Zend_Image - Eric Potvin

<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_Image Component Proposal

Proposed Component Name	Zend_Image
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Image
Proposers	Eric Potvin
Revision	0.1 - July 22, 2006: First Release. (wiki revision: 11)

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_Image, Zend_Image_Validation, Zend_Image_Utility and Zend_Image_Thumbnail are the classes to help you to manipulate image for any purpose.

2. References

- ZendPHP blog

3. Component Requirements, Constraints, and Acceptance Criteria

- GD PHP library active.

4. Dependencies on Other Framework Components

- Zend_Exception

5. Theory of Operation

- Zend_Image must be instantiated.
- Zend_Image_Validation must be instantiated.
- Zend_Image_Utility does not need to me instantiated.
- Zend_Image_Thumbnail must be instantiated.

6. Milestones / Tasks

zone: Missing {zone-data:milestones}

7. Class Index

- Zend_Image
- Zend_Image_Validation
- Zend_Image_Utility
- Zend_Image_Thumbnail

8. Use Cases

UC-01

... (see good use cases book)

9. Class Skeletons

```
class Zend_Image_Exception extends Zend_Exception {}

/**
 * Image handling class.
 *
 * This class help you to manipulate an image.
 *
 * @category   Zend
 * @package    Zend_Image
 * @copyright  Copyright (c) 2006 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class Zend_Image
{
    /**
     * Constructor setup.
     */
    public function __construct() {}
```

```
    /**
     * Destructor setup.
     */
    public function __destruct() {}

    /**
     * Get the current object as string.
     * @param     string        $methodName    Method name.
     * @return     string
     */
    public function __toString(string $methodName = __METHOD__) {}

    /**
     * Call function setup.
     * @param     string        $method        Method name.
     * @param     string        $arguments    Arguments list.
     * @return     string
     */
    private function __call($method, $arguments = NULL) {}

################################################################################
#    Private function
################################################################################

    /**
     * Get the image type.
     * @return     integer
     */
    private function getImageType() {}

################################################################################
#    Public function
################################################################################

    /**
     * Set the image file to manipulate.
     * @param     string    $image    Image filename.
     */
    public function setImage(string $image) {}

    /**
     * Set the ouput quality.
     * @param     integer    $quality    Number between 0 and 100.
     */
    public function setQuality(integer $quality) {}

    /**
     * Resize an image.
     * @param     integer    $width    Number of pixel.
     * @param     integer    $height    Number of pixel.
     */
    public function resize(integer $width, integer $height) {}

    /**
     * Resize an image from his width.
     * @param     integer    $width    Number of pixel.
     */
    public function resizeToWidth(integer $newHeight) {}

    /**
     * Resize an image from his height.
     * @param     integer    $height    Number of pixel.
     */
    public function resizeToHeight(integer $newWidth) {}

    /**
     * Resize an image from a percentage.
     * @param     integer    $percentage    Number between 0 and 100.
     */
```

```
    public function resizeToPercentage(integer $percentage) {}

    /**
     * Crop an image.
     * @param     integer    $pixelXFrom    X pixel start point.
     * @param     integer    $pixelYFrom    Y pixel start point.
     * @param     integer    $pixelXTo    X pixel end point.
     * @param     integer    $pixelYTo    Y pixel end point.
     */
    public function crop(integer $pixelXFrom, integer $pixelYFrom,
                         integer $pixelXTo, integer $pixelYTo) {}

    /**
     * Convert an image.
     * @param     integer    $imageType    JPG, GIF or PNG.
     */
    public function convert(integer $imageType) {}

    /**
     * Write a text to an image.
     * @param     string    $text            Text to write.
     * @param     array    $writeProperty    Writing property
     *                                     (xPos, yPos, fontSize, color,
     *                                       fontName, angle)
     */
    public function writeText (string $text, array $writeProperty) {}

    /**
     * Save an image.
     * @param     string    $filename    New image filename.
     */
    public function save(string $filename) {}

    /**
     * Save an image.
     * @param     array    $imageProperty    Image property
     *                                     (alt, title,
     *                                       vertical space, horizontal space)
     * @return     string    HTML complete tag.
     */
    public function getHtml(array $imageProperty) {}

    /**
     * Get the header of an image.
     *
     */
    public function getHeader() {}

    /**
     * Get the height of the image.
     *
     * @return    integer    Height in pixel.
     */
    public function getHeight() {}

    /**
     * Get the width of the image.
     *
     * @return    integer    Width in pixel.
     */
    public function getWidth() {}

}


class Zend_Image_Validation_Exception extends Zend_Exception {}

/**
 * Image validation class.
 *
