# Web Services Bootcamp: Adding Value to Library Apps & Services

Jason A. Clark
Head of Digital Access and Web Services
Montana State University Libraries

# A quick look ahead…

- An overview of the different web services protocols
- Wading through the acronym soup and major terms associated with web services
- A discussion of the benefits of web services for libraries
- Walkthrough of the code that makes it happen

# A quick look ahead…

- Why web services for libraries?
- What do we gain in the move to "mashup"?
- Define the major terms of web services
- Resources for learning: wizard and tools (Yahoo Pipes, Google Code Playground)
- Code samples for downloading and practicing

# Questions?

- Ask anytime during the presentation
- This can be heady stuff
- twitter.com/jaclark or email

# Web as Platform

- Tim O'Reilly's concept for Web 2.0

http://oreilly.com/web2/archive/what-is-web-20.html

- software written above level of single device

- lightweight programming models

- small pieces, loosely joined

# Web as Platform

- Examples
  - go2collegemt.org
  - wildlifenearyou.com

# Terms: API

What is an API?

An application programming interface (or API) is a way for developers to access parts of a remote web site and integrate it with their own site.

MSU Libraries "lofiAPI" Example

http://www.lib.montana.edu/~jason/files/api/lofi/

# Terms: Web Service

What is a Web Service?

- Broader term
- Public interface (API)
- Provides access to data and/or procedures
- On a remote/external system (usually)
- Use structured data for data exchange (often XML)

# Terms: Structured Data

Structured data = XML and JSON

- Extensible Mark-up Language and Javascript Object Notation
- Flexible mark-up languages
- Lightweight and easy to parse
- Allow communication between disparate systems

# Terms: POST and GET

Two primary verbs for web services actions

- POST data to a web service
- GET data from a web service
- Read and Write actions

# Why use Web Services?

- Access to content/data stores you could not otherwise provide (zip codes, news, pictures, reviews, etc.)
- Enhance site with a service that is not feasible for you to provide (maps, search, products, etc.)
- Combine these services into a seamless service you provide (mash-ups)

# Provide Web Services?

- You have a service that benefits your users best if they can get to their data from outside the application
- You want others to use your data store in their applications

# Available Web Services

- Google
- Yahoo!
- Amazon
- eBay
- Flickr
- del.icio.us
- Google App Engine http://code.google.com/appengine/
- Amazon s3
- iTunes
- YouTube
- Many more...

# You'd be surprised...

- AllCDCovers.com http://www.allcdcovers.com/api
- ISBNdb.com http://isbndb.com/docs/api/index.html
- OpenDOAR http://www.opendoar.org/tools/api.html
- arXiv.org http://export.arxiv.org/api_help/
- Google Book Search APIs - http:code.google.com/apis/books
- LibraryThing APIs - http://www.librarything.com/services
- WorldCat Search API - http://www.worldcat.org/devet/wiki/SearchAPIDetails
- Open Library API - http://openlibrary.org/dev/docs/api

* See ProgrammableWeb http://www.programmableweb.com/apis/directory

# Types of Web Services

- SOAP
- XML-RPC
- REST

# What is SOAP?

- An acronym for Simple Object Access Protocol
- Version 1.2 of the W3C recommendation
- dropped the acronym
- Specification maintained at w3.org
- There's nothing simple about SOAP!

# Using SOAP

- Send a message specifying an action to take, including data for the action
- Receive a return value from the action
- Most SOAP services provide a WSDL file to describe the actions provided by the service

# What's WSDL?

- Web Services Description Language
- XML mark-up for describing the functionality provided by a SOAP service

# SOAP Example

EBAY wsdl

http://api.google.com/GoogleSearch.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.
org/soap/envelope/"
xmlns:ns1="urn:ebay:apis:eBLBaseComponents">
<SOAP-ENV:Header>
...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<ns1:GetSearchResultsRequest>
<ns1:Version>425</ns1:Version>
<ns1:Query>*</ns1:Query>
<ns1:TotalOnly>true</ns1:TotalOnly>
</ns1:GetSearchResultsRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# SOAP: Final Thoughts

- Complex
- Messaging and Data mingled
- Usually seen in software APIs, but many scripting languages have libraries
- Google API has moved away from it

# What is XML-RPC

- XML Remote Procedure Call
- Specification maintained at xmlrpc.com
- Provides a means to call methods/procedures on a remote server and make changes and/or retrieve data
- An early specification

# Using XML-RPC

- Most common implementation of XML-RPC used today is that of blog ping services
- Technorati, Flickr, FeedBurner, others?

# XML-RPC: Final Thoughts

- An updating protocol
- Early adoption, but little recent development

# What is REST?

- The greatest thing since sliced...
- Representational State Transfer
- Unique data resources with addresses

# Theory of REST

- Focus on diversity of resources (nouns), not actions (verbs)
- Every resource is uniquely addressable
- All resources share the same constrained interface for transfer of state (actions)
- Must be stateless, cacheable, and layered

# REST = Web Protocol

Web As Prime Example

- URLs uniquely address resources
- HTTP methods (GET, POST, HEAD, etc.) and content types provide a constrained interface
- All transactions are atomic
- HTTP provides cache control

# REST: Final Thoughts

- Similarity to web - easy to understand
- URL is the method
- Most popular type of web service

# Formats for Data from Web Services

- XML
  - Lots of different formats
  - Can use a particular standard
    - MARC XML
    - Dublin Core
    - RSS
    - Atom
  - Or may be a proprietary format
- JSON (Javascript Object Notation)
  - very popular
  - easy to use with Javascript
  - can be simpler to work with
- HTML

# Web Services in Libraries

- Plymouth State: Scriblio
- Repository66: mash-up of OpenDOAR data with Google Maps and repository growth charts from ROAR, developed by Stuart Lewis of the University of Aberystwyth, Wales http://maps.repository66.org/
- lofiAPI: MSU Libraries (ETD, RMT)
- MSU Library Lifestream: RSS services (Twitter, del.icio.us, last.fm, MSU Library Blog)
- TERRApod Youtube and blip.tv admin

# Web Services in Libraries

- Web Services from OCLC
  - WorldCat Search API
  - xISBN
  - xISSN
  - WorldCat Registry (Registry Search and Registry Detail)
  - WorldCat Identities
  - Virtual International Authority File
  - Terminology Services
- LibraryThing API
- Google Book API
- Open Library API

# Under the hood...

Making the examples work... a closer look at the web services handout.

Which examples do you want to talk about?

- Yahoo Pipes
- Google Code Playground
- Basic API examples

# What I've Learned

- Web services are closed source software
- Documentation and online support is vital
- Debugging can be hard
- Similarities to common protocols are important
- Practice and finding your development kit is essential

# Last thoughts...

- This stuff is just beginning…
- Digital Library Federation API recommendation
- Library mashups are here - WorldCat widget for Wordpress

# Contact Information

**Jason A. Clark**
Head of Digital Access and Web Services
Montana State University Libraries
jaclark@montana.edu
www.jasonclark.info
twitter.com/jaclark
406-994-6801