Zend_Service_Audioscrobbler Proposal - Chris Hartjes, Derek Martin

<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_Audioscrobbler Component Proposal

Proposed Component Name	Zend_Service_Audioscrobbler
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Audioscrobbler
Proposers	Chris Hartjes Derek Martin
Revision	1.0 - 1 July 2006 (wiki revision: 21)

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_Audioscrobbler provides a simple API to access Audioscrobbler's web services

2. References

- Audioscrobbler Web Services
- Last.fm

3. Component Requirements, Constraints, and Acceptance Criteria

- A simple API for accessing Audioscrobbler's web services

4. Dependencies on Other Framework Components

- Zend_Service_Rest
- Zend_Service_Exception

5. Theory of Operation

Results from Audioscrobbler's REST services are returned as SimpleXML objects, otherwise a Zend_Service_Exception is thrown containing an informative error message.

6. Milestones / Tasks

alpha 0.01 released November 15th, 2006

7. Class Index

- Zend_Service_Audioscrobbler

8. Use Cases

UC-01

```
$as = new Zend_Service_Audioscrobbler();
$as->set('user', 'BigDaddy71);
$profile_info = $as->userGetProfileInformation();

print "Pulling information on {$profile_info->realname}<br />";
print "User info can be found at {$profile_info->url}<br />";
```

UC-02

```
$as = new Zend_Service_Audioscrobbler();
$as->set('artist', 'LCD Soundsystem');
$related_artists = $as->artistGetRelatedArtists();

foreach ($related_artists as $artist)
{
    print "<a href={$artist->url}>{$artist->name}</a><br />";
}
```

UC-03

```
$as = new Zend_Service_Audioscrobbler();
$as->set('user', 'lo_fye');
$from = '1114965332';
$to = '1115570132';
$previous_weekly_artists = $as->userGetWeeklyArtistChart($from, $to);

foreach ($previous_weekly_artists as $artist)
{
    print "<a href={$artist->url}>{$artist->name}</a><br />";
}
```

9. Class Skeletons

Zend_Service_Audioscrobbler __construct ()
void albumGetInfo ()
SimpleXML artistGetRelatedArtists ()
SimpleXML artistGetTopAlbums ()
SimpleXML artistGetTopFans ()
SimpleXML artistGetTopTags ()
SimpleXML artistGetTopTracks ()
void get ($field, string $key)
void group ()
void groupGetWeeklyArtistChartList ()
void groupGetWeeklyChartList ()
void set (string $field, $value, string $val)
void tagGetTopArtists ()
void tagGetTopTags ()
void tagGetTopTracks ()
void trackGetTopFans ()
void trackGetTopTags ()
SimpleXML userGetFriends ()
SimpleXML userGetNeighbours ()
array userGetProfileInformation ()
SimpleXML userGetRecentBannedTracks ()
SimpleXML userGetRecentLovedTracks ()
SimpleXML userGetRecentTracks ()
SimpleXML userGetTopAlbums ()
array userGetTopArtists ()
SimpleXML userGetTopTags ()
SimpleXML userGetTopTagsForAlbum ()
SimpleXML userGetTopTagsForArtist ()
SimpleXML userGetTopTagsForTrack ()
SimpleXML userGetTopTracks ()
SimpleXML userGetWeeklyAlbumChart ($from, $to)
SimpleXML userGetWeeklyArtistChart ($from, $to)
SimpleXML userGetWeeklyChartList ()
SimpleXML userGetWeeklyTrackChart ($from, $to)

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

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