THE EVERGREEN HTTP INTERFACE

Generating New Item Lists Elegantly and Automatically

OTHER OPTIONS

- LibraryThing Widget
- Goodreads Widget
- LibraryThing for Libraries Widget (Subscription)
- Manual entry and retrieval
POTENTIAL ISSUES

- Add additional steps for technical services and other staff.
- Limited control over display and layout.
- Inconsistent behavior of some third-party widgets.

BUT THERE’S A BETTER WAY!
EVERGREEN HTTP CALLS

- Up-to-date information!
- Automatically generated!
- You have control over how the data is handled!
- Compatible with any system!

ISSUES WITH EVERGREEN

- Higher learning curve!
- You are responsible for integration!
- Higher learning curve!
- Calls can be slow, especially when they’re big!
- Higher Learning Curve!
Retrieving data in a standardized XML format

THE ATOMIC CALL

http://gapines.org/opac/extras
/browse/atom-full/item-age
/<PINES Name>/1
/<Number of Items To Retrieve>
WHAT IT DOES

- http://gapines.org/opac/extras/browse
  - The PINES Evergreen server OPAC, extras, browsing service
- /atom-full/
  - Return the results as an Atom-standard XML document
- /item-age/
  - Sort the results by age.

WHAT IT DOES

- /<PINES NAME>/
  - Variable, specifies the location to retrieve items for. Examples – GCHR, ARL-ATH, etc.
- /1/
  - First page of results
- /<Number of Items to Retrieve>/
  - Variable…the number of items to retrieve per page.
http://gapines.org/opac/extras/browse/atom-full/item-age/GCHR/1/5

WHICH GIVES US...
THAT’S A LOT OF INFORMATION!

- Not supported on all hosting services.
- Low learning curve in general, higher curve for exporting data.
- Might not play well with your CMS.

**PYTHON**

Using feedparser to convert the XML output of Evergreen into a JSON file which can be manipulated easily with Javascript.
- Near universal support
- Higher learning curve than Python, but incorporates web requests more intuitively
- More likely to play well with your CMS.

**PHP**

Using simplexml and json_encode to create an easily manipulated JSON file.

**OTHER OPTIONS**

- Ruby on Rails
  - self-contained MVC framework
  - similar compatibility issues to Python
- Perl
  - Ubiquitous
  - XML requires some extra libraries and steps to deal with
- C#
  - Requires an ASP.net server
TWO METHODS

- MVC Based
  - Translate XML into relevant content
  - Save content in a database model
  - Retrieve information from the database at will
  - Example at https://gchrl.org

- Live Feed
  - Retrieve XML content
  - Extract relevant information
  - Translate into an easily used format (JSON, as an example)
  - Example at http://gchrlphp.dreamhosters.com/json_test.php

HOW TO GET THE DATA - PHP

- Using $xml = simplexml_load_file($url)
- Important! Use xpath() to navigate the atom structure!
  - foreach($xml->xpath('//atom:entry') as $entry);
  - $entry->registerXPathNamespace('atom', 'http://www.w3.org/2005/atom');
HOW TO GET THE DATA - PYTHON

- Using feedparser
  - First, install feedparser using PIP - `pip install feedparser`
- `item_json = feedparser.parse('<insert url here>')`