Zend_Tool_Locale - Kim Blomqvist

<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_Tool_Locale Component Proposal

Proposed Component Name	Zend_Tool_Locale
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Tool_Locale
Proposers	Kim Blomqvist
Zend Liaison	TBD
Revision	1.0 - 1 September 2010: Initial Draft. (wiki revision: 26)

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_Tool_Locale parses project source files for translation message ids. Translation resource file is created or updated according to the matches.

2. References

- Demo code in github
- The parser presented here is possibly related to Zend\Code\Scanner.

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** create and update project translation resource files for array adapter
- This component **will** support plurals
- This component **will** support array keys

4. Dependencies on Other Framework Components

- Zend_Tool_Project_Provider_Abstract

5. Theory of Operation

The component is used from CLI, like this ...

```
zf create locale module[=application] kwords
```

6. Milestones / Tasks

- Milestone 1: [DONE] Working prototype
- Milestone 2: Migrate to ZF2
- Milestone 3: Working prototype checked into the incubator supporting use cases #1, #2, ...
- Milestone 4: Working prototype checked into the incubator supporting use cases #3 and #4.
- Milestone 5: Unit tests exist, work, and are checked into SVN.
- Milestone 6: Initial documentation exists.

If a milestone is already done, begin the description with "[DONE]", like this:

- Milestone #: [DONE] Unit tests ...

7. Class Index

- Zend_Tool_Locale
- Zend_FunctionParamParser

8. Use Cases

Test files

```
// FILE 1: application/forms/Test.php
public function init()
{
    /* Form Elements & Other Definitions Here ... */
    $this->addElement('text', 'username', array(
        'label' => 'Username',
        'description' => 'Type your username here',
        'filters' => array('StringTrim'),
        'validators' => array(
            array('Validator' => 'StringLength', 'options' => array(0, 20))
        )
    ));
}

// FILE 2: application/controllers/IndexController.php
public function indexAction()
{
    // Error
    $error = new My_Error('Error occured');
    $this->_helper->flashMessenger($error);

    // Success
    $success = new My_Success('Password changed successfully');
    $this->_helper->flashMessenger($success);
}

// FILE 3: application/views/scripts/index.phtml
$this->translate('Lorem ipsum dolor sit amet, consectetuer adipiscing elit.');
$this->translate('Hello %s', $username);
$this->translate(array('Car', 'Cars', $number));
```

UC-01 Create Finnish translations

```
// data/locales/fi/application.php does not exist yet

$ zf create language fi
Added 6 message id(s)
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Hello %s
Car
Cars
Username
Type your username here

// data/locales/fi/application.php is now created
<?php
return array (
  'Lorem ipsum dolor sit amet, consectetuer adipiscing elit.' => '',
  'Hello %s' => '',
  'Car' => '',
  'Cars' => '',
  'Username' => '',
  'Type your username here' => '',
);
```

UC-02 Update with additional keywords

```
$ zf create language fi application My_Error:My_Success
Added 2 message id(s)
Error occured
Password changed successfully

// data/locales/fi/application.php updated
<?php
return array (
  'Lorem ipsum dolor sit amet, consectetuer adipiscing elit.' => '',
  'Hello %s' => '',
  'Car' => '',
  'Cars' => '',
  'Username' => '',
  'Type your username here' => '',
  'Error occured' => '',
  'Password changed successfully' => '',
);
```

9. Class Skeletons

```
class Zend_Tool_Locale
    extends Zend_Tool_Project_Provider_Abstract
{
    const DEFAULT_PATTERN = "^'(.+)'\s*,|^'(.+)'$";

    protected $_func_kwords = array(
        'translate' => array(
            self::DEFAULT_PATTERN,
            "^array\(\s*'(.+)'\s*,\s*'(.+)'",
        ),
        'plural' => array(
            "^'(.+)',\s*'(.+)'",
        ),
        'setLabel',
        'setLegend',
        'setDescription',
    );

    protected $_arr_kwords = array(
        'label',
        'legend',
        'description',
    );

    public function create($locale, $module = 'application', $kwords = null);

    protected function _addKeyWords($kwords);
    protected function _loadTranslations($filename);
    protected function _saveTranslations($filename, array $translations);
}

class Zend_FunctionParamParser
{
    protected $_func_kwords = array();
    protected $_arr_kwords = array();

    protected $_basepath;
    protected $_relativePaths = array();

    protected $_extensions = array('php', 'phtml');
    protected $_exclude = array();

    protected $_parseRecursively = true;
    protected $_defaultPattern = '(.+)';

    protected $_matches;

    public function __construct($options = array());
    public function setOptions(array $options);
    public function getMatches();
    public function parseFile($filename, $throw = false);
    public function parseFolder($folder, $recursive = true);

    protected function _catMatches($a1, $a2);
    protected function _parseFuncParams($pattern, $params);
}
```

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