Web-Mediated Resource Distribution for a Library Web Site

Sigrid Kelsey, Librarian, Louisiana State University
Angela Kristin VandenBroek, Library Associate, Louisiana State University

Introduction

"There will be much temptation, mixed with necessity, to curtail institutional expenditures in the immediate future. There has been too little interest, in times of prosperity, to give the college library adequate support. Lessening that interest, in times of depression, would be unfortunate. Lack of funds may make financial dieting a library necessity. It will not be the library alone, but the college and the university as a whole which will suffer if that dieting is, through official lack of interest, carried on so far as to become fasting or starvation" (Walter 1931, 311).

Walter's 1931 observation about the lessening interest and following that, support, of college libraries due to economic downturn is timely as Louisiana's academic libraries face journal cancellations, hiring freezes, and reductions of hours. As demonstrated by the Walter article, librarians have and will continue to provide assistance and instruction to library users, while facing rising costs of library materials and uncertain budgets. In making a case for the importance of librarians, Walter asserts that providing "skilled assistance to students, faculty, and research workers in search for informational material" is critical in academia, and that a trend at the time of the article's publication, "individualized courses," would "awaken a desire on the part of faculty and students for more and better assistance. Such assistance," he notes, "is a commonplace in the laboratory. It is no less important in the library" (309). Nearly eighty years later, librarians remain central to providing the "skilled assistance to students, faculty, and research workers in search of informational material." While technology has changed, students continue to require assistance with finding and evaluating information, and the librarians continue to be the faculty members frequently fulfilling that role. At the same time, when universities are cutting budgets, libraries are often positioned to lose funding. Librarians should be persistent, like Walter, in promoting their importance on the university campus while devising means for achieving more with less funding and fewer human resources.

A lot has changed since 1931, especially in technology. Libraries can provide copies of quality peer- reviewed journal articles for free using email and open access articles. We can communicate faster and more easily with colleagues around the world using the internet. Unlike Walter, we have opportunities to contribute to shared online resources at very little cost to our libraries. This article provides an overview of Delicious and provides techniques (including CSS code and step-by-step directions) and examples of using Delicious to distribute resources on a library web site. The LSU Delicious Subject Guide Project is described in depth, with instructions, ideas to expand it, and an invitation to contribute to it. Delicious provides a platform for academic librarians to use technology to share resources quickly and free of cost, and this article explains several complementary innovations that have the potential to make the gathering and sharing of resources among librarians, and with students, faster and easier. Louisiana academic librarians are moving ahead with identifying and sharing resources, while facing waning fiscal resources.

Exploring Web 2.0

This often misused term describes the transition in the internet experience, from reading static webpages to engaging with dynamic web content that can be contributed to, manipulated, transported and linked. "Web 2.0," a term originally coined by Nancy DiNucci in 1999 and popularized by a 2004 conference on Web 2.0 and subsequent publication by Tim O'Reilly (2005), is a popular topic among Louisiana libraries this year. Bringing opportunities to engage with patrons in new ways, many libraries are using Facebook, Twitter, blogs, and other web 2.0 tools to communicate up-to-date news about resources, events, hours, research, and more. Further, when the global economic downturn is causing budget cuts in most institutions, web 2.0 technologies can succeed in times when other resources and platforms are out of reach.

Shortly after LUC 2008, Ralph Boé, director of LOUIS, suggested that a group of librarians meet at McNeese State University to discuss the possibilities of a “Statewide Web 2.0 initiative” (email from Boé to Blakeley, October 28, 2008). A group of librarians were identified by Ralph Boé and Sara Zimmerman at the October 2008 LUC conference because they had expressed interest in web 2.0 at the conference. The group met on November 20th, and included: Ralph Boé and Sara Zimmerman of Louis, Rebecca Blakeley, Jessica Hutchings, Debbie

Johnson-Houston, and Jerome Marcantel from McNeese State University, and Rebecca Miller and Sigrid Kelsey from Louisiana State University. With a common interest in using web 2.0 technologies and in identifying quality open-access resources for our constituents, the group discussed possibilities for collaboration, blending what they had already worked on into a broad resource for the entire state. The meeting resulted in a consensus to explore the uses and options associated with one particular web 2.0 technology, Delicious. A more detailed account of the meeting and instructions on tagging Delicious bookmarks is available in the Spring 2009 LOUIS Column of Louisiana Libraries.

