Zend_Controller_Router_Route_Translatable - Ben Scholzen

<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_Controller_Router_Route_Translatable Component Proposal

Proposed Component Name	Zend_Controller_Router_Route_Translatable
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Controller_Router_Route_Translatable
Proposers	Ben Scholzen
Zend Liaison	Matthew Weier O'Phinney
Revision	1.0 - 2 March 2009: Initial Draft. (wiki revision: 9)

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_Controller_Router_Route_Translatable allows translation of fixed texts as well as an enumeration of parameters within URLs. The route basically works like the standard route.

2. References

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** allow translation of fixed texts
- This component **will** allow translation of enumerated parameters
- This component **will not** need a change of the way a user assembles the route or gets the parameters.
- This component **will** allow one-time-locales for translations into a different language than the current one.

4. Dependencies on Other Framework Components

- Zend_Translate
- Zend_Controller_Router_Route

5. Theory of Operation

Initialisation of the Route
To be able to use the route, it requires an instance of Zend_Translate. You can either set it via the the static class method:

```
Zend_Controller_Router_Route_Translatable::setTranslate(Zend_Translate $translate)
```

If the instance was not set via this method, the route looks for an instance in Zend_Registry with the key 'Zend_Translate'. If that one isn't set either, it will throw an exception.

Zend_Translate should be an instance, which already has the current locale set. After that you can simply use the route like any other.

Route definition

The route itself works like the usual Zend_Controller_Router_Route, as it separates the parameters with a slash by default. To insert a fixed translatable text, you simply write @foobar, where foobar is the message-id. For inserting a translatable parameter, you write :@foobar.

One-Time-Locale
Sometimes you need to create a route, which is translated into another language as the the one, for example when you have a language-switch link which should directly go to the same route again. In this case you call

```
Zend_Controller_Router_Route_Translatable::setOneTimeLocale(Zend_Locale $locale)
```

right before you assemble the route.

6. Milestones / Tasks

- Milestone 1: Community reviews collected
- Milestone 2: Prototype finalized and checked into SVN
- Milestone 3: Unit tests exist, work, and are checked into SVN.
- Milestone 4: Initial documentation exists.

7. Class Index

- Zend_Controller_Router_Route_Translatable

8. Use Cases

UC-01

```
// Set translate object with english locale set and create the route
Zend_Controller_Router_Route_Translatable::setTranslate($translate);
$route = new Zend_Controller_Router_Route_Translatable('@user/:@mode/:id');
```

UC-02

```
// Assemble the route in english: user/edit/1
echo $route->assemble(array('mode' => 'edit', 'id' => 1));
```

UC-03

```
// Assemble the route in german: benutzer/bearbeiten/1
Zend_Controller_Router_Route_Translatable::setOneTimeLocale('de_DE');
echo $route->assemble(array('mode' => 'edit', 'id' => 1));
```

UC-04

```
// Match the route in german. This assumes that setTranslate($translate)
// was set with a german locale.
$match = $route->match('benutzer/bearbeiten/1');
echo $match['mode']; // edit
echo $match['id'];   // 1
```

9. Class Skeletons

```
/**
 * Translatable route
 */
class Zend_Controller_Router_Route_Translatable extends Zend_Controller_Router_Route
{
    /**
     * Set a one time assembling locale
     *
     * @param  string $locale
     * @return void
     */
    public static function setOneTimeLocale($locale);

    /**
     * Set the translate object
     *
     * @param  Zend_Translate $translate
     * @return void
     */
    public static function setTranslate(Zend_Translate $translate);

    /**
     * Instantiates route based on passed Zend_Config structure
     *
     * @param Zend_Config $config Configuration object
     */
    public static function getInstance(Zend_Config $config);

    /**
     * Prepares the route for mapping by splitting (exploding) it
     * to a corresponding atomic parts. These parts are assigned
     * a position which is later used for matching and preparing values.
     *
     * @param string $route Map used to match with later submitted URL path
     * @param array $defaults Defaults for map variables with keys as variable names
     * @param array $reqs Regular expression requirements for variables (keys as variable names)
     */
    public function __construct($route, $defaults = array(), $reqs = array());

    /**
     * Matches a user submitted path with parts defined by a map. Assigns and
     * returns an array of variables on a successful match.
     *
     * @param string $path Path used to match against this routing map
     * @return array|false An array of assigned values or a false on a mismatch
     */
    public function match($path);

    /**
     * Assembles user submitted parameters forming a URL path defined by this route
     *
     * @param  array $data An array of variable and value pairs used as parameters
     * @param  boolean $reset Whether or not to set route defaults with those provided in $data
     * @return string Route path with user submitted parameters
     */
    public function assemble($data = array(), $reset = false, $encode = false);
}
```

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