

PHPSpec Reporting Gets A Needed Boost

PHPSpec is closing in on its first stable release, so the time had finally come to spruce up its output! No more the simple reporting of failed specs - now you get a few more details in a readable format, exceptions and errors even come with traces. In addition, I've implemented specdoc output as an option (using "-s") so you can get a list of specs in their plain text form.

I've pasted in a copy of the output from a real PHPSpec run and added a few issues to show how they are output. I've been writing some experimental source code for a possible Mock Object framework in PHP (see last blog entry), so I used PHPSpec to apply Behaviour-Driven Development in its development. You can read the actual spec files over at <http://phpmock.googlecode.com/svn/branches/padraic/specs/>.

PHPSpec development snapshots have been updated on <http://dev.phpspec.org>, which also hosts the manual being written (HTML).

```
$ phpspec -rs
```

```
EE.....P..
```

```
Finished in 0.334152936935 seconds
```

```
p h p mock method expectations
-should expect each method exactly once unless otherwise specified (ERROR)
-should throw default exception if method call count is unexpected (EXCEPTION)
-should expect number of methods in times term
-should return self from times term invocation
-should set return value in terms and default to returning value for all calls
-should return values in order of setting but return last value remaining on
other calls
-should return self from return term invocation
-should set expected args using with term
-should return self from with term invocation
-should allow no calls for zeroormoretimes term
-should allow one call with once term
-should throw exception if less than once using once tern
-should throw exception if greater than once using once term
-should return self instance from once term
-should allow two calls with twice term
-should throw exception if less than twice using twice term
-should throw exception if greater than twice using twice term
-should return self instance from twice term
-should allow no calls with never term
-should throw exception if any call after never term
-should return self instance from never term
-should allow any calls for zeroormoretimes term
-should allow zero calls for zeroormoretimes term
-should return self instance from zeroormoretimes term
-should set times minimum with once using atleast term
-should set times minimum with times using atleast term
```

- should throw exception if less than minimum using atleast term
- should return self instance from atleast term
- should set times maximum with once using atmost term
- should set times maximum with times using atmost term
- should throw exception if more than twice using twice term
- should return self instance from atmost term
- should allow any args using withanyargs term
- should return self instance from withanyargs term
- should disallow args using withnoargs term
- should return self instance from withnoargs term
- should obey ordering via sequence of ordered term calls
- should disallow method calling if method has specified order
- should allow method calling of unordered expectations in any order
- should return self instance from ordered term
- should throw specified exception using andthrow term
- should throw specified exception with message using andthrow term
- should throw exception if class passed to andthrow term not an exception
- should return self instance from andthrow term

p h p mock

- should create mock inheriting class type from original
- should create mock inheriting type if original an interface
- should create mock with specified name
- should create stub from an array of methods and return values
- should throw exception if class is final
- should throw exception if class does not exist
- should not throw class not exists exception if an interface
- should mock all allowed public methods which do nothing since mocked
- should mock public static methods which do nothing since mocked (PENDING)
- should retain method parameter definitions in mocks
- should return true on validation by default if no expectations set

55 examples, 0 failures, 1 error, 1 exception, 1 pending

Errors:

1)

```
'p h p mock method expectations should expect each method exactly once unless otherwise specified' ERROR
exception 'PHPSpec_Runner_ErrorException' with message 'more bad stuff happenin'' in
/user/opt/projects/phpmock/specs/PHPMockMethodExpectationsSpec.php:19
Stack trace:
#0 [internal function]: PHPSpec_ErrorHandler(1024, 'more bad stuff ...', '/user/opt/...', 19, Array)
#1 /user/opt/projects/phpmock/specs/PHPMockMethodExpectationsSpec.php(19): trigger_error('more bad stuff ...')
#2 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Example.php(81): DescribePHPMockMethodExpectations->itShouldExpectEachMethodExactlyOnceUnlessOtherwiseSpecified()
#3 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Collection.php(73):
```

```
PHPSpec_Runner_Example->execute()  
#4 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Base.php(51):  
PHPSpec_Runner_Collection->execute(Object(PHPSpec_Runner_Result))  
#5 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Base.php(44):  
PHPSpec_Runner_Base->executeExamples()  
#6 /user/opt/projects/phpspec/trunk/src/PHPSpec/Console/Command.php(82):  
PHPSpec_Runner_Base::execute(Object(PHPSpec_Runner_Collection),  
Object(PHPSpec_Ru  
nner_Result))  
#7 /user/opt/projects/phpspec/trunk/src/PHPSpec/Console/Command.php(125):  
PHPSpec_Console_Command::main()  
#8 {main}
```