Delicious

Delicious is a web-based method of saving bookmarks that makes one's bookmarks available from any computer or device with internet access. Delicious expands the potential of traditional browser-based bookmarks by making bookmarks web-based, searchable and organized by tags (keywords describing the bookmark). Tagging and searching creates a more nuanced organization system than traditional folder systems and makes it easier to find and maintain large numbers of bookmarks.

The social aspect of Delicious can be split into five methods of sharing. First, each user's public bookmarks (public is the default setting) can be accessed by using the following URL: http://delicious.com/username, where username is the login of the user. Because each Delicious account has a unique URL, a user’s bookmarks can be accessed by anyone, anywhere. Second, each user's public bookmarks are added to a directory of bookmarked webpages across the entire population of Delicious users. This directory can be searched or browsed by the most recently bookmarked items or by the most popular bookmarks and tags. Third, like many other social sites online, such as Facebook, Twitter, and MySpace, Delicious allows each user to add other users to his or her network. Users can view the bookmarks of people in their network all from one page. The fourth social method is direct sending. Each time a user bookmarks a new item, he or she can use the "send" field to send the bookmark directly to other Delicious users, Twitter users or the user's email contacts. Finally, Delicious creates RSS feeds for each user account and each tag combination, so that users are able to subscribe to bookmark feeds. Titles for bookmarks can be edited and Delicious users can add a description of each bookmarked page, which can be included in the feed.

Delicious in the Library

"Librarians are so careful about what is bought for the print collection, but then we watch our googlers race around the internet among unedited and ill-founded and repetitive single-page resources. [Using Delicious] is our attempt to select suitable material, to post it without delay, and to provide access points and comments on content," says school librarian Jan Radford, whose Delany Library (Delany College, Granville, Australia) is embracing Delicious, LibraryThing, and other social tools (Rethlefsen 2007, 26). Delany Library is not the only library embracing Delicious. A search within Delicious for the tag "del.icio.us_libraries" produces 157 links to libraries using Delicious to display online resources. These libraries present diverse resources, ranging from bookmarks of local utility companies to bookmarks of reading and learning games, research tutorials, and open access electronic journals.

At LSU Libraries, subject librarians maintain online guides for their respective subjects. Subject guides vary in depth and content, depending on the topic and the librarian who maintains it. After looking into Delicious, the potential for using it to help the subject librarians at LSU Libraries maintain their online subject guides quickly became evident. For the librarians who seldom create or maintain web pages, an online subject guide also means devoting time to maintaining a competency for HTML and other web technologies. Allowing the subject librarians to use Delicious to maintain their subject guides, therefore, would solve some of the time-intensive training associated with keeping the subject guides up-to-date. As well, it would enable the subject librarians to focus on what they excel at; identifying and organizing resources in their areas, while Delicious would accomplish the work of putting it into a web readable format. To use Delicious as a tool for maintaining the subject guides, the information from a Delicious account needs to display on the subject guide web pages, either by using exported information using the Delicious API or by means of a RSS feed. While, as discussed in the initial meeting, some librarians refer students directly to Delicious accounts, displaying the information on the LSU Libraries web site keeps the patrons from straying from the LSU Libraries' web site while viewing our recommended resources.

A literature and web search identified several libraries embedding Delicious feeds on subject guides. The College of New Jersey's subject guide for Italian Studies (http://www.tcnj.edu/~library/moulaison/ItalianStudies.html), for example, demonstrates the use of a Delicious RSS feed embedded in a subject guide. Styled to integrate with the web site colors and design, the feeds on The College of New Jersey's web site maintain the brand of the college. A librarian with a Delicious account can bookmark and tag web pages, consequently posting them immediately on the subject guide. The Delicious web site offers some information about including feeds on a web site, and this article has more detailed instructions for embedding a feed from a Delicious account and styling it with CSS (Appendix 1). This feed technology was shared among the LOUIS group, and both McNeese State University Library and LSU Libraries have incorporated Delicious feeds into several subject guides. For example, MSU's Gov Guides Wiki, a Wiki maintained at their library listing resources for research, displays Delicious bookmarks related to the Environmental Law at http://msugovguides.pbworks.com/Environmental+Law.

