Consuming Web Services
Using PHP 5

Adam Trachtenberg
Senior Manager of Platform Evangelism, eBay
Plan of Attack

REST

SOAP
Weapons

- PHP 5
- SimpleXML
- Streams
- ext/soap
REST

URL + HTTP Verb -> XML Reply
Thai Food near 94306

http://api.example.com/yellowpages?zip=94306&query=thai+restaurant

<ResultSet>
  <Result>
    <Title>Thai City</Title>
    <Address>3691 El Camino Real</Address>
    <City>Palo Alto</City>
    <State>CA</State>
    <Phone>(650) 493-0643</Phone>
    <BusinessUrl>http://www.thaicity.com/</BusinessUrl>
  </Result>
  <Result>
    <Title>Thai Garden Restaurant</Title>
    <Address>4329 El Camino Real</Address>
    ...
  </Result>
</ResultSet>
# REST (In Theory)

<table>
<thead>
<tr>
<th>SQL</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>POST</td>
</tr>
<tr>
<td>SELECT</td>
<td>GET</td>
</tr>
<tr>
<td>UPDATE</td>
<td>PUT</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
REST (In Practice)

<table>
<thead>
<tr>
<th>SQL</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>GET</td>
</tr>
<tr>
<td>SELECT</td>
<td>GET</td>
</tr>
<tr>
<td>UPDATE</td>
<td>GET</td>
</tr>
<tr>
<td>DELETE</td>
<td>GET</td>
</tr>
</tbody>
</table>
del.icio.us

• Social bookmarking site
• Save pages for yourself
• Share pages with others
• Provide meta-data via folksonomy
• Easy to insert and query the site
del.icio.us/rss

- RSS feeds for everything
- Prepend rss before path
- http://del.icio.us/popular/ebay
- http://del.icio.us/rss/popular/ebay
<rdf:RDF>
    <channel rdf:about="http://del.icio.us/popular/ebay">...
    </channel>
    <item rdf:about="http://the-winning-bid.com/">
        <title>The-Winning-Bid.com</title>
        <link>http://the-winning-bid.com/</link>
        <dc:creator>minenet</dc:creator>
        <dc:date>2005-10-10T03:40:09Z</dc:date>
    </item>
    
    <item rdf:about="http://ebaysupplies.usps.com/">
        <title>USPS | eBay</title>
        <link>http://ebaysupplies.usps.com/</link>
        <dc:creator>kresston</dc:creator>
        <dc:date>2004-11-17T14:51:34Z</dc:date>
    </item>
    
    ...
</rdf:RDF>
RSS + SimpleXML

$url = 'http://del.icio.us/rss/popular/ebay';

$xml = simplexml_load_file($url);

foreach ($xml->item as $item) {
    printf("%20.20s : %-30.30s\n", (string) $item->title, (string) $item->link);
}
Popular eBay Pages

The-Winning-Bid.com : http://64.34.178.175/results.p
USPS | eBay : http://ebaysupplies.usps.com/
GarageSale : http://www.iwascoding.com/Gara
:: gumshoo - eBay se : http://www.gumshoo.com/
mapBid - View eBay a : http://www.mapbid.com/
Auction Mapper | Vis : http://www.auctionmapper.com/
ebay api developer d : http://developer.ebay.com/
what is it? : http://www.whatis-it.com/
eBay Developer Chall : http://ebay.promotionexpert.co
ebay + google maps : http://www.markovic.com/markov
eBay drops charges f : http://www.cbronline.com/artic
Mark Pincus Blog: Sh : http://markpincus.typepad.com/
foreach (simplexml_load_file('http://del.icio.us/rss/popular/ebay')-&gt;item as $item) {
    printf("%20.20s : %-30.30s\n",
            (string) $item-&gt;title, (string) $item-&gt;link);
}
REST + GET

- URL + query string
- Test in Firefox
- Still process XML
flickr

- Social digital photo site
- Save photos for yourself
- Share photos with others
- Provide meta-data via folksonomy
- Easy to insert and query the site
flickr/services/rest

- Multiple REST API calls
- API decoupled from site URL structure
- flickr.com/services/rest/?method=...
flickr + GET

- Requires developer authentication token
- Search flickr

http://www.flickr.com/services/api/
  flickr.photos.search.html

http://www.flickr.com/services/rest/?
  method=flickr.photos.search&
  tags=ebay&
  api_key=giantlonguglystring
<xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
  <photos page="1" pages="75" perpage="100" total="7422">
    <photo id="71550241" owner="27503448@N00" secret="51fb62fb95" server="35" title="champ1" ispublic="1" isfriend="0" isfamily="0"/>
    ...
  </photos>
</rsp>
$base = 'http://www.flickr.com/services/rest/?';
$qs = array(
    'method' => 'flickr.photos.search',
    'api_key' => 'biglonguglystring',
    'tags' => 'ebay',
);

