

## CHAPTER 14

# Social Bookmarking and bibliographic systems

One of the fundamental challenges of using the Web is keeping found things found, whether it be at the basic level of simple URLs, but also other digital content such as images, data sets. Social bookmarking has arisen to be a popular solution to this problem. According to "7 Things you should know about....social bookmarking"<sup>1</sup>, "[s]ocial bookmarking is the practice of saving bookmarks to a public Web site and 'tagging' them with keywords." A specific type of reference is a "bookmark" or URL or "favorite" or URL. One can try to manage them in the browser by storing them as "favorites" or "bookmarks." Social bookmarking involves storing one's references online and making bookmarking a collaborative process.

There are several reasons for using social bookmarking:

- \* To keep track of interesting items (URLs) you find on the web in the hope that you will be able to find this again. This process is nicely described by the words of a research project "keeping found things found".<sup>2</sup>
- \* To have basic metadata or citation information about bookmarks, including metadata about the item so that you can cite the item and tell others about these items.
- \* To save the full text of the entry. So you can still access the article if the original source goes away. (with full text, you'll also be able to build your own discovery services.)
- \* To find materials that are similar to what you already have. The ability to find more and related materials is a major reason for the existence social bookmarking. You can certainly save bookmarks on your own browser. Of course, you can just copy down the URL, paste it in a Word document, drop them in to an e-mail message, but doing so either keep us this information to yourself. or just a very select group of friends. Social bookmarks, put the focus on sharing bookmarks with others so that others can find and learn from you. tagging is widely used to forge connections among disparate sources. ( think about applying to all we talked about in Chapter 3 of this topic.)

Social bookmarking is an area in flux. There is a helpful chart (from 2006) comparing features of 19 systems<sup>3</sup> The Wikipedia has a list of social bookmarking sites.<sup>4</sup> It's difficult

---

<sup>1</sup> <http://www.educause.edu/ir/library/pdf/ELI7001.pdf>

<sup>2</sup> <http://kftf.ischool.washington.edu/>

<sup>3</sup> [http://www.irox.de/file\\_download/3](http://www.irox.de/file_download/3) -- linked from <http://www.irox.de/roxomatic/1050>

<sup>4</sup> [http://en.wikipedia.org/wiki/List\\_of\\_social\\_software#Social\\_bookmarking](http://en.wikipedia.org/wiki/List_of_social_software#Social_bookmarking)

to keep track of the many sites that come and go. A list that is reasonably up to date (last updated 4/16/2007) lists 200+ sites<sup>5</sup>.

Social bookmarking is of further interest in the context of this book because of the extensibility/remixability being built into these systems. Social bookmarks also lend insight into other systems. Delicious, the grand-daddy of social bookmarking sites, is generally credited with kicking off not only the latest wave of social bookmarking and tagging.

This chapter:

- \* provides an overview of the social bookmarking landscape, which has many competitors.
- \* walks through a select set of social bookmarking systems: del.icio.us, Yahoo!'s myweb 2, and connotea
- \* walks through the APIs, focusing in detail on delicious (which is influential and the model, for instance, for Shadows) and then comparing it to those of Yahoo! and Connotea
- \* discusses how to use all these services together, including interoperability issues

## So many social bookmarking sites...

There are a lot of social bookmarking sites. A reasonable approach is to focus on delicious, the one to which all other social bookmarking system are compared. Moreover, delicious has an API, and there is much to learn from it.

There are other social bookmarking sites that I will mention and examine briefly:

- \* Connotea and CiteULike more scholarly social bookmarking sites. Connotea is backed up by Nature Publishing and therefore it's likely to have some longevity.
- \* Yahoo! MyWeb 2.0 and Bookmarks because Yahoo! is pursuing these properties it in spite of all ready owning delicious. Functionally, Yahoo! MyWeb allows you to store pages and to make use of my Yahoo! identity network., a major social network.