The Benefits of Delicious Subject Guides

Each new technology necessitates a process of evaluation to be sure that the costs of implementation, such as programming and staff instruction, do not outweigh the benefits received. At LSU Libraries, we found that on both the patron and librarian side Delicious generally improved the subject guide experience.

The Patron Side

Using the Delicious interface allows the patron to use multiple methods for filtering and searching resources. Patrons can use the provided search box to search the resource list by keyword. Librarian-provided tags allow the patron to filter the list of resources quickly and easily by selecting a combination of tags relevant to their work. These methods aid the patron in narrowing a list of resources from hundreds to five or ten with a few clicks. The RSS component of Delicious creates a mechanism for patrons to easily stay current with the updates and additions to the resource list. By subscribing to the RSS feed, patrons can receive a notification each time the librarian adds to or updates the resource list. For the patron with a Delicious account, collecting the resources they frequently use is as simple as signing in and clicking the save button next to the resource. This also allows patrons to organize the resources within the framework of their interests, or to research by adding their own tags and descriptions. Therefore, compared to the static web pages that the Delicious Subject Guide Project would enhance, Delicious would offer flexibility and easier navigation, in addition to advantages common to Web 2.0, like customization, transportability, and alerts.

The Library Side

The benefits of using Delicious to identify links on subject guides, as opposed to listing resources on a static web page, are numerous. Updating subject guides becomes faster and easier for a variety of reasons. For example, Delicious accounts may be accessed from any device with internet access, and saving a bookmark in Delicious is faster and easier than adding a link to a static HTML page. A wider range of staff, such graduate assistants, can be offered access to a Delicious account, given that they may not be able to have access to the library's or university's server. A variety of Delicious link-checkers are available for free on the internet, making monitoring and updating links easier as well. In addition to speed and accessibility for those maintaining the subject guides, feeds to Delicious bookmarks can be included on a variety of web pages, and used in different ways. For example, RSS feeds of Delicious accounts can be included on course management system pages, or given to faculty and students so that they might subscribe to the feed using their choice of RSS reader. Finally, using Delicious makes it easier for librarians to share their bookmarks with each other, offering the potential to build a shared resource. Overall, using Delicious can enable subject librarians to produce more while expending less effort.

