Zend_Filter_Minify_Css - Nick Daugherty

<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_Filter_Minify_Css Component Proposal

Proposed Component Name	Zend_Filter_Minify_Css
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Filter_Minify_Css
Proposers	Nick Daugherty
Zend Liaison	TBD
Revision	1.0 - 23 April 2010: Initial Draft. (wiki revision: 29)

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_Filter_Minify_Css is a filter to reduce the size of CSS strings and files, resulting in faster client downloads

2. References

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** minify Css code.
- This component **will** implement Zend_Filter_Interface
- This component **will** generate valid Css code
- This component **will** provide an adapter system for implementing different minification solutions
- This component **will** provide options for control of minification (such as opting not to remove comments, for example)

4. Dependencies on Other Framework Components

- Zend_Filter_Exception
- Zend_Filter

5. Theory of Operation

Minifying Css leads to faster load times on the client by reducing the overall size of Css code. This component aims to perform the work of removing comments, extra whitespace, etc while retaining the familiar Zend_Filter interface.

This component can be integrated into other framework components, such as HeadStyle view helper for automatic Css minification.

The component is used like any other Zend_Filter_* component, passing the string to be minified to the filter() method.

6. Milestones / Tasks

- Milestone 1: design notes will be published here
- Milestone 2: Working prototype checked into the incubator supporting minification of Css strings
- Milestone 3: Unit tests exist, work, and are checked into SVN.
- Milestone 4: Initial documentation exists.

7. Class Index

- Zend_Filter_Minify_Exception
- Zend_Filter_Minify_Css
- Zend_Filter_Minify_Interface
- Zend_Filter_Minify_Css_RegEx

8. Use Cases

UC-01

Minifying a string of Css

```
//If no options are passed, will use the default adapter (regex), which is currently the only adapter
available
$filter = new Zend_Filter_Minify_Css();

$minified = $filter->filter('body
    {
        color: #666;
        background-color: #000;

    }');
```

9. Class Skeletons

```
/**
 * @see Zend_Filter_Interface
 */
require_once 'Zend/Filter/Interface.php';


/**
 * @category   Zend
 * @package    Zend_Filter
 * @copyright  Copyright (c) 2005-2010 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class Zend_Filter_Minify_Css implements Zend_Filter_Interface
{
 /**
     * Minification adapter
     */
    protected $_adapter;

    /**
     * Store options
     */
    protected $_options;

    /**
     * Class constructor
     *
     * @param string|array $options (Optional) Options to set, if no adapter set, RegEx is used
     */
    public function __construct($options = null)
    {
        $this->setOptions($options);

        $this->setAdapter($options);
    }

    /**
     * Returns the adapter instance
     *
     * @return Zend_Filter_Minify_Css_Interface
     */
    public function getAdapter()
    {
        return $this->_adapter;
    }

    /**
     * Sets minification adapter
     *
     * @param  string|array $options (Optional) Minification options
     * @return Zend_Filter_Minify_Css
     */
    public function setAdapter($options = null)
    {
        if (is_string($options)) {
            $adapter = $options;
        } else if (isset($options['adapter'])) {
            $adapter = $options['adapter'];
            unset($options['adapter']);
        } else {
            $adapter = 'RegEx';
        }

        if (!is_array($options)) {
            $options = array();
        }

        if (Zend_Loader::isReadable('Zend/Filter/Minify/Css/' . ucfirst($adapter). '.php')) {
```

```
            $adapter = 'Zend_Filter_Minify_Css_' . ucfirst($adapter);
        }

        if (!class_exists($adapter)) {
            Zend_Loader::loadClass($adapter);
        }

        $this->_adapter = new $adapter($options);

        if (!$this->_adapter instanceof Zend_Filter_Minify_Interface) {
            require_once 'Zend/Filter/Minify/Exception.php';
            throw new Zend_Filter_Minify_Exception("Minification adapter '" . $adapter . "' does not
implement Zend_Filter_Minify_Interface");
        }

        return $this;
    }

    public function getOptions()
    {
     return $this->_options;
    }

    public function setOptions($options)
    {
     $this->_options = $options;

     return $this;
    }

    /**
     * Defined by Zend_Filter_Interface
     *
     * Returns a string containing the minified css. Accepts a string of css to be minified
     *
     * @param  string $value
     * @return string
     */
    public function filter($value)
    {
```

```
        return $this->_adapter->minify($value);
    }
}
```

```
require_once 'Zend/Filter/Minify/Interface.php';

class Zend_Filter_Minify_Css_RegEx implements Zend_Filter_Minify_Interface
{
 public function minify($value){
        $value = preg_replace('#\s+#', ' ', $value);
  $value = preg_replace('#/\*.*?\*/#s', '', $value);
  $value = str_replace('; ', ';', $value);
  $value = str_replace(': ', ':', $value);
  $value = str_replace(' {', '{', $value);
  $value = str_replace('{ ', '{', $value);
  $value = str_replace(', ', ',', $value);
  $value = str_replace('} ', '}', $value);
  $value = str_replace(';}', '}', $value);
        $value = trim($value);

        return $value;
    }
}
```

```
/**
 * Minification interface
 *
 * @category   Zend
 * @package    Zend_Filter
 * @copyright  Copyright (c) 2005-2010 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
interface Zend_Filter_Minify_Interface
{
 public function minify($value);
}
```

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