

The PHPSpec Zend Framework App Testing Manifesto: ZF Integration


In the [Preamble](#) to this Manifesto, I set out the basic proposal for direct integration of support for applying Behaviour-Driven Development (BDD) to the Zend Framework. The ideal was to move away from line-by-line setup and manipulation of Zend instances towards a more simplified model which is standardised for any [PHPSpec](#) specs.

This morning I merged the experimental branch for the implementation of the Zend Framework specific `PHPSpec_Context` into the main trunk for release in PHPSpec 0.3.0. Documentation of the Zend Context will be completed within the next day or two and added to the growing PHPSpec Manual at <http://dev.phpspec.org/manual> now available.

The main customisation required to get the Zend Context operational is to set up elements of the FrontController, I've selected two primary methods. The first is to use several static methods on `PHPSpec_Context_Zend` (`addControllerDirectory()` and `addModuleDirectory()`) and a more flexible `addFrontControllerSetupCallback()` which will accept a custom function or static method reference to execute. The former is handy since you can easily organise your bootstrap file into a static class and segregate FrontController, Dispatcher, Action Helper, etc. elements you are using to customise how the Zend Framework operates. Both can be called from any file included into a Context file (e.g. `SpecHelper.php`).

Here's a really simple example of a Zend Framework spec. I'll post a more in-depth tutorial next week once the Manual is finalised, and some final cleanups applied (e.g. allowing for Router specification). The only priority for basic operations left is an implementation of a `Zend_Factory` class to allow you to replace objects directly instantiated within controllers (to maintain the golden standard of specifying/testing controllers isolated from it's dependencies). We'll really start kicking some ZF ass when `PHPMock` has an initial devel release!


```
PHPSpec_Context_Zend::addControllerDirectory('../src/application/controllers');
class DescribeTwitterController extends PHPSpec_Context_Zend
{
    public function itShouldDisplayHomeTweetsFromDefaultRetrieveAction()
    {
        $this->get('retrieve');
        $this->response()->should->haveText('Tweets From Twitterers Followed:');
    }
    public function itShouldPostTweetOnSendActionUsingPostData()
    {
        $this->post('send', array('message'=>'@padraicb: All your specs are belong to us'));
        $this->response()->should->haveText('You submitted a new Tweet!');
    }
    public function itShouldRetrieveMessagesForUserIdInParamsFromRetrieveAction()
    {
        $this->get('twitter/retrieve/user/padraicb');
        $this->response()->should->beSuccess();
    }
}
```

It's obvious this is bare minimum. There is still a lot of ground to cover incl. integrating PHPMock, so you can literally mock/stub the Twitter service package being used in the controller action itself (Note to Matthew - implement Zend_Service_Twitter already... 

) and set expectations of how the Controller should use that Twitter service. Adding some basic HTML/XML safe Matchers would also add some fairly substantial help for specifying what the response output should be, or what template any controller action should render.


But the basics, and in short the most time consuming, bit of integration to get something working with the Zend Framework at a level not requiring any devious workarounds is now complete. The weekend should see some really useful additions to this base. Your comments are most welcome as usual whether here, or on the mailing list.

Posted by Pádraic Brady in PHP General, PHP Security at 01:21

Just a thought - If you intend for the test code itself to act as documentation, you could create more meaningful test methods names, and not just repeat the test code essence in concentrated manner. (Seriously, those method names are overkill 

) Anonymous on Jan

22 2008, 06:59

Or I could just phpspec -rs and leave well enough alone 

. I like the longer names so I know what the method is specifying. It's not an unusual BDD practice. Anonymous on Jan 22 2008, 17:35

Having read some of your posts, it would seem you have a big affinity to mock everything you can. I'm interested to know what you thing about Martin Fowler's approach - <http://martinfowler.com/articles/mocksArentStubs.html> Anonymous on Jan 26 2008, 22:27

As Fowler states in the article, since I follow the BDD oriented practices I'm essentially an obsessive Mockist who will mock controllers in the absence of any real Model or View classes - i.e. while writing controllers my views and models may not yet exist since I'm using mocks to examine their future APIs before coding them.

Granted it's an absolutely aberrant practice in PHP which essentially has only one working Mock Object library (and that integrated solely into SimpleTest) but the approach is not uncommon outside PHP for Rails, Django, etc. If anything non-PHP tutorials on the technique are drowning Google...


As for the article - I regularly reference it. It's an excellent dissection of the two approaches to TDD isolation between those who Mock, and those who use varying degrees of non-interrogated test doubles. Anonymous on Jan 27 2008, 06:08

I have been developing in PHP for over 4 years and only recently got into unit testing in a serious way. I have been reading much on testing from what I could find on the Internet and in professional literature (xUnit test patterns) and am trying to figure out the correct testing strategy for my current project.

It reminds me a lot of journey I took when I was introduced into OOP - and I wish to learn from the lessons of the past and get to the best (or simply better) practices earlier.

I would very much appreciate it if you could spare the time to discuss with me via email some of the testing practices and strategies, as you seem passionate about the subject and generally well informed. Anonymous on Jan 27 2008, 06:47

Hi Eran, you can startup a conversation with me by email to:

padraic dot brady at yahoo dot com 

Anonymous on Jan 28 2008, 17:23