Zend Framework: making using web services easier

Karen A. Coombs
Product Manager
OCLC Developer Network

Friday, May 6, 2011
Lots of Useful Data for Libraries in Web Services

- Book covers
- Reviews
- Ratings
- Additional Bibliographic data
- Media
Using Web Services Can Be Hard

- Every API is different
- Lots of parameters to deal with
- Parsing XML output is tricky
- Authentication can be a pain
Basic Web Service Functions In Zend

- JSON
- OAuth
- REST Client
- HTTP Client
- Feed
REST Client

- Interact with RESTful web services
  - `get()`, `post()`
- Set parameters in request
- Access data returned via DOM
- Doesn’t handle complex namespaces
- Doesn’t handle accept headers - use HTTP Client if want this
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Rest_Client');
$isbn = $_REQUEST['isbn'];

$isbn_url = 'http://xisbn.worldcat.org/webservices/xid/isbn/' . $isbn;
$classify = new Zend_Rest_Client($isbn_url);
 $classify->method('getMetadata');
 $classify->format('xml');
 $classify->fl('*');
 $result = $classify->get();

foreach($result->isbn as $item ) {
    echo $item['title'];
}
HTTP Client

- Interact with RESTful web services
  - GET, POST, PUT, DELETE
- Handles setting accept headers
- Can pass the information retrieved off to SimpleXML
- Send several requests with a single client
Feed

- Easily consume Atom or RSS feeds
- or a single Atom entry
- Output its various parts
- Create or modify feeds
- Knows Dublin Core, Syndication, Creative Commons namespaces
- Extend it to consume other namespaces
Web Services & Zend

- Amazon Advertising API
- Flickr
- Delicious
- Twitter
- Several URL shorteners
- Google API
Amazon API

★ Search for Items in Amazon
★ Looking up a Specific Amazon Item
★ Info Returned
★ Lists
★ Similar Products
★ Accessories
★ Reviews: Customer and Editorial Reviews
Flickr API

- Search for photos
  - group, user
- Search for users
- Retrieve Image Details
  - id, title, license, different image version
Delicious

- Retrieve posts
  - filter based on tag
- Retrieve post information
  - Title, url
- Add, Edit and Delete Posts
- Retrieve posts tags and bundles
Twitter

- Search Twitter
- Show timelines
  - User, friends, public
- Show user info
  - friends, followers, status, replies
- Add/Update
  - friends, favorites, direct messages
URL Shorteners

- URL Shortening for several services including TinyUrl
- Shorten url
- Unshorten url
Google APIs

- Book Search Data API
- Calendar
- Documents
- Spreadsheets
- Picasa
- YouTube
Extending a Zend Class

- Zend_Feed_Reader
  - Processes RSS/Atom feeds
- WorldCat Search API OpenSearch returns these
  - but it takes parameters
- Want to extend class to handle parameters
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Feed_Reader');
define ('WORLDCAT_URL', 'http://worldcat.org/webservices/catalog/search/opensearch');
class Zend_Feed_WorldCat extends Zend_Feed_Reader {
    public static function import ($query, $format, $wskey) {
        $query = urlencode($query);
        return parent::import(WORLDCAT_URL . '?q=' . $query . '&format=' . $format . '&wskey=' . $wskey);
    }
}
What if your services don’t return Atom/RSS?

Extend REST Client Class!

This is how most of the Zend Service classes are built

Two parts

Build the query or queries

Process the response(s)
ADVANTAGES OF A CLASS

- Makes development easier
- Abstracting requests to the web service
- Removing xPath data selection complexity
- Makes common tasks simpler to perform
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Service_Amazon');

$AWSAccessKeyId = '1234';
$SecretAccessKey = '1234';

$amazon = new Zend_Service_Amazon($AWSAccessKeyId, 'US', $SecretAccessKey);

$item = $amazon->itemLookup('0451223276', array(
    'IdType' => 'ISBN',
    'MerchantId' => 'Amazon',
    'SearchIndex' => 'Books',
    'ResponseGroup' => 'Medium,Offers',
    'Condition' => 'New'
));

if (count($item->Offers->Offers) > 0) {
    echo '<p>'. $item->Title .'('</p>,'
    echo '<p>'. $item->ASIN .'('</p>,'
    echo '<p>'. ($item->Offers->Offers[0]->Price * .01) .'('</p>,'
    echo '<p>'. $item->Offers->Offers[0]->Availability .'('</p>,'
} else {
    echo '<p>Sorry this item is not available from Amazon.';
    echo '<p>Please check to see if this is available from another seller via the Amazon site.</p>'';