Zend_Dojo - Arbitrary Dijit Support - Matthew Weier O'Phinney

<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_Dojo Arbitrary Dijit Support - Matthew Weier O'Phinney Component Proposal

Proposed Component Name	Zend_Dojo Arbitrary Dijit Support - Matthew Weier O'Phinney
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Dojo Arbitrary Dijit Support - Matthew Weier O'Phinney
Proposers	Matthew Weier O'Phinney
Zend Liaison	Ralph Schindler
Revision	1.0 - 14 January 2009: Initial Draft. (wiki revision: 5)

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
Rendering a custom dijit
Rendering a custom container dijit
9. Class Skeletons

1. Overview

Zend_Dojo provides scaffolding for the Dojo environment, as well as view helpers and form support for a variety of Dijits. One key feature of Dojo is the ability to define your own dijits as well as to extend existing dijits. This goal of this proposal is to provide a view helper that can be used to generate markup and/or javascript for arbitrary dijits, as well as clearly define how to create view helpers to support your own dijits.

2. References

- Widget creation tutorial
- Zend_Dojo documentation

3. Component Requirements, Constraints, and Acceptance Criteria

- This proposal **will** define a generic view helper for generating arbitrary Dijit markup and/or javascript
- This proposal **will** result in documentation on how to create view helpers to support custom Dijits.

4. Dependencies on Other Framework Components

- Zend_Dojo

5. Theory of Operation

Zend_Dojo view helpers already have a standard signature:

```
string helperName(string $id, string $content, array $params, array $attribs)
```

where $params defines Dijit-specific parameters, and $attribs defines HTML attributes to be included in markup. $content may be ignored depending on the dijit.

Additionally, layout container dijits allow you to return the helper instance in order to perform tasks such as capturing content or adding sub items:

```
Zend_Dojo_View_Helper_Dijit helperName()
```

The generic dijit view helper that will be created with this proposal will come in two flavors: dijit and container. These will follow the above signatures, and allow the developer to use arbitrary dijit types. The only difference in calling is that the $params member 'dojoType' **must** be specified; this will be used to ensure that the appropriate dojo.require statement is called and that the dijit type is associated with the markup and/or javascript generated. Additionally, an optional $params member, 'displayType', will be used to indicate if the dijit is rendered as a "block" (which will render a by default) or as an "inline" element (rendered as a by default); "block" will be the default. Finally, the tag to use for rendering may be specified using the "tagType" $params member.

6. Milestones / Tasks

- Milestone 1: Create proposal and submit for review
- Milestone 2: Create working prototypes, including unit tests, of generic dijit view helpers
- Milestone 3: Document generic dijit view helpers
- Milestone 4: Create documentation indicating how to create new dijit view helpers for custom dijits

7. Class Index

- Zend_Dojo_View_Helper_CustomDijit
- Zend_Dojo_View_Helper_CustomDijitContainer

8. Use Cases

UC-01

Rendering a custom dijit

```
$this->customDijit('foo', $content, array(
    'dojoType'    => 'custom.Foo',
    'displayType' => 'inline',
));
```

UC-02

Rendering a custom container dijit

```
<?php
$this->customDijitContainer()->captureStart('foo', array('dojoType' => 'custom.Foo'));
?><p>This is some content to be captured.</p><?php
echo $this->customDijitContainer()->captureEnd('foo'); ?>
```

9. Class Skeletons

TBD; will mimic existing dijit support

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