Zend_Service_Geonames

<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_Geonames Component Proposal

Proposed Component Name	Zend_Service_Geonames
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Geonames
Proposers	Nick Daugherty Boris Guéry
Zend Liaison	TBD
Revision	1.0 - 4 November 2010 - Initial Draft 1.1 - 28 November 2010 - Completed proposal (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_Geonames aims to provide full access to the Geonames.org API. Geonames.org provides many free geographic services such as POI lookups, Postal Code lookups, and regular/reverse geocoding.

Current work is available on GitHub ~~Bgy_Service_Geonames~~ Now part of my extension library to Zend Framework Bgy_Service_Geonames

2. References

- Geonames.org
- GeoNames Web Services Overview
- GeoNames Web Services Documentation
- GeoNames Web Services Exception

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** provide full access to the Geonames.org API

- This component **will** abstract the interaction with Geonames.org's database and XML/Json responses
- This component **will** provide methods to return Wikipedia documents based on various geographic queries
- This component **will** provide programmatic access to Geonames.org's nearly 7 million POI's
- This component **will** handle Geonames.org web services errors and throw a Zend_Service_Geonames_Exception accordingly
- This component **should** handle failover servers
- This component **will not** manage web-service using other output than XML or JSON

4. Dependencies on Other Framework Components

- Zend_Rest
- Zend_Json
- Zend_Exception

5. Theory of Operation

6. Milestones / Tasks

- Milestone 1: Initial proposal
- Milestone 2: Class overview and method signature
- Milestone 3: Use cases
- Milestone 4: Working prototype
- Milestone 5: Acceptance
- Milestone 6: Unit tests exist, work
- Milestone 7: Initial documentation exists

7. Class Index

- Zend_Service_Geonames
- Zend_Service_Geonames_Exception

8. Use Cases

9. Class Skeletons

```
class Zend_Service_Geonames
{
    const API_URI = 'http://ws.geonames.org';

    /**
     * Supported methods
     * Describe prefered output type and root property/node
     * to format the result in a user-friendly manner
     *
     * @var array
```

```
     */
    protected static $_availableMethods = array(
        'astergdem' => array(
            'output' => 'json',
        ),
        'children' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'cities' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'countryCode' => array(
            'output' => 'json',
        ),
        'countryInfo' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'countrySubdivision' => array(
            'output' => 'json',
        ),
        'earthquakes' => array(
            'output' => 'json',
            'root'   => 'earthquakes',
        ),
        'extendedFindNearby' => array(
            'output' => 'xml',
        ),
        'findNearby' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'findNearbyPlaceName' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'findNearbyPostalCodes' => array(
            'output' => 'json',
            'root'   => 'postalCodes',
        ),
        'findNearbyStreets' => array(
            'output' => 'json',
            'root'   => 'streetSegment',
        ),
        'findNearbyStreetsOSM' => array(
            'output' => 'json',
            'root'   => 'streetSegment',
        ),
        'findNearByWeather' => array(
            'output' => 'json',
            'root'   => 'weatherObservation',
        ),
        'findNearByWikipedia' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'findNearestAddress' => array(
            'output' => 'json',
            'root'   => 'address',
        ),
        'findNearestIntersection' => array(
            'output' => 'json',
            'root'   => 'intersection',
        ),
        'findNearestIntersectionOSM' => array(
            'output' => 'json',
            'root'   => 'intersection',
        ),
```

```
        'get' => array(
            'output' => 'json',
        ),
        'gtopo30' => array(
            'output' => 'json',
        ),
        'hierarchy' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'neighbourhoud' => array(
            'output' => 'json',
            'root'   => 'neighbourhood',
        ),
        'neighbours' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'postalCodeCountryInfo' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'postalCodeLookup' => array(
            'output' => 'json',
            'root'   => 'postalcodes',
        ),
        'postalCodeSearch' => array(
            'output' => 'json',
            'root'   => 'postalCodes',
        ),
        'search' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'siblings' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'srtm3' => array(
            'output' => 'json',
        ),
        'timezone' => array(
            'output' => 'json',
        ),
        'weather' => array(
            'output' => 'json',
            'root'   => 'weatherObservations',
        ),
        'weatherIcao' => array(
            'output' => 'json',
            'root'   => 'weatherObservation',
        ),
        'wikipediaBoundingBox' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
        'wikipediaSearch' => array(
            'output' => 'json',
            'root'   => 'geonames',
        ),
    );

    /**
     * Username
     *
     * @var string
     */
    protected $_username;

    /**
```

```
     * Token
     *
     * @var string
     */
    protected $_token;

    /**
     * Zend_Rest_Client instance
     *
     * @var Zend_Rest_Client
     */
    protected $_rest = null;

     /**
     * Options passed to constructor
     *
     * @var array
     */
    protected $_options = array();

    /**
     * Construct a new Geonames.org web service
     *
     * @param array $options
     * @return void
     */
    public function __construct($options = null) {}


    /**
     * Set username
     *
     * @param  string $username
     * @return Zend_Service_Geonames Provides a fluent interface
     */
    public function setUsername($username) {}

    /**
     * Get username
     *
     * @return string
     */
    public function getUsername() {}


    /**
     * Set token
     *
     * @param  string $token
     * @return Zend_Service_Geonames Provides a fluent interface
     */
    public function setToken($token) {}

    /**
     * Get token
     *
     * @return string
     */
    public function getToken() {}

    /**
     * Retrieve all the supported methods
     *
     * @return array Available methods
     */
    public static function getAvailableMethods() {}

    /**
     * Method overloading which checks for supported methods
     *
```

```
     * @param string $method The webservice method
     * @param array $params The parameters
     * @throws Zend_Service_Geonames_Exception
     * @return array
     */
    public function __call($method, $params = array()) {}

    /**
     * Handles all GET requests to a web service
     *
     * @param   string $method Requested API method
     * @param   array  $params Array of GET parameters
     * @return  mixed  decoded response from web service
     * @throws  Zend_Service_Geonames_Exception
     */

    public function makeRequest($method, $params = array()) {}

     /**
     * Evaluates result
     *
     * @param   Array $result
     * @return  void
     * @throws  Zend_Service_Geonames_Exception
```

```
     */
    private function _evalResult($result) {}
}
```

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