Exceptions:

```
1)  
'p h p mock method expectations should throw default exception if method call  
count is unexpected' EXCEPTION  
exception 'PHPSpec_Exception' with message 'no specification object created yet'  
in /user/opt/projects/phpspec/trunk/src/PHPSpec/Context.php:92  
Stack trace:  
#0 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Example.php(89):  
PHPSpec_Context->getCurrentSpecification()  
#1 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Collection.php(73):  
PHPSpec_Runner_Example->execute()  
#2 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Base.php(51): PHSp  
ec_Runner_Collection->execute(Object(PHPSpec_Runner_Result))  
#3 /user/opt/projects/phpspec/trunk/src/PHPSpec/Runner/Base.php(44):  
PHPSpec_Runner_Base->executeExamples()  
#4 /user/opt/projects/phpspec/trunk/src/PHPSpec/Console/Command.php(82):  
PHPSpec_Runner_Base::execute(Object(PHPSpec_Runner_Collection),  
Object(PHPSpec_Runner_Result))  
#5 /user/opt/projects/phpspec/trunk/src/PHPSpec/Console/Command.php(125):  
PHPSpec_Console_Command::main()  
#6 {main}
```

Pending:

```
1)  
'p h p mock should mock public static methods which do nothing since mocked'  
PENDING  
Static methods will need static handling  
Posted by PÃ¡draic Brady in PHP General, PHP Security at 21:27
```

Hey PÃ¡draic,

I looked through the docs, but I don't see how to run the specs with a the Runner rather than from the command line. I recall you saying that you were going to make it so you could just point it at a directory and it would run everything.

Christopher

PS - I may have a project to run PHPSpec through its paces Anonymous on Nov 15 2007, 05:33

Hi Christopher,

I haven't gotten around to adding a file runner just yet. If you check the test coverage, the specific area of running tests is largely uncovered (it wasn't a major concern until the framework became useful). I'll likely refactor/rewrite the running system during tomorrow so you can have a file based runner (http) and command line runner (cli) which both share possible Reporters (current only plain text).

I think after that it's down to adding a few more matchers, and looking into better Exception expectations via the DSL, and it's ready for final review to an alpha or beta release of 0.2.0.

Any observations from a real project are immensely valuable - I started using it over the weekend for PHPMock (need Mock support in PHPSpec, in an independent form since it will likely be usable from PHPT and PHPUnit at a minimum). Any development since the weekend is now needs based - get any final badly needed functionality defined and implemented outside the usual XP process we're using. Anonymous on Nov 15 2007, 06:24

PÃ¼draic,

I am in the process of trying to resurrect a PHP5 version of the Skeleton framework. Given the disenchantment with Zend and other frameworks I figured I would bring back that code, which I had abandoned. The plan is lighter weight than the current frameworks and compatible so it is designed to use external components (e.g. Swift, Doctrine, Zend, etc.).


The frame had light SimpleTest coverage, but in the upgrade I tossed out a lot of the tests rather than try to upgrade. Which leaves me open to new ideas, but I do need a runner.

Also, have you given any thought to either a test scaffolding builder or something that would automate creating tests for existing classes? Could I help the project in any of these areas -- scaffolding, automation, runners?

Christopher Anonymous on Nov 16 2007, 04:42

A few days behind my comment approval - sorry for the delay in approving and responding!

>I am in the process of trying to resurrect a PHP5 version of the
>Skeleton framework. Given the disenchantment with Zend and other
>frameworks I figured I would bring back that code, which I had
>abandoned. The plan is lighter weight than the current
>frameworks and compatible so it is designed to use external
>components (e.g. Swift, Doctrine, Zend, etc.).

I noticed an upsurge in ap3 posts to my SF lists folder 

>The frame had light SimpleTest coverage, but in the upgrade I tossed out a
>lot of the tests rather than try to upgrade. Which leaves me open to new
>ideas, but I do need a runner.


As previously, it's upcoming. PHPSpec has a long list of goals and user stories with assigned weightings - the wall in front of me only has 7 post-its on PHPSpec so it's just a few days from implementation.

>Also, have you given any thought to either a test scaffolding builder or
>something that would automate creating tests for existing classes? Could I
>help the project in any of these areas -- scaffolding, automation,
>runners?

Scaffolding is something I want soon - but it bears discussion on what to implement for scaffolding. For example, I use post-its here to record user stories, which lead to plain text specifications stored in a local DB. So I need scaffolding to take in newline delimited output of specs in an agreed format, and generate base Specs which are all marked pending. On the otherside, having a class based scaffolder is likewise useful - a bit limited in that BDD is spec related, over method related.

If you want to contribute along these lines, even just what you need personally, you should post the proposal to the main mailing list so Travis has a chance to offer any comments. 0.2.0 will likely see an alpha release by end of November so there is room for some needs-based contributions to give PHPSpec a nice round sense of purpose for those considering using it.

<http://groups.google.com/group/phpspec-dev>

Note, any help is truly appreciated 

. I'm not big on code ownership as the local tin god. Anonymous on Nov 18 2007, 08:49

PÃdraic, I PM'd you as well. Let me know if I can help with the web based test runner. If you have a basic direction you want to go, I can run with that. As I said, I am looking for a system where you can create a index.php file that contains a few lines of PHPSpec code, put it in the top level of your tests directory, and it just works. I think if it was just dead simple it would be a major draw for PHPSpec.

Also, something that would create specs from existing classes just for coverage. I know it is backwards from you BDD intent, but again it would be a draw to get existing projects to use the tool.

Let me know ... feel free to email me directly. Anonymous on Nov 19 2007, 23:07