Zend_Service_Youtube - Andries Seutens

<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_Youtube Component Proposal

Proposed Component Name	Zend_Service_Youtube
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Youtube
Proposers	Andries Seutens Darby Felton, Zend liaison
Revision	1.0 - 31 January 2007: Added proposal draft (wiki revision: 10)

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_Youtube will offer a lightweight wrapper for using the Youtube APIs. It will allow you to easily integrate online videos from YouTube into your application.

2. References

- API Documentation

3. Component Requirements, Constraints, and Acceptance Criteria

- Implements all available YouTube API calls with consistent naming conventions.
- Returns server responses as lightweight data objects.

4. Dependencies on Other Framework Components

- Zend_Service_Rest
- Zend_Service_Exception
- Zend_Exception

5. Theory of Operation

Returns server responses as lightweight data objects.

6. Milestones / Tasks

- Milestone 1: write proposal
- Milestone 2: Class development
- Milestone 3: Unit tests and debugging
- Milestone 4: Documentation

7. Class Index

- Zend_Service_Youtube
- Zend_Service_Youtube_Exception
- Zend_Service_Youtube_Result
- Zend_Service_Youtube_Resultset

8. Use Cases

retrieve a YouTube user's public profile

```
<?php
require_once 'Zend/Service/Youtube.php';

$youtube = new Zend_Service_Youtube(YOUR_API_KEY);
$profile = $youtube->getProfile('andriesdev');

echo $profile->first_name;
echo $profile->last_name;
// ...
?>
```

retrieve YouTube's featured video's

```
<?php
require_once 'Zend/Service/Youtube.php';

$youtube  = new Zend_Service_Youtube(YOUR_API_KEY);
$featured = $youtube->listFeatured();

foreach ($featured as $video) {
    echo $video->description;
    // ...
    echo '<br />';
}

?>
```

For more possible queries, please see the class skeletons (Zend_Service_Youtube)