Potential Disadvantages to Delicious

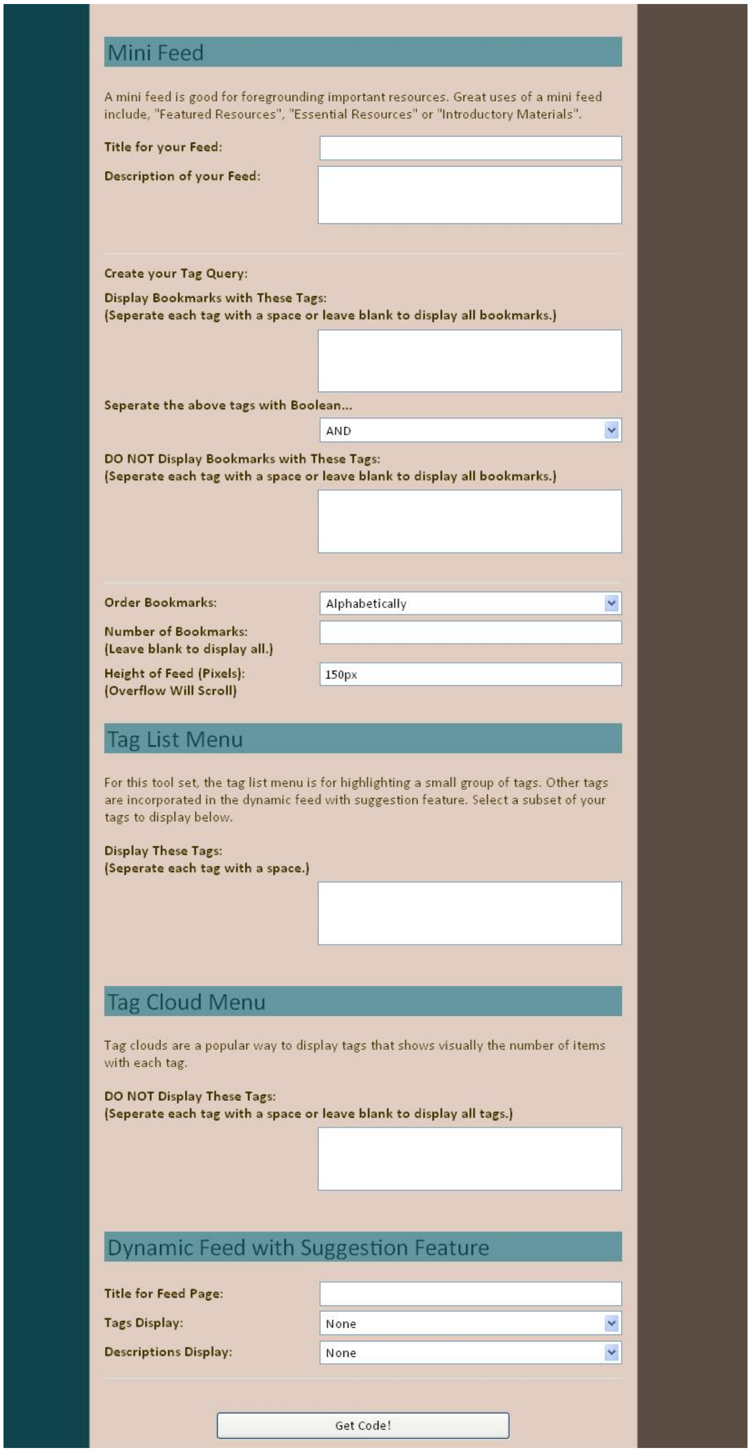
Using the Delicious interface and the embeddable link rolls, tag rolls and tag clouds has drawbacks. The primary disadvantage of using direct feeds from Delicious is the lack of flexibility in which the information can be displayed. As already demonstrated, the feeds can be styled for design, and they can also be sorted alphabetically or by date. Additionally, feeds can be filtered by tags, and one can choose whether to include a description of the web site bookmarked. Some other customization, like filtering feeds by tags, is also available, and feeds can be displayed in either a list or tag cloud format. The feeds, therefore, are fairly customizable, but do not offer the flexibility that having the data on one's own server offers, such as creating custom menus or tag rolls that link to and filter a feed while remaining on a library's web site. Another potential downside to feeding information from the Delicious web site is that one becomes reliant on another server to display one's content, potentially slowing loading time. Finally, Delicious limits their RSS outputs to the most recent one-hundred bookmarks, limiting those librarians with large subject areas. At LSU Libraries, exporting Delicious bookmarks into a database was a good solution to some of these problems.

Exporting and Manipulating Delicious Bookmarks

The easiest option for transferring Delicious bookmarks to another web server is by accessing the "Export/ Backup Bookmarks" on the Delicious web site. This tool creates a static web page of links, like many subject guides found on library web pages. Such a page offers potential to libraries with short lists of links, whose librarians want to save the time of setting up a web page. The static web page looses flexibility such as sorting and narrowing, so that it might not be suitable for a library with many bookmarks, other than for backup purposes.

MySQLicious, a PHP script that provides automated "backups" of Delicious bookmarks into a MySQL database, was a good choice for LSU initially, because we already use PHP/MySQL to search and display information on our web site. MySQLicious served well until Yahoo integrated its login parameters into Delicious accounts. As of the writing of this article, MySQLicious has not been updated to accommodate the new login feature, so VandenBroek wrote a program to replace MySQLicious (hereafter referred to as LSUlicious). Technical requirements to use LSUlicious, which is included as Appendix 3, include PHP 4 and some knowledge of PHP, MySQL, access to cron for scheduling (cron is a process to schedule tasks to run on a server), and a Delicious account. LSUlicious does not include the PHP to search and display the Delicious bookmarks from the database, so a library using LSUlicious for more than backing up a Delicious account needs to be prepared to set up a way for users to access the data. LSUlicious runs with cron jobs, so that a library can choose how often Delicious bookmarks are transferred to its database. At LSU Libraries, a LSUlicious script was set up for each subject librarian, which imports the respective librarians' Delicious bookmarks into a different table in one database each night.

