Zend_Service_Currency

<ac:macro ac:name="info"><ac:parameter ac:name="title">New Proposal Template</ac:parameter><ac:rich-text-body>
This page has been created from a template that uses "zones." To proceed:

Edit the page
Replace sample content within each zone-data tag
Remove this notice
Save the page
When you are ready for review, remove the <ac:emoticon ac:name="warning" /> Under Construction notice

</ac:rich-text-body></ac:macro>

<ac:macro ac:name="note"><ac:parameter ac:name="title">Under Construction</ac:parameter><ac:rich-text-body>
This proposal is under construction and is not ready for review. </ac:rich-text-body></ac:macro>

<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_Currency Component Proposal

Proposed Component Name	Zend_Service_Currency
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Currency
Proposers	André Hoffmann Pádraic Brady Darby Felton, Zend liaison
Revision	1.0 - not yet: created proposal (wiki revision: 8)

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

The purpose of this class is to utilise one or more FX rate sources to perform FX translations or return spot FX rates between any two currencies. Currencies will be denoted with Country Codes compliant with ISO 4217.

2. References

- XMethods WSDL
- [Pear::Service_ExchangeRates]

3. Component Requirements, Constraints, and Acceptance Criteria

- This component will fetch spot FX rates from a selection of possible services incl. at a minumum ECB (European Central Bank) and FRB (US Federal Reserve) published rates.
- This component will validate Country Currency codes correctly against a) Zend_Currency or b) the UNECE published list.
- This component will perform FX translations between any two currencies.
- This component will return correct rate for translation between any two currencies.
- This component will correctly handle locale issues such as comma/period differences from en_UK or en_US and mainland Europe.
- This component may save data for a specified time using Zend_Cache.
- This component may include a factory method.
- This component will allow subclassing for alternative rate sources from country specific institutions.

4. Dependencies on Other Framework Components

- Zend_Service_Exception
- Zend_Service_Rest
- Zend_Locale
- Zend_Currency
- Zend_Cache

5. Theory of Operation

Based on any two compliant ISO 4217 Country Currency codes, Zend_Service_Currency would fetch the relevant exchange rate from a selected source, or range of sources, and either:

a) perform a currency translation and return the result; or
b) return the relevant exchange rate

6. Milestones / Tasks

- Milestone 1: Publish relevant design notes and background on design decisions
- Milestone 2: Write Unit Tests for Use Case #1 and check implementing code into Incubator
- Milestone 3: Write Unit Tests/Code for remaining Use Cases and check implementing code into Incubator
- Milestone 4: Write documentation sufficient for core Use Cases
- Milestone 5: Perform review and additional refactoring. Assess additional Use Cases (if any).

7. Class Index

- Zend_Service_ExchangeRates_Interface
- Zend_Service_ExchangeRates
- Zend_Service_ExchangeRates_ECB
- Zend_Service_ExchangeRates_FRB
- Zend_Service_ExchangeRates_XMethods
- Zend_Service_ExchangeRates_CountryCodes_Interface
- Zend_Service_ExchangeRates_CountryCodes_UNECE

8. Use Cases

UC-01

```
/* en_UK query (no Zend_Currency use)
 *
 * Simple use case fetching the FX rate for translating between two currencies.
 */

// Fetch instance based on source (European Central Bank) from static
// Factory method
$fx = Zend_Service_ExchangeRates::factory('ECB');

// set To and From currencies (ISO 4217 Currency Codes)
$fx->setFromCurrency('EUR');
$fx->setToCurrency('USD');

// return the rate...
echo $fx->getFXRate();
```

9. Class Skeletons

```
class Zend_Service_ExchangeRates {

    protected $source = '';

    protected $toCode = '';

    protected $fromCode = '';

    protected $options = array();

    protected $sourceURL = '';

    protected $fxRate = 1;

    private function __construct();

    public static function factory($source, $options);

    public function setFromCurrency($fromCode);

    public function setToCurrency($toCode);

    public function translate($amount);

    public function getFormula();

    private function isValidCurrency();

    private function format();

    public abstract function getFXRate();

    public abstract function getAllRates();

}
```

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