

Zend_Navigation - Robin Skoglund

<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_Navigation Component Proposal

Proposed Component Name	Zend_Navigation
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Navigation
Proposers	Robin Skoglund
Zend Liaison	Alexander Veremyev (Zend Liaison)
Revision	1.0 - 5 February 2009: Initial Draft. 1.1 - 21 March 2009 Updated based on community feedback. (wiki revision: 14)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
 - Pages and containers
 - Finder methods
 - Rendering
 - l18n, L13n
 - ACL
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_Navigation and its set of view helpers intend to simplify the creation and rendering of navigational structures, such as menus, breadcrumbs, navigational head links, and XML sitemaps.

2. References

- [Zym_Navigation implementation](#)
- [Zym_Navigation documentation](#)
- [Breadcrumbs, Yahoo! Design Pattern Library](#)
- [Navigation Tabs, Yahoo! Design Pattern Library](#)
- [Sitemaps XML format](#)
- [Document relationships: the LINK element](#)
- [Link types](#)

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** be able to generate URLs for pages
 - This component **will** allow assembling of page URLs based on module/controller/action
 - This component **will** allow assembling of page URLs based on routes
 - This component **will** allow assembling of page URLs with user params
 - This component **will** allow custom URLs (i.e. not based on any of the above)
- This component **will** be able to automatically detect the currently active page in a navigation container
- This component **will** be able to render menus
- This component **will** be able to render breadcrumbs
- This component **will** be able to render navigational head links
- This component **will** be able to render sitemaps (Sitemaps XML)
- This component **will** have optional integration with ACL
- This component **will** have optional implicit localization
- This component **will not** contain any references to the Cthulhu Mythos

4. Dependencies on Other Framework Components

- Zend_Exception
- Zend_Controller_Action_Helper_Url
- Zend_Validate_Abstract
- Zend_View_Helper_HtmlElement

5. Theory of Operation

Zend_Navigation is a component for managing trees of pointers to web pages. Simply put: It can be used for creating menus, breadcrumbs, links, and sitemaps, or serve as a model for other navigation related purposes.

Pages and containers

There are two concepts in Zend_Navigation:

1. Pages

A page `Zend_Navigation_Page` in `Zend_Navigation` — in its most basic form — is an object that holds a pointer to a web page. In addition to the pointer itself, the page object contains a number of other properties that are typically relevant for navigation, such as label, title, etc.

2. Containers

A navigation container (`Zend_Navigation_Container`) is a container class for pages. It contains methods for adding, retrieving, deleting and iterating pages. It implements the SPL interfaces `RecursiveIterator` and `Countable`, and can thus be iterated recursively, either by using the SPL class `RecursiveIteratorIterator` class, or by implementing recursive iteration yourself using `foreach` loops or other iterators.

Both `Zend_Navigation` and `Zend_Navigation_Page` extend `Zend_Navigation_Container`, so both can contain any number of hierarchic levels of pages.

By design, it is not possible to create arbitrary graphs of pages. That is, a page X cannot have a parent Y that is also a descendant of page X. It is only possible to create trees.

Finder methods

Containers have finder methods for retrieving pages. They are `findOneBy($property, $value)`, `findAllBy($property, $value)`, and `findBy($property, $value, $all = false)`. Those methods will recursively search the container for pages matching the given `$page->$property == $value`. The first method, `findOneBy()`, will return a single page matching the property with the given value, or `null` if it cannot be found. The second method will return all pages with a property matching the given value. The third method will call one of the two former methods depending on the `$all` flag.

The finder methods can also be used magically by appending the property name to `findBy`, `findOneBy`, or `findAllBy`, e.g. `findOneByLabel('Home')` to return the first matching page with label `Home`. Other combinations are `findByLabel(...)`, `findOnlyByTitle(...)`, `findAllByController(...)`, etc. Finder methods also work on custom properties, such as `findByFoo('bar')`.

Rendering

Classes in the `Zend_Navigation` namespace do not deal with rendering of navigational elements. Rendering is done with navigational view helpers. However, pages contain information that is used by view helpers when rendering, such as; label, CSS class, title, lastmod and priority properties for sitemaps, etc.

I18n, L13n

The navigational helpers support translating of page labels and titles. You can set a translator of type `Zend_Translate` or `Zend_Translate_Adapter` in the helper using `$helper->setTranslator($translator)`, or like with other I18n-enabled components; adding the translator to `Zend_Registry` using the key `Zend_Translate`, in which case it will be found by the helpers.

ACL

All navigation view helpers support ACL inherently from the class `Zend_View_Helper_NavigationBase`. A `Zend_Acl` object can be assigned to a helper instance with `$helper->setAcl($acl)`, where `$helper` refers to an instance of a helper, and `$acl` is an ACL instance containing roles and possibly resources. The helpers can be assigned a role to use when iterating pages, by doing `$helper->setRole('member')` to set a role id, or `$helper->setRole(new Zend_Acl_Role('member'))` to set an instance. If ACL is used in the helper, the role in the helper must have rights for the page's resource and/or privilege to be included in a menu/breadcrumb/sitemap.

6. Milestones / Tasks

- Milestone 1: [DONE] Proposal created
- Milestone 2: [DONE] Working prototype checked into [public repository](#)
- Milestone 3: [DONE] Community review of proposal
- Milestone 4: [DONE] Proposal acceptance
- Milestone 5: [DONE] Passing unit tests and initial documentation committed to incubator
- Milestone 6: Review for inclusion in trunk

7. Class Index

- `Zend_Navigation`
- `Zend_Navigation_Container`
- `Zend_Navigation_Exception`
- `Zend_Navigation_Page`
- `Zend_Navigation_Page_Mvc`
- `Zend_Navigation_Page_Uri`
- `Zend_Validate_Sitemap_Changefreq`
- `Zend_Validate_Sitemap_Lastmod`
- `Zend_Validate_Sitemap_Loc`
- `Zend_Validate_Sitemap_Priority`
- `Zend_View_Helper_Navigation`
- `Zend_View_Helper_Navigation_Abstract`
- `Zend_View_Helper_Navigation_Breadcrumbs`
- `Zend_View_Helper_Navigation_Interface`
- `Zend_View_Helper_Navigation_Links`
- `Zend_View_Helper_Navigation_Menu`
- `Zend_View_Helper_Navigation_Sitemap`

8. Use Cases

9. Class Skeletons

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