

OpenBook WordPress Plugin: Open Source Access to Bibliographic Data

OpenBook is a WordPress PHP plugin that implements the Open Library APIs to insert book covers, titles, authors and publishers into web pages. The motive behind the development was to provide an easy alternative to the common practice of linking to Amazon. Open Library was selected as a data source because it is both open source and open data. The plugin is useful for book reviewers, library webmasters, anyone who wants to put book covers and data on their WordPress blog or website. The plugin also allows users to add links to publisher websites, a feature that was considered significant to independent publishers.

by John Miedema

Part I - Impetus and High Level Design Choices

Impetus

Like many other book bloggers, I have had a long-standing practice of linking book reviews to their associated book data on Amazon's website. It was a convenience for readers seeking detailed bibliographic data or wishing to purchase the book. Amazon was the *de facto* source. A year ago, an employee of Random House stumbled across my blog. She offered occasional free books in exchange for a link to their website when I posted a review. This practice seemed fair. After all, a publisher puts more work and risk into a book than a distributor. I adopted a policy of linking to the publisher's website for all the books I reviewed.

Recent aggressive moves by Amazon compelled me to think deeper about the need to link to an alternate source. The Wall Street Journal reported that Amazon had "notified publishers who print books on demand that they will have to use its on-demand printing facilities if they want their books directly sold on Amazon's Web site" (Trachtenberg, 2008, March 28, p. B.4). The analysis indicated that the move would likely generate a significant profit for Amazon. The New York Times (Carvajal, 2008, June 16) reported that Amazon had disabled its "buy button" for print-on-demand titles published by the British unit of Hachette Livre. The publisher lost single-click purchase power.^[1] Readers would have to purchase these titles from third-party vendors. The announcement of Amazon's policy and reaction is being tracked at Writer's Weekly (2008). The developments are bad news for independent publishing and thought; no doubt they made every librarian wince. As the "buy button" incident demonstrates, links are coin on the web. If anything is to be done to encourage diversity, it will have to be with links.

The OpenBook WordPress plugin was developed to provide an easy way for non-technical web users to link to an alternate source of bibliographic data. The plugin was developed hacker style, and blogged while in process (Miedema, 2008a). It got picked up by fellow book bloggers, and by library webmasters who build their website with WordPress. The ongoing interest is encouraging. This article describes the development process and technical details about the software.

Platform and Language

OpenBook is built for WordPress because it is one of the popular blogging platforms^[2], and because WordPress is distributed under the GNU General Public License (GPL) Version 2 (WordPress, 2008). GPL is the most widely used free and open source (FOSS) license. Among other things, this license disallows commercial exclusivity to code. It seemed the right place to start.

Since the platform was WordPress it followed that the programming language is PHP. With minor modifications, the code can quickly be adapted to any environment that supports PHP.

Data Source

The most significant decision was the choice of the bibliographic data source. The ideal source would provide an application programming interface (API) that returned bibliographic data on publisher websites in return for a book identifier, e.g., ISBN. There is no single source of this kind, and publishers do not use a common standard across their websites. Three other sources were evaluated: LibraryThing (<http://librarything.com/>), a social cataloguing web application developed by Tim Spalding; Open Library (<http://openlibrary.org/>), a project of the Internet Archive and the Open Content Initiative; and WorldCat (<http://worldcat.org/>), the catalogue of OCLC.

Data Richness. Of the three sources, LibraryThing has the richest social software functions and data. WorldCat has social software but it is hardly used. Open Library does not have social software but it does have the full text of public domain books that have been scanned. While rich book data is of value to book bloggers and readers, it was not considered a differentiator for the first version of OpenBook.

Profit Status. The purpose of OpenBook is to provide an alternative to Amazon because of its aggressive commercial moves. Given that, a non-profit might seem a better choice for OpenBook than a private operation. LibraryThing is a private operation, though it is not a large corporation. The Internet Archive and OCLC are non-profits. However, it was recognized that neutrality is always difficult, and non-profits have agendas too. Furthermore, there may be advantages to shared projects with private companies willing to invest resources in innovation. For

these three sources, profit status alone was not a significant differentiator.

Open Data. WorldCat has the greatest number of titles but only for member libraries, with data accessible only to librarians. LibraryThing and Open Library both have open data - anyone can add and modify titles. It may be argued that open data is more prone to error, but only librarians can correct errors in WorldCat and the incentive for doing so is insufficient (Beall, 2008). The ability to add outlier titles is central to the current purpose of encouraging diversity in the book industry. WorldCat was ruled out.