VandenBroek was charged with exploring ways to search and display the database of bookmarks using PHP. VandenBroek created a variety of “tools” all available in one “toolbox” for the subject librarians to access and create subject guides. This idea was inspired by pay services such as LibGuides that feature coding-free subject guide creation. With the Subject Guide Toolbox (SGT), subject librarians are able to choose from templates and cut and paste the codes in order to set up their subject guides. Thus, our subject librarians are able to set up a subject guide with a few clicks and copy and paste it into a web site. Once complete, the links on the subject guide can be updated with the librarian’s Delicious account. This process requires virtually no web programming on the side of the librarian. The tools VandenBroek developed offer a variety of searching and displaying options that are not available with the feeds from Delicious. Additionally, she integrated several non-Delicious



Mini Feed

A mini feed is good for foregrounding important resources. Great uses of a mini feed include, "Featured Resources", "Essential Resources" or "Introductory Materials".

Title for your Feed:

Description of your Feed:

Create your Tag Query:

Display Bookmarks with These Tags:
(Seperate each tag with a space or leave blank to display all bookmarks.)

Seperate the above tags with Boolean... AND

DO NOT Display Bookmarks with These Tags:
(Seperate each tag with a space or leave blank to display all bookmarks.)

Order Bookmarks: Alphabetically

Number of Bookmarks:
(Leave blank to display all.)

Height of Feed (Pixels): 150px
(Overflow Will Scroll)

Tag List Menu

For this tool set, the tag list menu is for highlighting a small group of tags. Other tags are incorporated in the dynamic feed with suggestion feature. Select a subset of your tags to display below.

Display These Tags:
(Seperate each tag with a space.)

Tag Cloud Menu

Tag clouds are a popular way to display tags that shows visually the number of items with each tag.

DO NOT Display These Tags:
(Seperate each tag with a space or leave blank to display all tags.)

Dynamic Feed with Suggestion Feature

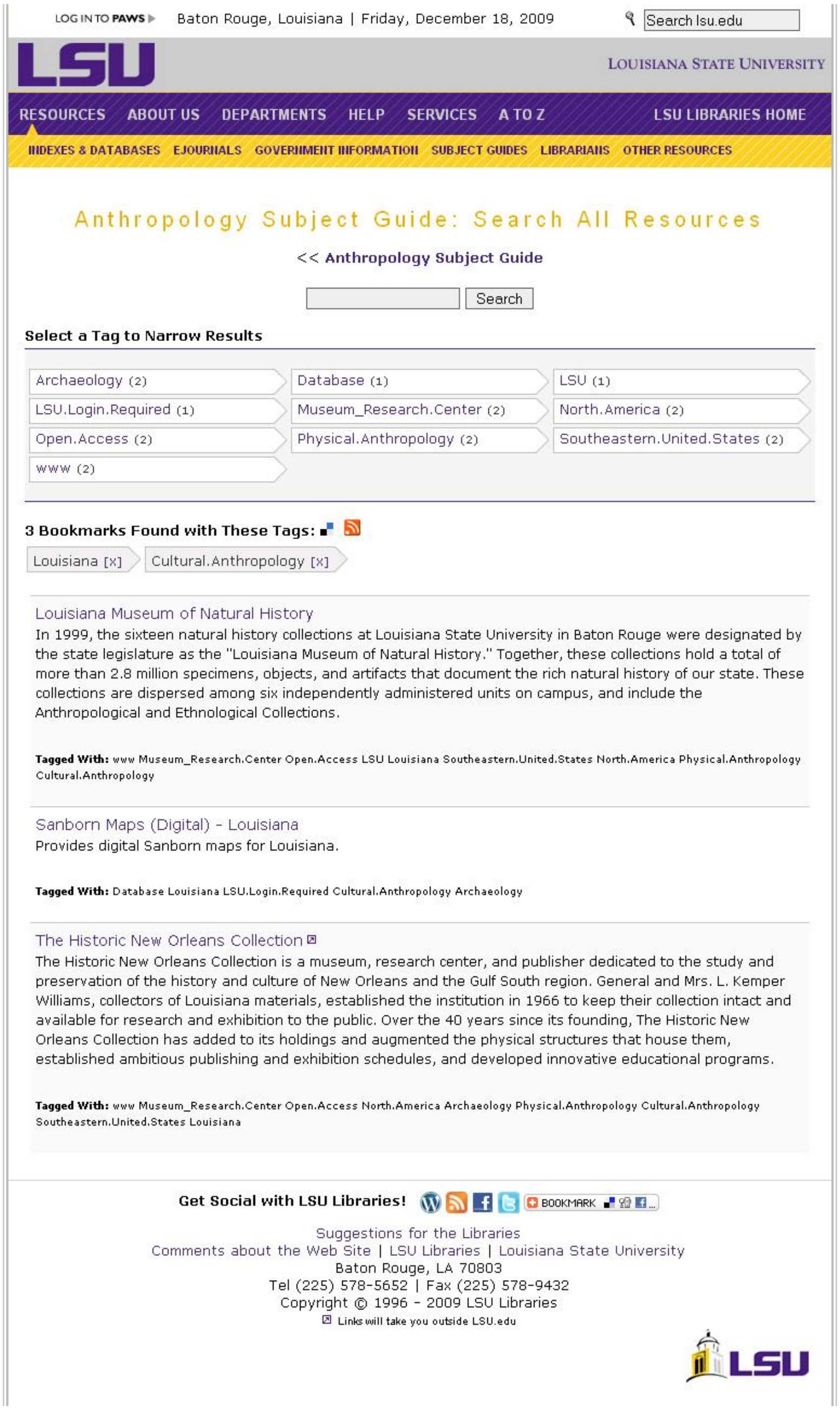
Title for Feed Page:

