Zend_View_Helper_Abstract and _Interface - Jurriën Stutterheim

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_View_Helper_Abstract Component Proposal

Proposed Component Name	Zend_View_Helper_Abstract
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_View_Helper_Abstract
Proposers	Jurriën Stutterheim
Zend Liaison	Matthew Weier O'Phinney
Revision	1.0 - 11 June 2008: Initial Draft. 1.1 - 05 Aug 2008: Addition of direct() method. (wiki revision: 16)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_View_Helper_Abstract is a very simple abstract class for view helpers. It provides the setView() method and the public $view attribute. At the moment these two are being copy/pasted in every view helper that needs it. Most helpers however can just as easily take this abstract class as parent to save duplicate code.

Additionally a Zend_View_Helper_Interface could be added. It would contain the signature for the setView() method. This would allow for more complex view helpers, like HeadTitle, which already inherit from another class to be identified as well.

Finally, a long-term goal for view helpers is to remove the tedious and difficult to understand naming of the helper method and instead implement the strategy pattern: all helpers would use the same method name. This would simplify implementation of calling helpers within Zend_View, and make it easier to visually identify the method that does the helper work. The proposal is to use the method name 'direct', which is already currently used in action helpers and is a fairly standard name for use with the strategy pattern. In the short term, it would go un-used, but would be available for future compatibility.

2. References

- Zym_View_Helper_Abstract
- Matthew's post about naming and PHP 5.3

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** provide the basic setView() implementation for view helpers
- This component **will** provide an interface with a setView() signature
- This component **will** provide an interface with a direct() signature
- This component **will not** implement any functionality around direct() at this time.

4. Dependencies on Other Framework Components

5. Theory of Operation

6. Milestones / Tasks

- Milestone 1: [DONE] Prototype available
- Milestone 2: [DONE] Unit tests available
- Milestone 3: [DONE] Component checked into incubator
- Milestone 4: [DONE] View helper documentation is updated
- Milestone 5: [DONE] Component checked into core
- Milestone 6: [DONE] Update the existing helpers to utilize these classes.

7. Class Index

- Zend_View_Helper_Abstract
- Zend_View_Helper_Interface

These classes should get another name, because of ZF's future adoptation of PHP 5.3 namespaces. See the link to Matthew's blog post in the references section. Suggestions:

- Interface:
  - Zend::View::Helper::IHelper
  - Zend::View::Helper::Helper
- Abstract:
  - Zend::View::Helper::HelperAbstract
  - Zend::View::Helper::Base

8. Use Cases

UC-01

```
class Zend_View_Helper_Form { .. }
```

Would become:

```
class Zend_View_Helper_Form extends Zend_View_Helper_Abstract { .. }
```

This allows:

```
/**
 * @var Zend_View_Instance
 */
public $view;

/**
 * Set view object
 *
 * @param  Zend_View_Interface $view
 * @return Zend_View_Helper_Form
 */
public function setView(Zend_View_Interface $view)
{
    $this->view = $view;
    return $this;
}
```

To be removed from the Form helper

UC-02

```
abstract class Zend_View_Helper_Placeholder_Container_Standalone implements IteratorAggregate,
Countable, ArrayAccess { }
```

Could become:

```
abstract class Zend_View_Helper_Placeholder_Container_Standalone extends Zend_View_Helper_Abstract
implements IteratorAggregate, Countable, ArrayAccess { }
```

UC-03

Currently helpers are checked if they have setView like so:

```
if (method_exists($store[$name], 'setView')) {
    $store[$name]->setView($this);
}
```

This can be done much nicer now:

```
if ($store[$name] instanceof Zend_View_Helper_Interface) {
    $store[$name]->setView($this);
}
```

Additionally, eventually instead of creating a formatted version of the helper method name, we will simply be able to call direct:

```
return call_user_func_array(array($store[$name], 'direct'), $args)
```

9. Class Skeletons

```
interface Zend_View_Helper_Interface
{
    public function setView(Zend_View_Interface $view);
    public function direct();
}
```

```
abstract class Zend_View_Helper_Abstract implements Zend_View_Helper_Interface
{
    public $view;

    public function setView(Zend_View_Interface $view)
    {
        $this->view = $view;
        return $this;
    }

    public function direct()
    {
    }
}
```

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>