## Using ProgrammableWeb to Examine the Popularity of APIs

A good way to pick up on what to focus on is to see what is listed on [programmableWeb.com](http://programmableweb.com) -- to focus on the systems that actually have APIs. Go to

<http://www.programmableweb.com/apilist/bycat>

and look for services that are in the "Bookmarks" category. On April 29, 2007, they were:

- \* del.icio.us
- \* Simpy
- \* Blogmarks
- \* Scribble

---

<sup>5</sup> <http://3spots.blogspot.com/2006/01/all-social-that-can-bookmark.html>

- \* Shadows
- \* Jots
- \* Rrove
- \* OnlyWire
- \* linkaGoGo
- \* Ma.gnolia
- \* Rrove

If you look at what social bookmarking sites have been mashed up, let's list them sorted by mashup count:<sup>6</sup>

---

API	Description	Category	Mashups
del.icio.us	Social bookmarking	Bookmarks	75
Ma.gnolia	Social bookmarking service	Bookmarks	2
Shadows	Social bookmarking and community	Bookmarks	2
Simple	Social bookmarking	Bookmarks	1
Rrove	Social Bookmarking for Places Create & Share Maps	Bookmarks	1
Jots	Social bookmarking	Bookmarks	1

(Table generated on April 26, 2007)

The fact that del.icio.us is the object of an order of magnitude more mashup activity than the rest of bookmark services combined points to our focusing on del.icio.us.

## del.icio.us

Del.icio.us is the grand-daddy of social bookmarking sites, the site that kicked the whole folksonomic craziness. Its website is

<http://del.icio.us/>

The main object of importance in del.icio.us are bookmarks, ie., URLs. You can associate tags with a given URL. You can look at an individual's collection of URLs and the tags that he or she uses. We look again at the URL structures by browsing through the site and noting the corresponding URLs.

You can look at the public bookmarks for a specific user (e.g., for **rdhyee**):

<http://del.icio.us/rdhyee>

You can see all the bookmarks tagged "NYTimes" by rdhyee:

<http://del.icio.us/rdhyee/NYTimes>

---

<sup>6</sup> <http://www.programmableweb.com/apilist/bymashups>

You can see all the URLs that people have tagged with "NYTimes":

<http://del.icio.us/tag/NYTimes>

or just the popular ones:

<http://del.icio.us/popular/NYTimes>

Today's popular items:

<http://del.icio.us/popular/>

or just the "fresh" popular ones:

<http://del.icio.us/popular/?new>

Now correlating a URL to a del.icio.us page is a bit trickier. Consider the URL

<http://harpers.org/TheEcstasyOfInfluence.html>

which you can reference from del.icio.us at

<http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5>

---

How do you get 53113b15b14c90292a02c24b55c316e5 from <http://harpers.org/TheEcstasyOfInfluence.html>?  
Answer -- it's an md5 hash. In python: `md5.new("http://harpers.org/TheEcstasyOfInfluence.html").hexdigest()`  
yields '53113b15b14c90292a02c24b55c316e5'

---

Note that

<http://del.icio.us/url?url=http://harpers.org/TheEcstasyOfInfluence.html>

does work -- and redirects to

<http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5>

## Using the del.icio.us API

From the documentation for the API

<http://del.icio.us/help/api/>

we learn that

- \* All del.icio.us API call must be sent over https
- \* All calls require HTTP-Auth -- that means there are no API keys *per se* but all API calls are tied to a specific user account.
- \* You need to watch for the 503 HTTP error (which would mean that your calls are being throttled and that you need to slow down the rate of your calls.)

---

Implication: you can work on your own delicious references but not others unless they give you their credentials. All API calls to delicious are made in the context of a specific user. There are no un-authenticated calls as there

are Flickr.

---

There are four major sections of the API:<sup>7</sup>

- \* Update - "Check to see when a user last posted an item"
- \* Tags - "Get a list of tags, rename them."
- \* Posts - "Get a list of posts, add, delete."
- \* Bundles - "Get bundles, create and delete."

In the following subsections, I give you a flavor of the capabilities of the delicious API but do not comprehensively document it.

## Update

The documentation for the update method is found at:

<http://del.icio.us/help/api/update>

Let's look at two ways to work through various methods listed in the API. The first is to use a web browser, while the second uses **curl**. Let's use the update method as an example:

1. With a web browser, go to:

<https://api.del.icio.us/v1/posts/update>

and when prompted, enter your del.icio.us username and password.

