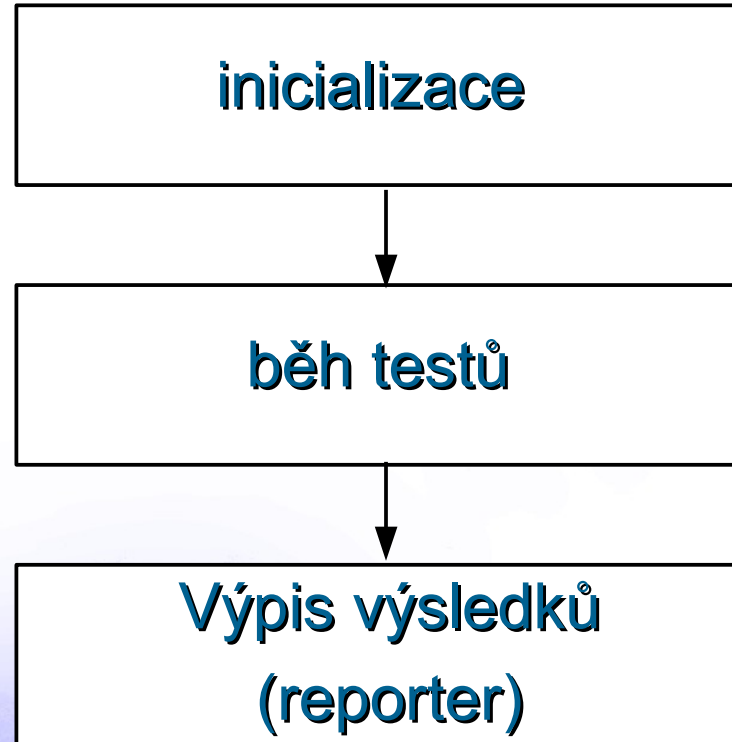
// in the end, whether succeed or failed clean up
// the 'world' to not disturb other tests or code
teardown();
```

SimpleTest

- framework pro unit testy aplikací v PHP (4.3, 5.3)
- verze 1.0.1
- Unit test case
- Html display
- Autoloading of test cases
- Mock objects
- Web test case
- Partial mocks
- Web cookie handling
- Following redirects
- Form parsing
- Command line interface
- XML output and parsing
- Browser component
- HTTP authentication
- SSL support
- Proxy support
- Frames support
- File upload testing



SimpleTest – běh testu



SimpleTest - instalace

1. <http://www.simpletest.org/en/download.html>
2. přidat umístění do PHP `include_path` (pokud lze) nebo nahrát k testované aplikaci

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

class MyTestCase extends UnitTestCase {
    function test...() {
    }
}
?>
```

SimpleTest – Unit test case

■ UnitTestCase

- výchozí třída pro vytváření testovaných případů
- konstruktor se typicky nepoužívá
- Inicializace – setUp() / ukončení tearDown()

```
require_once('simpletest/autorun.php');  
  
class MyTestCase extends UnitTestCase {  
    function setUp() { ... }  
  
    function tearDown() { ... }  
  
    function testFirst() {  
        $this->assertTrue( ... );  
    }  
}
```

název test case

inicializace

úklid

název testu

test

tvrzení

SimpleTest – UnitTestCase - tvrzení

- `assertTrue($x) - assertFalse($x)`
- `assertNull($x) - assertNotNull($x)`
- `assertIsA($x, $t) - assertNotA($x, $t) // class, type`
- `assertEqual($x, $y) - assertNotEqual($x, $y) // ==`
- `assertReference($x, $y)`
- `assertClone($x, $y)`
- `assertPattern($p, $x) // regexp`
- `expectError($x) // zachycení chyby`
- `assert($e) // Expectation objekt`

SimpleTest – UnitTestCase – další metody

- metody se mohou hodit zejména při úpravách reporteru
 - `pass()` - `fail()` // vyvolá úspěch/neúspěch testu
 - `error()` // vyvolá chybu
 - `signal($type, $payload)` // zpráva do reporteru
 - `dump($var)` // `print_r()`
- jelikož rozšiřujeme `UnitTestCase`, můžeme libovolně přidávat tvrzení (metody *assert*)

SimpleTest – Test suite

- testy lze seskupovat do pojmenovaných skupin – test suite
 - komplexní testy
 - více test case může být v jednom souboru, přeskočení → abstract
 - snadné spouštění

```
require_once('simpletest/unit_tester.php');  
require_once('simpletest/reporter.php');
```

```
$test = &new  
    TestSuite('All tests');  
$test->addTestCase  
    (new FileTestCase());  
$test->run( ... );
```

```
class MyTestSuite extends  
TestSuite {  
    function FileTestSuite() {  
        $this->TestSuite('Alltests');  
        $this->addTestFile('...');  
    }  
}
```

SimpleTest – Mock objekty

■ „Herec“

- simulace objektů, které je komplikované vytvořit pro účely testu (závislosti, časová náročnost, bezpečnost, ...)
- kontrola prostředí, rychlost běhu
- relativně nízká náročnost

■ „Kritik“

- provádí kontrolu zaslaných zpráv
 - správné volání
 - správné parametry
 - ...



SimpleTest – Mock objekty - vytvoření

```
require_once('simpletest/unit_tester.php');
require_once('simpletest/mock_objects.php');

class Configuration {
    function getValue() { ... }
}

Mock::generate('Configuration');

class MyTestCase extends UnitTestCase {
    function testSomething() {
        $connection = &new MockConfiguration();
    }
}
```

Vytvoření třídy
MockConfiguration
Použití třídy

SimpleTest – Mock objekty – návratové hodnoty

```
// objekt $iterator má metodu next()  
$iterator->setReturnValue('next', false);  
$iterator->setReturnValueAt(0, 'next', 'First');  
$iterator->setReturnValueAt(1, 'next', 'Second');
```

Pořadí

Návratová hodnota

```
// objekt $config má metodu get($key)  
$config->setReturnValue('get', '1', 'aaa');  
$config->setReturnValue('get', '2', 'bbb');
```


SimpleTest – Mock objekty – „Kritik“

- `expect($method, $args)`
- `expectAt($timing, $method, $args) // 0`
- `expectCallCount($method, $count)`
- `expectMaximumCallCount($method, $count)`
- `expectMinimumCallCount($method, $count)`
- `expectNever($method)`
- `expectOnce($method, $args)`
- `expectAtLeastOnce($method, $args)`

SimpleTest – reporter I

- **HtmlReporter**
 - HTML výstup, jednoduchý, výchozí
- **TextReporter**
 - pro příkazovou řádku

```
$test = &new TestSuite('All tests');  
...  
$test->run(new HtmlReporter());
```

- **SimpleReporter**
 - Lze si vytvořit vlastní reporter rozšířením třídy

SimpleTest – reporter II

- `HtmlReporter($encoding)`
- `paintHeader($test_name)`
- `paintFooter($test_name)`
- `paintMethodStart($test_name)`
- `paintMethodEnd($test_name)`
- `paintFail($message)`
- `paintPass($message)`
- `getTestList()`
- `GetPassCount()`
- `GetFailCount()`
- `getExceptionCount()`



SimpleTest – příklad



Υ26TW1 – SimpleTest
(20)



Děkuji za pozornost.