$url = $base . http_build_query($qs);
$out = file_get_contents($url);
Create a Gallery

```php
$xml = simplexml_load_string($out);
foreach ($xml->photos->photo as $photo) {
    $server = (string) $photo['server'];
    $id = (string) $photo['id'];
    $secret = (string) $photo['secret'];
            \${server}/\${id}_\${secret}_s.jpg"/>";
}

print $img;
```
REST + POST

- URL + POST Data
- Can’t test in Firefox
- POST body can be anything
- Return data is XML
flickr + POST

• Requires developer authentication token
• Requires user authentication token
• Add tags to a photo
• http://www.flickr.com/services/api/flickr.photos.addTags.html
Define POST Data

$url =
    'http://www.flickr.com/services/rest/?';
$qs = array(
    'method' => 'flickr.photos.addTags',
    'api_key' => 'biglonguglystring',
    'auth_token' => 'anotherbiglonguglystring',
    'tags' => 'hongkong',
    'photo_id' => '50021321'
);

Compute the Signature

```php
$api_sig = 'would_you_believe_another_long_string?';

ksort($qs);
foreach ($qs as $k => $v) {
    $api_sig .= $k . $v;
}

$qs['api_sig'] = md5($api_sig);
```
POST a Request

$content = http_build_query($qs);
$options = array(
    'http' => array(
        'method' => 'POST',
        'content' => $content
    )
);

$context = stream_context_create($options);
$out = file_get_contents($url, false, $context);
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">

</rsp>
SOAP

- Leaky abstraction around XML
- Mapped conversion of native data types to XML Schema types and vice versa
- Independent of HTTP
- API described using WSDL
Use ext/soap

- Bundled with PHP 5
- Enabled by default in PHP 5.1
- Written in C not PHP
- Most compatible with other SOAP servers
- Actively maintained
eBay

- Social marketplace site
- Buy items for yourself
- Sell items to others
- Provide meta-data via attributes
- Easy to insert and query the site
api.ebay.com

- Totally decoupled from eBay
- SOAP interface
- Requires developer and user authentication
- Testing and production environments
// Create and configure session
$session = new eBaySession('long_string', 'another_long_string', 'ya_long_string');
$session->token = 'one_more_long_string';
$session->site = 100; // 100 = Motors;
$session->location = https://api.ebay.com/wsapi;
try {
    $client = new eBaySOAP($session);
    $params = array(
        'Version' => 435,
        'Query' => '*',
        'CategoryID' => 6001); // Cars
    $results =
        $client->GetSearchResults($params);
} catch (SOAPFault $f) {
}
<?xml version="1.0" encoding="UTF-8"?>
xmlns:ns1="urn:ebay:apis:eBLBaseComponents">
  <SOAP-ENV:Header>
    <ns1:RequesterCredentials>
      <ns1:eBayAuthToken>one_more_long_string</ns1:eBayAuthToken>
      <ns1:Credentials>
        <ns1:AppId>one_long_string</ns1:AppId>
        <ns1:DevId>another_long_string</ns1:DevId>
        <ns1:AuthCert>ya_long_string</ns1:AuthCert>
      </ns1:Credentials>
    </ns1:RequesterCredentials>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns1:GetSearchResultsRequest>
      <ns1:Version>425</ns1:Version>
      <ns1:Query>*</ns1:Query>
      <ns1:CategoryID>6001</ns1:CategoryID>
      <ns1:TotalOnly>true</ns1:TotalOnly>
    </ns1:GetSearchResultsRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
$total = number_format(
    $results->PaginationResult
    ->TotalNumberOfEntries);

print "There are {\$total} passenger vehicles for sale on eBay Motors";
eBay Motors Google Maps
What I’ve learned

• Web services are closed source software
• Documentation and online support is vital
• Debugging is hard
• SOAP sucks! SOAP rocks!
• SOAP interoperability is an issue
• Authentication is ad-hoc
Tips and Tricks

- You’re not querying a local MySQL database
- Cache your data
- Use debugging methods
- Sniff the wire
- Send requests to your own server
It Only Looks Simple

- Web services == (HTTP && XML) != PHP
- You must grok
  - HTTP
  - XML
  - XML Namespaces (Danger!)
  - XML Schema
  - XPath (I <heart> XPath)
  - SOAP
- The data is layered
What Shape Is Your Data?

- **REST**
  - Angled = <> = XML Database
  - FLOWR: /Items/Item[Price < 100]/ItemID

- **SOAP**
  - Square = [] = Relational database
  - SQL: SELECT ItemID FROM Items WHERE Items.Price < 100
References

- del.icio.us: http://del.icio.us/help/
  - RSS, HTML, Javascript, JSON, IE Active Channel, API (REST + GET)

- flickr: http://www.flickr.com/services/api/
  - RSS, Atom, REST, XML-RPC, SOAP

- eBay: http://developer.ebay.com
  - RSS, REST, XML API, SOAP, .NET SDK, Java SDK
Questions?

http://www.trachtenberg.com/talks/apachecon2005