2. With curl, you would run

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/update
```

where USER and PASSWORD are your username and password. In either case you, if your username/password are correct, you should get a response like:

```
<?xml version='1.0' standalone='yes'?>  
<update time="2007-04-29T22:49:55Z" />
```

---

For the following examples, we will use curl only.

---

## Tags

To return the complete list of tags used by a user and the number of times a given is used:

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/tags/get
```

To rename the tag FEDORA to fedora:

---

<sup>7</sup> <http://del.icio.us/help/api/>

```
curl -u USER:PASSWORD "https://api.del.icio.us/v1/tags/rename?old=FEDORA&new= Fedora"
```

## Posts

The posts method has several submethods: [get](#), [recent](#), [all](#), [dates](#), [delete](#).<sup>8</sup>

### get submethod

Use

```
https://api.del.icio.us/v1/posts/get?
```

with optional parameters tag, dt (for date in CCYY-MM-DDThh:mm:ssZ) format), and url to return posts matching the arguments. For example, to get posts with the tag mashup:

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/get?tag=mashup
```

You can use this submethod to figure out the number of times an article has been posted. Consider the following scenario. I've posted a URL

```
http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm
```

to del.icio.us and want to track the number of times it has been posted to del.icio.us. How do I do so?

```
curl -u USER:PASSWORD  
https://api.del.icio.us/v1/posts/get?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm
```

returns

```
<?xml version="1.0" standalone="yes"?>  
<posts dt="2007-04-26" tag="" user="rdhyee">  
  <post href="http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm"  
    description="ALA | Changing Roles of Academic and Research Libraries"  
    hash="fa5be4b4401acf147ff6c8634b55cdda" others="49" tag="library2.0 libraries  
    academic future" time="2007-04-26T19:45:56Z"/>  
</posts>
```

The "others" attribute in the post tag gives the number 49.

What if I don't have the URL in the library? What would the API return? It returns

```
<?xml version='1.0' standalone='yes'?>  
<posts dt="" tag="" user="rdhyee">  
</posts>
```

You need to add something to the library to follow it -- at least with the <https://api.del.icio.us/v1/posts/get> method. It does to work if you put the URL in your library but not share it.

---

<sup>8</sup> <http://del.icio.us/help/api/posts>

## recent submethod

`https://api.del.icio.us/v1/posts/recent?`

returns a list of the most recent posts (up to 100), filtered by the optional arguments tag and count. For example:

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/recent?count=5
```

returns the last 5 posts.

## all submethod

The all submethod returns all your posts. You are advised to use this call sparingly and use the **update** function to see if you need to do this call at all. You can filter by tag, such as in the following call (to get all posts with the tag architecture):

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/all?tag=architecture
```

## add submethod

You use add to add posts to delicious. The Method name:

```
https://api.del.icio.us/v1/posts/add?
```

with the following parameters. Two are required:

- \* `&url` (required) - the url of the item.
- \* `&description` (required) - the description of the item.

while the rest are optional:

- \* `&extended` (optional) - notes for the item.
- \* `&tags` (optional) - tags for the item (space delimited).
- \* `&dt` (optional) - datestamp of the item (format "CCYY-MM-DDThh:mm:ssZ"). Requires a LITERAL "T" and "Z" like in ISO8601 at <http://www.cl.cam.ac.uk/~mgk25/iso-time.html> for example: "1984-09-01T14:21:31Z"
- \* `&replace=no` (optional) - don't replace post if given url has already been posted.
- \* `&shared=no` (optional) - make the item private

Let's put description to be "ALA | Changing Roles of Academic and Research Libraries" and tags to be "library 2.0 academic ACRL technology"

```
curl -u USER:PASSWORD  
"https://api.del.icio.us/v1/posts/add?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm&description=ALA+%7C+Changing+Roles+of+Academic+and+Research+Libraries&tags=library+2.0+academic+ACRL+technology"
```

which returns

```
<?xml version='1.0' standalone='yes'?>  
<result code="done" />
```

## dates submethod

For instance:

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/dates?tag=NYTimes
```

## delete submethod

```
curl -u USER:PASSWORD "https://api.del.icio.us/v1/posts/delete?url=
http://del.icio.us/help/api/posts"
```

to delete a post with the URL of <http://del.icio.us/help/api/posts>

## Bundles

Bundles in [del.icio.us](http://del.icio.us) allow you to collect tags into organizational sets. The API enables you to manipulate them. See the documentation at

<http://del.icio.us/help/api/bundles>

Let's look at some examples:

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/tags/bundles/all
```

retrieves all the bundles.

You can create a "source" bundle with two tags: [NYTimes](#) and [MITTechReview](#).