FOSS. LibraryThing was built on open source software, and provides open source APIs, but the core LibraryThing code is proprietary. It must be so; otherwise competitors would steal their labour and put LibraryThing out of business. The open source code at LibraryThing may be called commercial open source software (COSS) not FOSS (described earlier). Any private business that uses COSS has the ability and right to discontinue free use of its software at any time. Open Library's code is licensed under GNU Affero General Public License (<http://www.gnu.org/licenses/agpl.txt>). Should the day come that Open Library misuses its monopoly on bibliographic data, another operation could take its code, and compete with it on equal footing.

Open Library was selected as the data source to replace Amazon. Since the deciding factors were open data and FOSS, the product was named OpenBook. It was recognized early on that Open Library is still in beta and has some bugs; this risk was considered acceptable, especially since the OpenBook plugin is itself an experimental piece, designed more for the argument it makes than for its technical quality. The source will be re-evaluated in the future for stability and enhanced features. It should be noted that each of the sources links to Amazon, in addition to links to other sellers and WorldCat. There is no intent to block access to Amazon, if such a thing was even possible, but rather to inject some diversity into the book business.

Part II - Technical Design

OpenBook is a relatively small piece of software, implemented in 250 lines of code. The following sections highlight significant technical features. The source code can be [downloaded](#) from the WordPress server (Miedema, 2008b).

Graphic Design

The graphic design for OpenBook is a simple combination of a cover image, title, author and publisher. Figure 1 shows a sample of OpenBook used on my website, a review of *The Republic of Nothing* by Lesley Choyce (2007), published by Goose Lane. The cover image is an alternate cover that I added to Open Library. It comes from the publisher's website where it is expressly offered as a download (Goose Lane, 2008). Many publishers encourage the distribution of their cover images because of their promotional value.

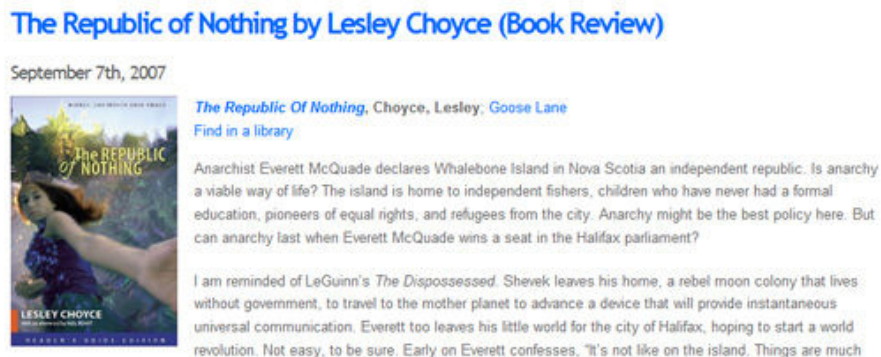


Figure 1. Sample OpenBook Data

Light styling is added to float the text next to the image, and to distinguish the text items. The cover and title link to the detailed web page in Open Library, and the publisher text links to the publisher's website. For now, OpenBook only displays data for the first set of tags it finds, and always displays it in the upper left corner of the page. The simplicity of the design was deliberately chosen, to avoid formatting problems in the first version of the plugin.

The OpenBook Tags and Book Identifier

To display the bibliographic data, users need only insert two tags and a book identifier into their page, as in Sample 1.

```
01. view plain copy to clipboard print ?  
[openbook]0864921535[/openbook]
```

Sample 1

The goal of Open Library is to have a web page for every book ever published (Open Library, 2007). Every book has a webpage with a distinct URL, as in Sample 2.

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```
01. <a href="http://openlibrary.org/b/OL882707M">http://openlibrary.org/b/OL882707M</a>
```

Sample 2

Sample 2 contains the Open Library key,

```
"/b/OL882707M"
```

[3]. The key is required to look up the item with the Open Library APIs.

OpenBook allows users to look up books in two ways. If a user provides an Open Library key (detected by the “/b/OL” pattern), OpenBook can look up the book data directly using the Open Library “get” API. Users should not have to look up these proprietary numbers, so OpenBook also allows users to look up books by their ten or thirteen digit ISBN. There can be multiple keys for a single ISBN, so OpenBook must first use the Open Library “search” API to find a list of matching keys. By default, OpenBook uses the most recent record, but users can pass an additional argument so select a particular version. Once the key is selected, the “get” API can be used as above. See the examples in the following section.

The ability to search by Open Library key serves two important purposes. First, during the development process, it was learned that Open Library does not automatically add titles to its search index if the titles have been added manually through the website. In this case, the title cannot be found with the search API until a scheduled manual routine is performed to add it to the search index. Users cannot count on finding new titles by ISBN, so they must use the Open Library key as a workaround. The folks at Open Library told me that a fix was high on their priority list. I have recently opened a ticket on the problem (2008). It has been marked as high importance and work is in progress, but the problem is not yet resolved. Second, a main purpose of OpenBook was to provide access to independent publishers. Some books and e-books do not have an ISBN. The ability to search by Open Library key provides access to these titles.

The Open Library APIs

OpenBook uses two of the four APIs documented by Open Library (2007b)— the “get” API and the “search” API, as described above. The APIs for Open Library are exposed through a browser URL and querystring.

The “get” API retrieves bibliographic data for a specific key. The API and the key are assembled into a URL, such as the one in Sample 3:

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```
01. <a href="http://openlibrary.org/api/get?key=/b/OL882707M">http://openlibrary.org/api/get?key=/b/OL882707M</a>
```

