# MOCKERY, Reflection, 성공적

전창완 (im@wani.kr)

## Contents

- Test private property & method
- Tests the Closure with Mockery

1

## Test private property & method

mordern
php user group

## Private에는 어떻게 접근한담..?

## 변수의 경우

- assertAttributeInstanceOf
- assertAttributeEquals / Same
- …assertAttribute*

```php
<?php
namespace Wandu\PugSample;

class Customer
{
    /** @var string */
    private $name;

    /**
     * @param string $name
     */
    public function __construct($name)
    {
        $this->name = $name;
    }

    /**
     * @return string
     */
    public function getName()
    {
        return ucfirst($this->name);
    }
}
```

```php
<?php
namespace Wandu\PugSample;

class Greeter
{
    /** @var Customer */
    private $customer;

    /**
     * @param Customer $customer
     */
    public function __construct(Customer $customer)
    {
        $this->customer = $customer;
    }

    /**
     * @return string
     */
    public function sayHello()
    {
        return "Hello, {$this->customer->getName()}! (public)";
    }
}
```

```php
public function testSayHelloPublic()
{
    $mockCustomer = Mockery::mock(Customer::class);
    $mockCustomer->shouldReceive('getName')->andReturn('Changwan');

    $greeter = new Greeter($mockCustomer);

    $this->assertAttributeInstanceOf(Customer::class, 'customer', $greeter);
    $this->assertAttributeEquals($mockCustomer, 'customer', $greeter);

    $this->assertEquals('Hello, Changwan! (public)', $greeter->sayHello());
}
```

## 그렇다면 매서드는?

## Reflection을 이용하자!

http://php.net/manual/class.reflectionclass.php

http://php.net/manual/class.reflectionmethod.php

```php
<?php
namespace Wandu\PugSample;

class Greeter
{
    /** @var Customer */
    private $customer;

    /**
     * @param Customer $customer
     */
    public function __construct(Customer $customer)
    {
        $this->customer = $customer;
    }

    /**
     * @return string
     */
    public function sayHelloPublic()
    {
        return "Hello, {$this->customer->getName()}! (public)";
    }

    /**
     * @return string
     */
    private function sayHelloPrivate()
    {
        return "Hello, {$this->customer->getName()}! (private)";
    }
}
```

```php
public function testSayHelloPrivate()
{
    $mockCustomer = Mockery::mock(Customer::class);
    $mockCustomer->shouldReceive('getName')->andReturn('Wandu');

    $greeter = new Greeter($mockCustomer);

    $classReflection = new ReflectionClass(Greeter::class);
    $sayHelloPrivateMethod = $classReflection
        ->getMethod('sayHelloPrivate')
        ->getClosure($greeter);

    $this->assertInstanceOf(Closure::class, $sayHelloPrivateMethod);
    $this->assertEquals('Hello, Wandu! (private)', $sayHelloPrivateMethod->__invoke());
}
```

## 이것만 하기 심심해서 하나 더!

mordern
php user group

2

## Tests the Closure with Mockery

mordern
php user group

## 클로져는 Final이라서..
## Mockery 사용이 안된대요.

```php
<?php
namespace Wandu\PugSample;

use Closure;

class Greeter
{
    /** @var Customer */
    private $customer;

    /**
     * @param Customer $customer
     */
    public function __construct(Customer $customer)
    {
        $this->customer = $customer;
    }

    /**
     * @param callable $handler
     * @return mixed
     */
    public function doSomethingWithCustomer(Closure $handler)
    {
        return $handler->__invoke($this->customer);
    }
}
```

```php
public function testDoSomethingSimple()
{
    $mockCustomer = Mockery::mock(Customer::class);
    $mockCustomer->shouldReceive('getName')->andReturn('Wandu');

    $greeter = new Greeter($mockCustomer);

    $this->assertEquals(30, $greeter->doSomethingWithCustomer(function ($param1) use ($mockCustomer) {
        $this->assertSame($mockCustomer, $param1);
        return 30;
    }));
}
```

클로져 내부에 있는 테스트
구문이 정말 실행이 되는거야..?

mordern
php user group

Mockery에 아름다운 함수들을 도입해볼 방법이 없을까?

```php
<?php
namespace Wandu\PugSample;

class Dummy
{
    public function handler()
    {}
}
```

```php
public function testDoSomethingWithDummy()
{
    $mockCustomer = Mockery::mock(Customer::class);
    $mockCustomer->shouldReceive('getName')->andReturn('Wandu');

    $mockHandler = Mockery::mock(Dummy::class);
    $mockHandler->shouldReceive('handler')->once()->with($mockCustomer)->andReturn(30);

    $greeter = new Greeter($mockCustomer);

    $classReflection = new ReflectionClass(get_class($mockHandler));
    $handler = $classReflection->getMethod('handler')->getClosure($mockHandler);

    $this->assertEquals(30, $greeter->doSomethingWithCustomer($handler));
}
```

..끗