```
curl -u USER:PASSWORD
"https://api.del.icio.us/v1/tags/bundles/set?bundle=sources&tags=NYTimes+MITTechRevi
ew"
```

You can delete bundles:

```
curl -u USER:PASSWORD
"https://api.del.icio.us/v1/tags/bundles/delete?bundle=junkbundle"
```

## RSS

There are many ways to get RSS feeds from [delicious](#). See:

<http://del.icio.us/help/rss>

for a complete list of the various RSS feeds that are available.

## JSON

See

<http://del.icio.us/help/json/>

to learn about sources of JSON from [del.icio.us](http://del.icio.us)

## Third-Party tools for del.icio.us

A useful reference for what others have done with the [delicious](#) API is:



<http://del.icio.us/help/thirdpartytools>

I've started to look through the long list of tools developed around del.icio.us: Complete Tools Collection<sup>9</sup> I was surprised to find only a few things that I wanted to follow up with:

- \* Foxylicious - Firefox and del.icio.us bookmark integration<sup>10</sup>
- \* Cocoalicious: A Cocoa del.icio.us Client for Mac OS X might be a good desktop tool.<sup>11</sup>
- \* When it comes time to mirror del.icio.us to a local database, I can take a look at MySQLicious for "del.icio.us to MySQL Mirroring"<sup>12</sup>

## PHP and delicious

There are two easy to find PHP API kits for delicious:

- \* <http://dietrich.ganx4.com/delicious/>
- \* <http://www.ejeliot.com/pages/5>

## Yahoo! Bookmarks and MyWeb

Yahoo!'s social bookmarking system is worth looking at because Yahoo! is a big company (with tons of users), that is adamant about getting heavily in the folksonomic space (by buying del.icio.us and flickr for instance). Yahoo!'s MyWeb2.0 certainly has attractive features, including the ability to save web pages. In addition to del.icio.us, which Yahoo! owns, Yahoo! also has two other bookmarking service:

<http://bookmarks.yahoo.com/>

and

<http://myweb2.search.Yahoo.com/>

The relationship among the various Yahoo! owned services is a bit confusing. According to an explanation by one Yahoo! employee involved with the various bookmarking systems:<sup>13</sup>

- \* Yahoo! Bookmarks is for personal bookmarking while del.icio.us is for social bookmarking.
- \* "Behind the scenes, we are extending the social bookmarking platform built for MyWeb to store the data for Yahoo! Bookmarks and soon del.icio.us. This will allow for seamless migration from one service to another while preserving your existing bookmarks."

---

<sup>9</sup> <http://www.quickonlinetips.com/archives/2005/02/absolutely-delicious-complete-tools-collection/>

<sup>10</sup> <http://dietrich.ganx4.com/foxylicious/>

<sup>11</sup> <http://www.scifihifi.com/cocoalicious/>

<sup>12</sup> <http://nanovivid.com/projects/mysqlicious/#knownissues>

<sup>13</sup> <http://www.techcrunch.com/2006/10/24/Yahoo!-bookmarks-enters-21st-century/#comment-297657>

- \* " MyWeb and Yahoo! Bookmarks share the same backend database."

Let's look at using the API documentation:

<http://developer.yahoo.com/search/myweb/>

There are two versions of the API: v 1.0 and v 2.0. The following are illustrative calls to the API and not meant to demonstrate the full functionality of the API

- \* To list a user's folders:<sup>14</sup>

<http://search.yahooapis.com/MyWebService/V1/listFolders?yahooid=rdhyee&appid=mashupguide.net>

- \* To list the URLs in a folder:<sup>15</sup>

<http://search.yahooapis.com/MyWebService/V1/listUrls?appid=mashupguide.net&yahooid=rdhyee&folder=Testing+Yahoo+API>

- \* To the list of a user's tags (in v2.0):<sup>16</sup>

<http://api.search.yahoo.com/MyWebService/V1/tagSearch?appid=mashupguide.net&yahooid=rdhyee&results=50>

- \* To do a search for URLs with a certain tag:

<http://search.yahooapis.com/MyWebService/V1/urlSearch?appid=mashupguide.net&tag=mashup>

- \* You can search for related tags:<sup>17</sup>

<http://search.yahooapis.com/MyWebService/V1/relatedTags?appid=mashupguide.net&tag=mashup&results=50>

## Connotea: an academic social bookmarking system

There is an academically oriented social bookmarking systems that deserves further exploration: Connotea, run by Nature Publishing, specializing in the scientific literature. I've started to adding some bookmarks to my account:

<http://www.connotea.org/user/rdhyee>

Documentation for the connotea API:

<http://www.connotea.org/wiki/WebAPI>

Some major similarities to delicious API -- needs authentication -- but you can access other user's posts.

---

<sup>14</sup> <http://developer.yahoo.com/search/myweb/V1/listFolders.html>

<sup>15</sup> <http://developer.yahoo.com/search/myweb/V1/urlSearch.html>

<sup>16</sup> <http://developer.yahoo.com/search/myweb/V1/tagSearch.html>

<sup>17</sup> <http://developer.yahoo.com/search/myweb/V1/relatedTags.html>

To test out your username/password combination:

<http://www.connotea.org/data/noop>

