

# **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...

- 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](https://twitter.com/jaclark) or email

# Web as Platform

- Tim O'Reilly's concept for Web 2.0

[oreilly.com/web2/archive/what-is-web-20.html](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

- [www.lib.montana.edu/channel/](http://www.lib.montana.edu/channel/)

- [www.lib.montana.edu/beta/bookme/](http://www.lib.montana.edu/beta/bookme/)

# 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" [www.lib.montana.edu/~jason/files/api/lofi/](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: Mashup

What is a mashup?

A Web application that uses and/or combines data from multiple sources within a single tool. Trendsmap [trendsmap.com](http://trendsmap.com)

# 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 (mashups)

# 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, Instagram
- New York Times
- Google App Engine ([code.google.com/appengine/](http://code.google.com/appengine/))
- Netflix
- iTunes
- YouTube
- pinboard.in
- Many more...

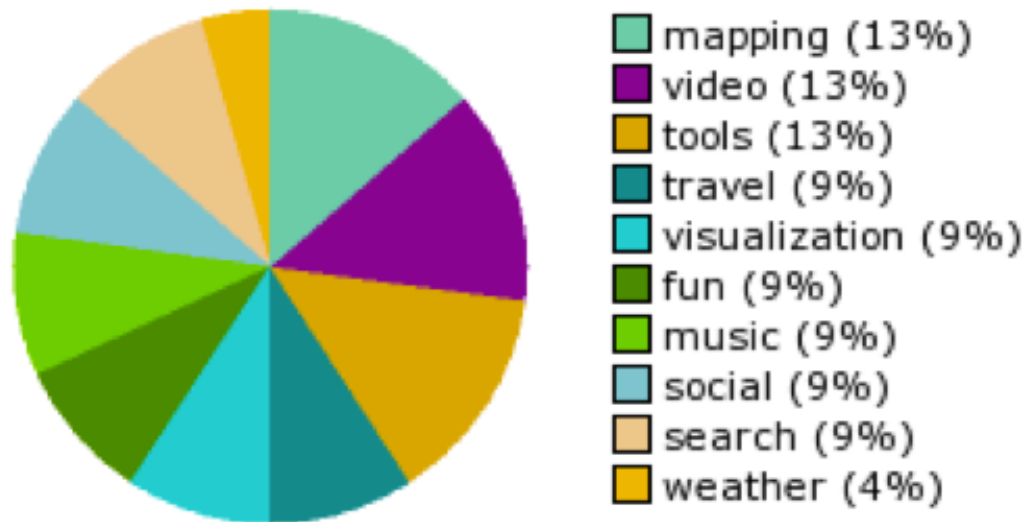
# Top Mashup Types - recent

## Top Mashup Tags

Last 14 days

See all time

Click on a slice or label to see details || Or see the whole  [Mashup Tag Cloud](#)



ProgrammableWeb.com 04/10/13

# Top Mashup Types

## Top Mashup Tags

Last 14 days

See all time

Click on a slice or label to see details || Or see the whole  [Mashup Tag Cloud](#)



- mapping (26%)
- deadpool (14%)
- search (10%)
- social (10%)
- shopping (7%)
- photo (7%)
- video (6%)
- travel (5%)
- music (5%)
- mobile (4%)



# 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/](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](http://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](http://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

- Vendors: LibGuides, Serial Solutions, ExLibris
- 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/>
- IofiAPI: MSU Libraries (ETD, RMT)
- MSU Library Lifestream: RSS services (Twitter, del.icio.us, last.fm, MSU Library Blog)
- Bento Box Search Interface at NCSU Libraries
  - [www.lib.ncsu.edu/search/?q=laser](http://www.lib.ncsu.edu/search/?q=laser)



# 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



# jsFiddle

- Can edit javascript and jquery scripts without a web server
- Can share code examples
- <http://jsfiddle.net/libdevel/4HbhB/7/>



Frameworks & Extensions

jQuery 1.6.4

onLoad

Fiddle Options

External Resources

Languages

Ajax Requests

Examples

Legal, Credits and Links

```
1 <div id="booklist"> </div>
```

HTML

```
1
```

CSS

```
1 $(document).ready(function(){
2     // add List of New Books
3     $(function addBooks(){
4         $.getJSON( "https://spreadsheets.google.com
5         /feeds
6         /list/0AwwFF5rXKpNsdEh5bVN0d2xLTmV0QmdMSzFnc1ducFE
7         /od6/public/values?alt=json-in-script&callback=?",
8         function (data) {
9             $('div#booklist').append('<div
10            class="new_books"></div>');
11            $.each(data.feed.entry, function(i,item){
12                var record = '<span
13                style="display:none">' + item.id.$t + '</span>';
14                // if there is an ISBN, look for an
15                image
16                if (item.gsx$isbn.$t) {
17                    var img = new Image();
18                    img.src =
19                    'http://covers.openlibrary.org/b/isbn/' +
20                    item.gsx$isbn.$t + '-S.jpg';
21                    if (img.width > 2) {
22                        record += '';
25                    } else {
26                        record += '';
29                    }
30                }
31            });
32        }
33    });
34 }
```

JavaScript



Result

[The wilderness from Chamberlain Farm : a story of hope for the American wild](#)  
Author: Dean B Bennett



NO IMAGE AVAILABLE

[The Allagash](#)  
Author: Lew Dietz  
Description: Wonderful history of the Allagash.  
Makes me want to visit



NO IMAGE AVAILABLE

[Night train at Wiscasset Station](#)  
Author: Lew Dietz; Kosti Ruohomaa



NO IMAGE AVAILABLE

[Away from it all.](#)

# Last thoughts...

- This stuff is becoming mature.
- 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](http://www.jasonclark.info)

[twitter.com/jaclark](https://twitter.com/jaclark)

406-994-6801