Zend_Service_Imageshack

<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_Service_Imageshack Component Proposal

Proposed Component Name	Zend_Service_Imageshack
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Imageshack
Proposers	My E-mail Address Darby Felton, Zend liaison
Revision	0.1 - 2 May 2007: Started proposal process 0.2 - 17 May 2007: First public version available for download (wiki revision: 13)

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_Service_Imageshack is a implementation of API for Imageshack XML Services.

2. References

- Imageshack XML API request

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** allow sending files to Imageshack as well as transloading function.

4. Dependencies on Other Framework Components

- Zend_Http_Client

5. Theory of Operation

Posting files to Imageshack, validating accounts.

6. Milestones / Tasks

- Milestone 1: design notes will be published here
- Milestone 2: Working prototype checked into the incubator supporting use cases #1, #2, ...
- Milestone 3: Working prototype checked into the incubator supporting use cases #3 and #4.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

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

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

7. Class Index

- Zend_Service_Imageshack

8. Use Cases

UC 01: Validating user account

```
// just check if our registration code is ok
$shack = new Zend_Service_Imageshack('b2b745fce6edf3055805b07709ac34f5');

if (!$shack->verifyRegCode())
echo "provided key is invalid";
```

UC 02: simple file uploading

```
// upload file as anonymous (without any optimization)
$shack = new Zend_Service_Imageshack();
$shack->uploadFile(_path_to_some_image_);
print_r($shack->dumpXMLResponse());
```

UC 03: resizing image on Imageshack after uploading

```
$img = new Zend_Service_Imageshack();
// resize to 1024x768
$img->setResolution('1024x768');
// remove resolution information bar from thumbnail
$img->removeBar(true);
// upload file
$img->uploadFile(_path_to_some_image);
// display links
echo "thumb link" . {$img->links->thumb_link} . "\n";
echo "image link" . {$img->links->image_link} . "\n";
```

UC 04: transloading file

```
$img = new Zend_Service_Imageshack();
// transload file
$img->transloadFile('http://static.php.net/www.php.net/images/php.gif');
// display links
echo "thumb link" . {$img->links->thumb_link} . "\n";
echo "image link" . {$img->links->image_link} . "\n";
```

UC 05: upload movie (flv format required)

```
$img = new Zend_Service_Imageshack();
// movie resolution
$img->setResolution('320x240');
// movie duration in ms
$img->setDuration(32000);
$img->uploadFile('gierka.flv', 'th1.jpg', 'fr1.jpg');
print_r($img->dumpXMLResponse());
```