```

```
 * This class help you to create an "anti-spam" image for html form.
 *
 * @category   Zend
 * @package    Zend_Image
 * @copyright  Copyright (c) 2006 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class Zend_Image_Validation extends Zend_Image
{
    /**
     * Constructor setup.
     */
    public function __construct() {}

    /**
     * Destructor setup.
     */
    public function __destruct() {}

    /**
     * Get the current object as string.
     * @param     string        $methodName    Method name.
     * @return     string
     */
    public function __toString(string $methodName = __METHOD__) {}

    /**
     * Call function setup.
     * @param     string        $method        Method name.
     * @param     string        $arguments    Arguments list.
     * @return     string
     */
    private function __call($method, $arguments = NULL) {}

################################################################################
#    Public function
################################################################################

    /**
     * Set the size of the validation image.
     * @param     integer        $width        Width in pixel.
     * @param     integer        $height        Height in pixel.
     */
    public function setSize(integer $width, integer $height) {}

    /**
     * Set the secret key.
     *
     * @param     string        $key        Your secret key.
     */
    public function setSecretKey(string $key) {}

    /**
     * Set the a random secret key.
     *
     * @param     integer        $length        Length of the secret key.
     */
    public function setRandomSecretKey(integer $length) {}

    /**
     * Get the the secret key.
     *
     * @return string    The secret key.
     */
    public function getSecretKey() {}

    /**
     * Display the secret key.
     *
     */
```

```
    public function show() {}

}


class Zend_Image_Utility_Exception extends Zend_Exception {}

/**
 * Image utility class.
 *
 * This class help you to convert html code color.
 *
 * @category   Zend
 * @package    Zend_Image
 * @copyright  Copyright (c) 2006 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class Zend_Image_Utility
{
    /**
     * Convert an hex color to a decimal color.
     *
     * @param     string        $hex        Hex color.
     * @return     integer
     */
    public static function hexToDec(string $hex) {}

    /**
     * Convert an decimal color to a hex color.
     *
     * @param     integer        $dec        Decimal color.
     * @return     string
     */
    public static function decToHex(array $dec) {}

    /**
     * Convert RGB color to html color.
     *
     * @param     array        $rgb    Array of color with blue, red and green as index.
     * @return     string        html color string.
     */
    public static function RgbToHtml(array $rgb) {}

    /**
     * Convert RGB color to html color.
     *
     * @param     string        $hex    html color string.
     * @return     array        Array of color with blue, red and green as index.
     */
    public static function HexToRBBG(string $hex) {}

    /**
     * Convert RGB color to html color.
     *
     * @param     string        $hex    html color string.
     * @return     string        html color string.
     */
    public static function inverse(string $hex) {}

}


class Zend_Image_Thumbnail_Exception extends Zend_Exception {}

/**
 * Image thumbnail class.
 *
 * This class help you to create a thumbnail from a image.
 *
 * @category   Zend
```

```
 * @package    Zend_Image
 * @copyright  Copyright (c) 2006 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class Zend_Image_Thumbnail extends Zend_Image
{
    /**
     * Constructor setup.
     */
    public function __construct() {}

    /**
     * Destructor setup.
     */
    public function __destruct() {}

    /**
     * Get the current object as string.
     * @param     string        $methodName    Method name.
     * @return     string
     */
    public function __toString(string $methodName = __METHOD__) {}

    /**
     * Call function setup.
     * @param     string        $method        Method name.
     * @param     string        $arguments    Arguments list.
     * @return     string
     */
    private function __call($method, $arguments = NULL) {}

################################################################################
#    Public function
################################################################################

    /**
     * Auto-resize the picture.
     */
    public function autoThumbnail() {}

    /**
     * Resize an image.
     * @param     integer    $width    Number of pixel.
     * @param     integer    $height    Number of pixel.
     */
    public function setSize(integer $width, integer $height) {}

    /**
     * Set the ouput quality.
     * @param     integer    $quality    Number between 0 and 100.
     */
    public function setQuality(integer $quality) {}

    /**
     * Set the percentage for rezising an image.
     * @param     integer    $percentage    Number between 0 and 100.
     */
    public function setPercentage(integer $percentage) {}

    /**
     * Save the thumbnail image.
     * @param     string    $filename    Filename of the thumbnail.
     */
    public function save(string $filename) {}

    /**
     * Get the thumbnail image.
     */
```

```
    public function get() {}

}
```

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