Sample 3

The “search” API looks up keys by its ten or thirteen digit ISBN, and is formatted like the example in Sample 4:

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```
01. http://openlibrary.org/api/search?q={"query":"(isbn_10:(0864921535) OR isbn_13:(0864921535))"}
```

Sample 4

By default, the APIs return an object. If you enter one of the above URLs in a browser’s address bar, you will be prompted to save the results to disk, a file that you can read with a text editor. You can return results as text by appending

```
"&text=true"
```

to the querystring. You can format them for easier reading in your browser by appending

```
"&prettyprint=true"
```

Retrieving the Data in PHP

PHP provides two methods by which to call the URLs and insert the results into a variable in order to work with them. In the first release of OpenBook I used the “file_get_contents” method, but user feedback indicated that many servers disallow this method. I switched to the curl method, as illustrated in Sample 5:

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```
01. $ch = curl_init($url);
02.
03. $bookdata = curl_exec($ch);
```

Sample 5

This sample creates a curl handle, `$ch`, based on the URL for the Open Library API. The handle is executed, and the book data is returned into a PHP variable, `$bookdata`.

Parsing the Data in PHP

The data returned by Open Library is in JavaScript Object Notation (JSON) format, an industry standard, text-based, language-independent data interchange format. It is easy for people to read and for machines to parse. PHP 5.2 comes with a JSON library that makes quick work of data extraction. My web host defaulted to PHP 4, but I was able to upgrade with a simple switch on my control panel (eNetfirms, 2008). Sample 6 shows the syntax for extracting bibliographic data into PHP variables from the `$bookdata` variable created in the last section.

```
view plain copy to clipboard print ?
01. $obj = json_decode($bookdata);
02. $result = $obj->{'result'};
03.
04. $title = $result ->{'title'};
05. $subtitle = $result ->{'subtitle'};
06.
07. $authors = $bookdataresult ->{'authors'};
08. $publishers= $result ->{'publishers'};
```

Sample 6

The upgrade to PHP 5.2 was easy, but it caused a few stumbles for people when I released OpenBook. A subsequent update of OpenBook included the JSON library with the plugin, making it compatible with PHP 5.0.

Cover Images and Publisher Links

The `$bookdata` results do not typically contain a URL for the book's cover image, only for alternate covers added by users. A blog reader commented that I could construct the URLs for the cover images because Open Library uses standard URL and file naming conventions. Logic was added to OpenBook to construct the cover URL or use the alternate URL if it was provided.

The results did not contain links from the book to the publisher's website. No known bibliographic data source provides this association. It was decided that for this first version of OpenBook, a link could be passed as an argument. More information about the arguments is provided in the readme file that comes with the plugin download. Sample 7 illustrates how a publisher link can be passed.

```
view plain copy to clipboard print ?
01. [openbook]0864921535,,,http://www.gooselane.com/[openbook]
```

Sample 7

Assembling the HTML and Hooking into WordPress

Once all the data was gathered, it was formatted into an HTML string according to the graphic design specification discussed above. WordPress allows for the creation of filter functions that can be used to search and replace content whenever a user saves a page. OpenBook searches for the tags, then replaces them with the HTML display of the bibliographic data. The completed code went into a function in a file with a "php" extension. The file was uploaded to the WordPress server where anyone can [download it](#) (Miedema, 2008b).

Part III - Progressive Software and Future Plans

Software that matters is shaped by progressive ideas, such as those of Brewster Kahle, founder of the Internet Archive. During the development of OpenBook, Microsoft announced that it was ending its role in the Open Content Alliance from which Open Library gets its scanned content (Anderson, 2008). Kahle made some noteworthy remarks. He thanked Microsoft for its generous contributions, and for doing more than its share to facilitate a smooth ramp-down. He also viewed the development as a potential move forward:

He's a firm believer in the idea that corporations should not be the entities we trust to provide access to important cultural data stores. If people think that corporations are the right way to access the history of human discourse, Kahle says they're in for "a series of very rude shocks." (Anderson, 2008)

Kahle's view is consistent with the purpose of OpenBook to provide an easy alternative linking practice to counter Amazon's near-monopoly on book sales, and more importantly, to encourage diverse media and thought. Librarians are rapidly increasing their technological capabilities, but progressive ideas are what make this software relevant.