```
curl -u USER:PASSWORD http://www.connotea.org/data/noop
```

get tags for rdhyee:

<http://www.connotea.org/data/tags/user/rdhyee>

Get all the bookmarks for user rdhyee

<http://www.connotea.org/data/bookmarks/user/rdhyee>

RSS feeds: <http://www.connotea.org/rss/recent/tag/social%20bookmarking>

<http://www.connotea.org/tag/social%20bookmarking> -- what's tagged social bookmarking

## Sketches of a specific integrative project around social bookmarking

As you grow in sophistication with using social bookmarking sites, you might want to incorporate richer methods of tagging your bookmarks and downloading references from del.icio.us. For such a purpose, one can imagine a user interface would be a desktop client with the following features:

- \* You would be able to select text in various contexts, the URLs will be extracted and presented in the interface for tagging en masse as well as a single form to fill in titles, descriptions, and notes.
- \* You could bring up previously tags that you've used and so that you can easily apply those tags to selected references.
- \* On the flip side, you might want to do (within this interface or with another one) searches on del.icio.us and bring results back in a number of formats. It would be nice, for instance, to be able to quickly format a reference as HTML or wiki markup or OpenDocument so that I can paste the reference in various places.
- \* Developing this interface could give one an opportunity to explore the potential of incorporating autotagging systems such as Tagyu<sup>18</sup> and tagthe.net.<sup>19</sup>

In integrating various social bookmarking systems, you might ultimately need your own object store in addition to Flickr and del.icio.us to keep track of all your digital objects and references.

Alternatively, can you just make del.icio.us into that central database? How far can del.icio.us be extended? How much bibliographic metadata can be shoehorned into del.icio.us while still honoring the spirit of the application. In other words, if you want to cite a book in del.icio.us and want to stick in things like author, title, isbn, etc, how should you do so? In terms of the mechanics, you can draw from the example of geotagging in Flickr as a precedent, especially with its use of machine tags -- in addition to geotagging. (Others

---

<sup>18</sup> <http://www.tagyu.com/>

<sup>19</sup> <http://tagthe.net/>

have had a similar notion: others have this idea: [Sci-Fi Hi-Fi » Blog Archive » Meta Tags: The Poor Man's RDF?](#)<sup>20</sup>). So should you go ahead and stick in 'bibliotagged' and things like `biblio:author=William_Shakespeare` into delicious? The specific tags are not clear; is the fundamental concept ok? Such shoehorning of metadata comes from a desire for to have an online database of references.

## Conclusions

In this chapter, you learned about social bookmarking as a whole, especially ones that have APIs. We then concentrated on looking at how to use the APIs of del.icio.us, Yahoo MyWeb, and Connotea. We concluded with a sketch of a user interface that could be build to integrate the various social bookmarking systems into a desktop and authoring context.

---

<sup>20</sup> <http://weblog.scifihifi.com/2005/08/05/meta-tags-the-poor-mans-rdf/>