9. Class Skeletons

```
class Zend_Service_Youtube
{
    /**
     * Your Youtube developer ID
     *
     * @var string
     */
    public $devId;

    /**
  * Base URI to which API methods and parameters will be appended
  *
  * @var string
  */
    protected $_baseUri = 'http://www.youtube.com';

 /**
  * Zend_Service_Rest object
  *
  * @var Zend_Service_Rest
  */
    protected $_rest;

    /**
     * Class constructor
     *
     * Sets the developer id
     *
     * @param string $devId - Your Youtube developer Id
     */
    public function __construct($devId)
    {}

    /**
     * Retrieves the public parts of a user profile.
     *
     * @param  string $username
     * @return Zend_Service_Youtube_Result
     * @throws Zend_Service_Exception
     */
    public function getProfile($username)
    {}
```

```
/**
 * Lists a user's favorite videos.
 *
 * @param  string $username
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function listFavoriteVideos($username)
{}

/**
 * Lists a user's friends.
 *
 * @param  string $username
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function listFriends($username)
{}

/**
 * Retrieves the details for a video.
 *
 * @param  string $videoId - The ID of the video to get details for
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function getDetails($videoId)
{}

/**
 * Lists all videos that have the specified tag.
 *
 * @param  string  $tag      - The tag to search for
 * @param  integer $page     - OPTIONAL the page of results you want to retrieve (e.g. 1, 2, 3)
 * @param  integer $perPage  - OPTIONAL number of results per page (default 20, maximum 100)
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function listByTag($tag, $page = null, $perPage = null)
{}

/**
 * Lists all videos that were uploaded by the specified user
 *
 * @param string $username - User whose videos you want to list
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function listByUser($username)
{}

/**
 * Lists the most recent 25 videos that have been featured on
 * the front page of the YouTube site.
 *
 * @return Zend_Service_Youtube_ResultSet
 * @throws Zend_Service_Exception
 */
public function listFeatured()
{}

/**
 * Lists all videos that match any of the specified tags.
 *
 * @param  string  $tag      - The tag to search for
 * @param  integer $page     - OPTIONAL the page of results you want to retrieve (e.g. 1, 2, 3)
 * @param  integer $perPage  - OPTIONAL number of results per page (default 20, maximum 100)
 * @return Zend_Service_Youtube_ResultSet
```

```
     * @throws Zend_Service_Exception
     */
    public function listByRelated($tag, $page = null, $perPage = null)
    {}

    /**
     * Lists all videos in the specified playlist.
     *
     * @param string  $playlistId - The id of the playlist
     * @param integer $page       - OPTIONAL the page of results you want to retrieve (e.g. 1, 2, 3)
     * @param integer $perPage    - OPTIONAL number of results per page (default 20, maximum 100)
     * @return Zend_Service_Youtube_ResultSet
     * @throws Zend_Service_Exception
     */
    public function listByPlaylist($playlistId, $page = null, $perPage = null)
    {}

    /**
     * Lists all videos in the specified time_range.
     *
     * @param string $timeRange - The time_range to list by ('day', 'week', 'month', 'all')
     * @return Zend_Service_Youtube_ResultSet
     * @throws Zend_Service_Exception
     */
    public function listPopular($timeRange)
    {}

    /**
     * Lists videos in a particular category
     *
     * @param string  $categoryId - The category id
     * @param integer $page       - OPTIONAL the page of results you want to retrieve (e.g. 1, 2, 3)
     * @param integer $perPage    - OPTIONAL number of results per page (default 20, maximum 100)
     * @return Zend_Service_Youtube_ResultSet
     * @throws Zend_Service_Exception
     */
    public function listByCategory($categoryId, $page = null, $perPage = null)
    {}

    /**
     * Lists all videos that have the specified category id and tag.
     *
     * @param string $categoryId - The category id to search in
     * @param string $tag        - The tag to search for
     * @return Zend_Service_Youtube_ResultSet
     * @throws Zend_Service_Exception
     */
    public function listByCategoryAndTag($categoryId, $tag, $page = null, $perPage = null)
    {}

    /**
     * Prepare options for the request
     *
     * @param  string $method         - Youtube method to call
     * @param  array  $options        - User Options
     * @param  array  $defaultOptions - Default Options
     * @return array - Merged array of user and required options
     * @throws Zend_Service_Exception
     */
    protected function _prepareOptions($method, $options = array())
    {}

    /**
     * Return a single result
     *
     * @param  string $xml - Youtube xml response to parse
     * @return Zend_Service_Youtube_Result
     */
    protected function _result($xml)
    {}
```

```
    /**
     * Return a result set
     *
     * @param  string $xml - Youtube xml response to parse
     * @return Zend_Service_Youtube_ResultSet
     */
    protected function _resultSet($xml)
    {}
}



class Zend_Service_Youtube_Exception extends Zend_Exception
{}


class Zend_Service_Youtube_Result
{
    protected $_result;

    public function __construct(SimpleXMLElement $data)
    {}

    public function __get($key)
    {
       if (isset($this->_result[$key])) {
           return $this->_result[$key];
       } else {
           throw new Zend_Service_Youtube_Exception('trying to access a none existing key');
       }
    }
}


class Zend_Service_Youtube_ResultSet implements SeekableIterator
{
    /**
     * The Youtube resultset
     *
     * @var array
     */
    protected $_results;

    /**
     * Current item for the Iterator
     *
     * @var integer
     */
    private $_currentItem = 0;

    /**
     * Parse the Youtube result set
     *
     * @param SimpleXMLElement $data - The Youtube result set
     */
    public function __construct(SimpleXMLElement $data)
    {}

    /**
     * Implement SeekableIterator::current
     *
     * Return the current element.
     *
     * @return Zend_Service_Youtube_Result
     */
    public function current()
    {}

    /**
```

```
     * Implement SeekableIterator::key
     *
     * Return the key of the current element.
     *
     * @return int
     */
    public function key()
    {}

    /**
     * Implement SeekableIterator::next
     *
     * Moves forward to next element.
     */
    public function next()
    {}

    /**
     * Implement SeekableIterator::rewind
     *
     * Rewind the Iterator to the first element.
     *
     * @return boolean
     */
    public function rewind()
    {}

    /**
     * Implement SeekableIterator::seek
     *
     * Seek to an absolute position.
     *
     * @param  integer $item
     * @return Zend_Service_Youtube_Result
     * @throws Zend_Service_Exception
     */
    public function seek($item)
    {}

    /**
     * Implement SeekableIterator::valid
     *
     * @param  integer $item
     * @return boolean
     */
```

```
    public function valid($item = null)
    {}
}
```

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