Tags Display: None

Descriptions Display: None

Figure 1. An example from the Subject Guide Tool box of tools the librarians can use generate code that creates feeds, tag clouds, and tag lists from their Delicious bookmarks

tools into the Subject Guide Toolbox, such as the LSU Libraries dynamic “I am on the Desk” script, so that subject librarians who have Reference Desk hours are able to use a tool to display their desk hours on their subject guide. The project has enabled our subject librarians to make better subject guides, while saving time and effort.



Baton Rouge, Louisiana | Friday, December 18, 2009

Anthropology Subject Guide: Search All Resources

<< Anthropology Subject Guide

Select a Tag to Narrow Results

Archaeology (2)
Database (1)
LSU (1)
LSU.Login.Required (1)
Museum_Research.Center (2)
North.America (2)
Open.Access (2)
Physical.Anthropology (2)
Southeastern.United.States (2)
www (2)

3 Bookmarks Found with These Tags:

Louisiana [x] Cultural.Anthropology [x]

Louisiana Museum of Natural History

In 1999, the sixteen natural history collections at Louisiana State University in Baton Rouge were designated by the state legislature as the "Louisiana Museum of Natural History." Together, these collections hold a total of more than 2.8 million specimens, objects, and artifacts that document the rich natural history of our state. These collections are dispersed among six independently administered units on campus, and include the Anthropological and Ethnological Collections.

Tagged With: www Museum_Research.Center Open.Access LSU Louisiana Southeastern.United.States North.America Physical.Anthropology Cultural.Anthropology

Sanborn Maps (Digital) - Louisiana

Provides digital Sanborn maps for Louisiana.

Tagged With: Database Louisiana LSU.Login.Required Cultural.Anthropology Archaeology

The Historic New Orleans Collection

The Historic New Orleans Collection is a museum, research center, and publisher dedicated to the study and preservation of the history and culture of New Orleans and the Gulf South region. General and Mrs. L. Kemper Williams, collectors of Louisiana materials, established the institution in 1966 to keep their collection intact and available for research and exhibition to the public. Over the 40 years since its founding, The Historic New Orleans Collection has added to its holdings and augmented the physical structures that house them, established ambitious publishing and exhibition schedules, and developed innovative educational programs.

Tagged With: www Museum_Research.Center Open.Access North.America Archaeology Physical.Anthropology Cultural.Anthropology Southeastern.United.States Louisiana

Get Social with LSU Libraries!