9. Class Skeletons

```
<?php
class Zend_Service_Imageshack extends Zend_Service_Abstract
{

    protected $_imageshacklogin = 'dev.imageshack.us';

    protected $_imgentrypoint = 'www.imageshack.us';

    protected $_videntrypoint = 'vid.imageshack.us';

    protected $_trnentrypoint = 'www.imageshack.us';

    /**
     * entrypoint choosen by _prepareParams
     * @var string
     */
    protected $_backend;

    /**
     * File types allowed by imageshack
     * @var array
     */
 protected $_extensions = array(
  'jpg', 'jpeg', 'png', 'gif', 'bmp', 'tif', 'tiff', 'flv'
```

```
);

   /**
    * Mime types required by Imageshack 'new' unified API
    * @var array
    */
protected $_mimetypes = array(
 'image/jpeg',
 'image/jpeg',
 'image/png',
 'image/gif',
 'image/bmp',
 'image/tiff',
 'image/tiff',
 'video/x-flv'
);

   /**
    * Optimize image or not ?
    * @var bool
    */
   protected $_optimage;

   /**
    * Imageshack login registration code
    * @var string
    */
   protected $_regCode;


   /**
    * A comma-separated list of tags to add to your video/image.
    * E.g. family,picture. Optional
    * @var string
    */
   protected $_tags;

   /**
    * Public/private marker of your video/picture. true means public (default),
    * false means private.
    * @var bool
    */
   protected $_public = true;

   /**
    * tell to ImageShack to leave (false) or remove (true) information bar on
 * thumbnail image generated by ImageShack.
    * @var bool
    */
   protected $_rembar = false;




   /**
    * uploaded video dimensions in form WxH. Mandatory for video upload
    * @var string
    */
   protected $_resolution;

   /**
    * uploaded video duration in milliseconds. Mandatory for video upload.
    * @var string
    */
   protected $_duration;



   /**
    * SimpleXML object after receiving response from imageshack
```

```
     * @var SimpleXML
     */
    protected $_simplexml;


    /**
     * raw xml response received from imageshack
     * @var string
     */
    protected $_xmlresponse = null;


    /**
     * Constructor
     *
     * @param string $regCode Imageshack Registration Login Code
     * @return void
     */
    public function __construct($regCode = "");

    /**
     * Upload file to Imageshack with given mimetype
     *
     * @param array $file
     * @return Zend_Service_Imageshack
     * @throws Zend_Service_Exception if something goes wrong
     */
    public function uploadFileAs($filearr);


    /**
     * Upload file to Imageshack
     *
     * @param string $file
     * @param string $thumbfile optional video thumbnail picture
     * @param string $frmfile optional video default frame picture
     * @return Zend_Service_Imageshack
     * @throws Zend_Service_Exception if something goes wrong
     */
    public function uploadFile($file, $thumbfile = null, $frmfile = null);


    /**
     * Transload file to Imageshack
     *
     * @param string $url
     * @return Zend_Service_Imageshack
     * @throws Zend_Service_Exception if something goes wrong
     */
    public function transloadFile($url);

    /**
     * Verify an Reg Login Code
     *
     * @param string $key Optional; Reg Code key to verify
     * @return boolean
     */
    public function verifyRegCode($regcode = null);

    protected function __get($param);

    /**
     * Dumps raw XML responase as string
     *
     * @return string
     */
    public function dumpXMLResponse();

    /**
     * Set Login Registration Code
```

```
     *
     * @param string $regCode
     * @return Zend_Service_Imageshack
     */
    public function setRegCode($regCode);

    /**
     * Retrieve API key
     *
     * @return string
     */
    public function getRegCode();

    /**
     * A comma-separated list of tags to add to your video/image.
     * E.g. family,picture. Optional
     *
     * @param string $tags
     * @return Zend_Service_Imageshack
     */
    public function addTags($tags);

    /**
     * Public/private marker of your video/picture. true means public (default),
     * false means private. Optional
     *
     * @param bool $public
     * @return Zend_Service_Imageshack
     */
    public function setPublic($public);

    /**
  * Resize image on Imageshack's side
  * [axis1]x[axis2] ie. 640x480
     * @param string $res
     * @return Zend_Service_Imageshack
     */
    public function resizeImage($res);

    /**
  * Preserve (false) or remove (true) information bar on
  * thumbnail image generated by ImageShack.
     * @param bool $public
     * @return Zend_Service_Imageshack
     */
    public function removeBar($remove);

    /**
  * Sets the duration time for video
     * @param integer $duration
     * @return Zend_Service_Imageshack
     */
public function setDuration($duration);

    /**
  * Sets the resolution of video
     * @param string $resolution
     * @return Zend_Service_Imageshack
     */
public function setResolution($resolution);

    /**
  * checks is file readable
     *
     * @param string $file
     * @throws Zend_Service_Exception if file is not readable
     */
    protected function _checkFile($file);

    /**
```

```
 * return mimetype by file extension
   *
   * @param string $file
   * @return string
   */
  protected function _getMimetypeByExtension($file);

  /**
 * returns params array used to send request to imageshack and
   * sets $this->_backend to proper host
   *
   * @param string $mimetype
   * @return array
   */
  protected function _prepareParams($mimetype);

  /**
 * Checks resolution for proper syntax
   * @param string $resolution
   * @return Zend_Service_Imageshack
   */
  protected function _checksize($res);

  protected function _parseResponse($xmlresponse);

  /**
   * Post a request
   *
   * @param string $host
   * @param string $path
   * @param array  $params
   * @return mixed
   */
```

```
    protected function _post($host, $path, array $params, $files = null);

}
```

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