Zend_Encoding

<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_Encoding Component Proposal

Proposed Component Name	Zend_Encoding
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Encoding
Proposers	Aldemar Bernal
Zend Liaison	TBD
Revision	1.0 - 21 January 2009: Initial Draft. (wiki revision: 14)

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_Encoding is a component that interacts with encoding.com api services.

2. References

- encoding.com

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** use a Zend_Config object in order to setup.
- This component **will** allow creation, update, and deletion of video encodings.
- This component **will** be able to list all the media used in encoding.com account.

4. Dependencies on Other Framework Components

- Zend_Config
- cURL

5. Theory of Operation

This is just a wrapper for the webservices api of encoding.com

6. Milestones / Tasks

- Milestone 1: [DONE] Created wiki page.
- Milestone 2: [DONE] Working prototype in laboratory.
- Milestone 3: Create unit tests.
- Milestone 4: Initial documentation exists.

7. Class Index

- Zend_Encoding

8. Use Cases

All cases uses the following config file:

```
[encoding]
id                              = ###
key                             = ################################
format.output                   = fl9
format.audio_sample_rate        = 22050
format.audio_channels_number    = 1
format.size                     = 0x240
format.logo.logo_source         = http://myurl.com/mylogo.gif
format.logo.logo_x              = 10
format.logo.logo_y              = 10
format.height                   = 240
```

UC-01
...

9. Class Skeletons

```
/**
* Zend Encoding class
* Interacts with encoding.com, requires cURL
*/
class Zend_Encoding
{
/**
* Encoding configuration
*
* @var Zend_Config
*/
protected $_config;

/**
* Internal messages
*
* @var string
*/
protected $_message;

/**
* Construct function
*
* @param Zend_Config $config
*/
function __construct(Zend_Config $config) {}

/**
* Set a new configuration file
*
* @param Zend_Config $config
*/
public function setConfig(Zend_Config $config) {}

/**
* Set a video parameter
*
* @param string $key
* @param string $value
*/
public function setParam($key, $value) {}

/**
* Returns a specific video parameter
*
* @param string $key
* @return string|null
*/
public function getParam($key) {}

/**
* Unsets a video parameter
*
* @param string $key
*/
public function unsetParam($key) {}

/**
* Internal method to send an HTTP Request
*
* @param string $xml
* @return string|null
*/
protected function sendRequest($xml) {}

/**
* Returns internal message
*
* @return string
*/
```

```
public function getMessage() {}

/**
* Returns a request xml object
*
* @param string $action
* @param string $source
* @param integer $mediaId
*/
protected function _getRequest($action, $source = null, $mediaId = null, $notifyUrl = null) {}

/**
* Adds new media to user's media list, creates new items
* in a queue according to formats specified in XML
*
* @param string $source
* @param string $notifyUrl could be either an HTTP(S) URL of the script the result
*                          would be posted to, or a mailto: link with email address
*                          the result info to be sent.
*/
public function addMedia($source, $notifyUrl = null) {}

/**
* Adds new media to user's media list and set a flag for NOT
* processing it after downloading. Format fields could be
* specified as well. If NotifyURL is set, a notification will
* be sent after the media will have been ready for processing.
*
* Note: The media will get 'Ready to process' status only if
* at least onewas specified.
*
* @param string $source
* @param string $notifyUrl could be either an HTTP(S) URL of the script the result
*                          would be posted to, or a mailto: link with email address
*                          the result info to be sent.
*/
public function addMediaBenchmark($source, $notifyUrl = null) {}

/**
* Replaces information about existing media's formats.
* All old format items will be deleted, and the new ones added.
*
* @param integer $mediaId
*/
public function updateMedia($mediaId) {}

/**
* Starts encoding of previously downloaded media
* (one that added with addMediaBenchmark function)
*
* @param integer $mediaId
*/
public function processMedia($mediaId) {}

/**
* Deletes specified media and all its items in queue
*
* @param integer $mediaId
*/
public function cancelMedia($mediaId) {}

/**
* Returns list of user's media
*
*/
public function getMediaList() {}

/**
* Returns information about selected user's media and all
* its items in queue
```

```
*
* @param integer $mediaId
*/
public function getStatus($mediaId) {}

/**
* Returns some video parameters of the specified media, if available
*
* @param integer $mediaId
```

```
 */
 public function getMediaInfo($mediaId) {}
 }
```

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