Suggestions for the Libraries
Comments about the Web Site | LSU Libraries | Louisiana State University
Baton Rouge, LA 70803
Tel (225) 578-5652 | Fax (225) 578-9432
Copyright © 1996 - 2009 LSU Libraries
Links will take you outside LSU.edu

Figure 2. The LSU Libraries Anthropology Subject Guide: A user has clicked on the “Louisiana” and “Cultural .Anthropology” tags to retrieve these bookmarks from the database. Note the links to the Delicious account and the RSS feed. Users can unclick tags to broaden the results, or narrow the results by clicking on more tags.

The Subject Guide Toolbox consists of two portions: the PHP/MySQL scripts that run all of the tools as they appear on each subject guide and the interface the librarians use to implement the various tools. The SGT interface comprises a series of forms that ask for the variables needed to make each tool customized to each librarian's needs. The variables consist of which Delicious account to pull from, which tags should be used to filter the display and various options on how descriptions and tags should be displayed. The PHP/MySQL scripts then use those variables to create the displays.

VandenBroek, who developed the toolbox, is completing a master's degree in anthropology, so with the anthropology librarian's permission, she set up a Delicious account and used the toolbox for the anthropology subject guides, the first subject guide published with the toolbox http://www.lib.lsu.edu/soc/anth. Several other subject librarians have implemented subject guides using the Subject Guide Toolbox, including Art and Design Librarian, Hillary Veeder and Biological Science Librarian, Jenna Ryan, whose subject guides are available at http://www.lib.lsu.edu/design/ and http://www.lib.lsu.edu/sci/bio/, respectively. VandenBroek and Kelsey have received positive feedback from the librarians using Delicious and the Subject Guide Toolbox. In general, the subject librarians appreciate the ease of using Delicious to identify and classify web sites, combined with the template in the Subject Guide Toolbox, which decreases the burden of web design. One librarian has suggested that tag clouds are counter-intuitive to what a librarian does, because they put the focus on the quantity of links rather than the quality.

Future developments could include making the code for the SGT and LSUlicious available open source. With the input of the librarians currently using the SGT, we are also experimenting with several new tools that could integrate other web 2.0 technologies and resource displays. For example, one project is looking at adding geographic tags that include longitude and latitude coordinates to be used with an SGT tool that would create a KML output, the language used for programming Google Earth and Google Maps, which would display bookmarked resources geographically. This is currently being implemented on the LSU Libraries' listing of Louisiana Libraries.

Moving Forward

Web 2.0 tools facilitate the sharing of resources and collaboration when building resources. Future development will benefit further if academic libraries in Louisiana collaborate on this effort, whether by contributing to the programming, or by contributing Delicious bookmarks to the effort. The meeting described in Blakeley's article laid out initial methods of sharing bookmarks, which should be evaluated. Additional methods for sharing bookmarks can also be explored. Resources for combining RSS feeds, like YahooPipes, are potential tools to combine efforts of several libraries. YahooPipes offers a way for users to combine and de-duplicate information from any variety of RSS feed or Delicious account. And while the subject guides at LSU have been a traditional aspect to the web site, an analysis of their usefulness would help determine what content should be included in them.

Conclusion

Regardless of the current trends in technology, librarians will continue to serve as an important resource for providing assistance in finding and evaluating information. The skilled assistance that Walters asserted was needed by patrons in search of information remains a pivotal role for librarians, and like then, libraries struggle with a lack of fiscal resources. Using web 2.0 technology and database technology have enabled the LSU Libraries subject librarians to address the continuing need to provide quality information in innovative, but fiscally sound ways. Librarians will continue to find methods of providing more assistance with less fiscal resources, especially in downward economic times. Web 2.0 technologies, database driven web sites, open access resources and open source computer code are some tools that Louisiana academic librarians use to provide their patrons with more quality information using fewer fiscal resources.

Sources Cited

DiNucci, D. "Fragmented Future," Print 53, no.4 (Jul/Aug 1999): 32, 221-222.

O'Reilly, T. "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," O'Reilly Media (2005). http://oreilly.com/web2/archive/what-is-web-20.html (accessed October 21, 2009).

Rethlefsen, M. L. "Tags Help Make Libraries Del.icio.us," Library Journal 132, no. 15 (2007): 26-28.