OpenBook was drafted quickly as more of an argument piece than an enduring piece of software. There has been a strong positive response to

the first release, and I have supported the plugin with enhancements based on user feedback. The latest version (1.6 beta) has new features, including a link to the WorldCat record for a valid ISBN, the ability to insert multiple covers in one post, and handling for Open Library down time. Keep up on the latest features at the OpenBook support page (Miedema, 2008a). It is expected that OpenBook will move out of beta stage by the end of 2008.

References

- [^],[^]Anderson, Nate (2008). Why killing Live Book Search is good for the future of books. Retrieved from <http://arstechnica.com/news.ars/post/20080526-why-killing-live-book-search-is-the-right-thing-for-ms.html> on August 7, 2008.
- [^]Beall, Jeffrey (2008). OCLC: A Review, in Roberto, K.R., Eds. *Radical Cataloging: Essays at the Front*, pp. 85-93. McFarland. Retrieved from <http://eprints.rclis.org/archive/00013701/> on August 7, 2008.
- [^]Carvajal, Doreen (2008, June 16). Small Publishers Feel Power of Amazon's 'Buy' Button. The New York Times. Retrieved from <http://www.nytimes.com/2008/06/16/business/media/16amazon.html> on August 7, 2008.
- [^]Choyce, Lesley (2007). *The Republic of Nothing*. Fredericton, New Brunswick: Goose Lane.
- GNU (2007). GNU AFFERO GENERAL PUBLIC LICENSE. Retrieved from <http://www.gnu.org/licenses/agpl.txt> on August 7, 2008.
- Goose Lane (2008). *The Republic of Nothing: Reader's Guide Edition*. Retrieved from <http://www.gooselane.com/book/9780864924933> on August 7, 2008.
(COinS)
- [^], [^]Miedema, John (2008a). OpenBook WordPress Plugin. Retrieved from <http://johnmiedema.ca/openbook-wordpress-plugin/> on August 7, 2008.
- [^], [^]Miedema, John (2008b). OpenBook Book Data - WordPress Plugins. Retrieved from <http://wordpress.org/extend/plugins/openbook-book-data/> on August 7, 2008.
- [^]Netfirms (2008). How do I switch between PHP4 and PHP5? Retrieved from <http://support.netfirms.com/article.php?id=713> on August 7, 2008.
- [^]Open Library (2007). About us. Retrieved from <http://openlibrary.org/about> on August 7, 2008.
- [^]Open Library (2007b). Open Library API. Retrieved from <http://openlibrary.org/dev/docs/api> on August 7, 2008.
- [^]Open Library (2008). Bug #267853 in Open Library: Manually added titles not added to search index. Retrieved from <https://bugs.launchpad.net/openlibrary/+bug/267853> on September 20, 2008.
- [^]Trachtenberg, Jeffrey A. (2008, March 28). Amazon Tightens Grip on Printing. Wall Street Journal (Eastern Edition), p. B.4. Retrieved from ABI/INFORM Global database. (Document ID: 1452793061) on August 7, 2008.
- [^]WordPress (2008). GPL. Retrieved from <http://codex.wordpress.org/GPL> on August 7, 2008.
- [^]Writer's Weekly (2008). Amazon BookSurge Information Clearinghouse. Retrieved from <http://www.writersweekly.com/amazon.php> on August 7, 2008.

Footnotes

- [1] Subsequent investigation (August 2008) has been performed on Amazon using the Hachette Livre title mentioned in the NY Times article, "The Hachette Guide to French Wine". The Buy-with-1-Click button is currently active for this title (after signing in). It is unclear if Amazon has back-tracked on the policy, perhaps due to the legal action reported at Writer's Weekly (2008).
- [2] WordPress distinguishes between blog posts and pages. OpenBook works equally well in both. For simplicity this article just refers to pages.
- [3] The "b" prefix stands for book. Other searchable objects have a different prefix, e.g., "a" for author.

About the Author

John Miedema is a library student at the University of Western Ontario and an IT Specialist with IBM. John balances out his busy IT world with slow reading. He is the author of a forthcoming book on slow reading to be published by Litwin Books. Contact John at mail@johnmiedema.ca.

Tags: book review resources, WordPress, WordPress plugins

Subscribe to comments: [For this article](#) | [For all articles](#)