Blakeley, Rebecca. "Louis Column: Delicious Louisiana," Louisiana Libraries 71, no. 4 (Spring 2009): 22-23.

"Delicious: social bookmarking," http://delicious.com/ (accessed September 24, 2009).

"Italian-Language Resources," The College of New Jersey, http://www.tcnj.edu/~library/moulaison/ItalianStudies.html (accessed September 24, 2009).

"Mysqlicious: Mirror your Delicious bookmarks into a MySQL database," http://code.google.com/p/mysqlicious/ (accessed September 24, 2009).

Walter, Frank K. "Libraries in Land-Grant Colleges," The Journal of Higher Education 2, no. 6 (June 1931): 305-311.

Appendix 1.

Directions for embedding an RSS feed from a delicious account and styling it with CSS.

Setting up a Delicious Feed on a Web Page

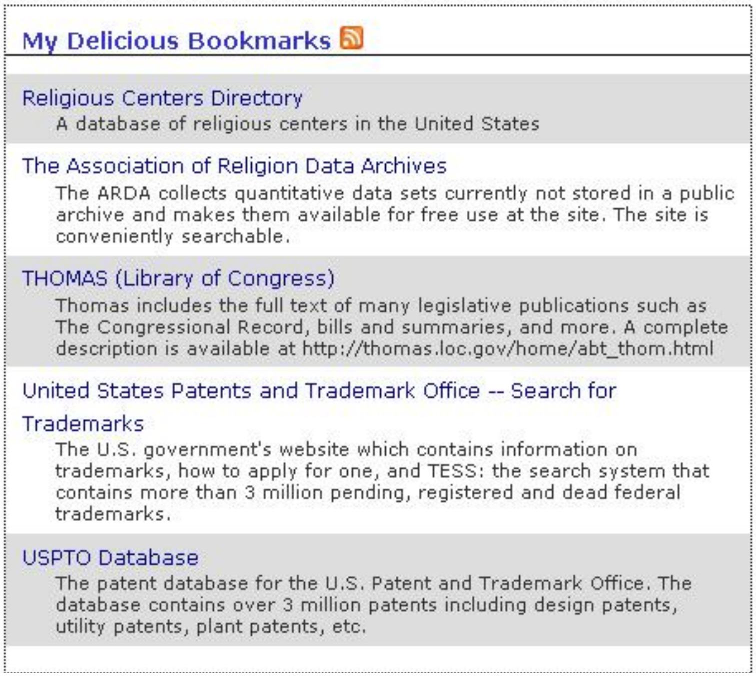
- After signing into a Delicious account, click "Settings" in the upper right hand of the screen.
- Click "Link Rolls" for a list of bookmarks, or "Tag Rolls" for a tag cloud.
- After clicking "Link Rolls," the following page presents a form to control some of the "display options" for a feed, as well as an input box for tags. The Tags input box should be used for displaying only the bookmarks with a specific tag. This can be used to categorize links in a subject guide, for example, by first displaying a feed of links tagged "databases," and under it, displaying another feed with links tagged "open source ejournals."
- Set the form options so that it displays in the format desirable,
- Styling the Feed with CSS
- If your library already uses a style sheet to designate the styles for links, fonts, and lists, your Delicious feed may be taken care of by the existing style sheets.

Here is a web page with a style sheet that can be used for additional styling:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Title of Document</title>

<style type="text/css">

/* ***************************************************************************************
Substitute delicioususername with the appropriate user name.  This is useful to apply
different styles on each feed in a page, based on username.
************************************************************************************* */
       #delicious-posts-delicioususername { }

/* ***************************************************************************************
This will style the entire feed.
************************************************************************************* */
       .delicious-posts {
              /* Background color of the box */
              background-color: white;

              /* The width of the box */
              width: 450px;
```

```
                /* The dotted border around the whole box */
                border: 1px black dotted;
        }


/* ***********************************************************************************
This styles the odd numbered posts.  Useful for separating links if they are more than
one line long.
*********************************************************************************** */
        .delicious-odd {
                /* The odd links are the ones with the gray background color. */
                background-color: #ddd;

                /* The text needs some padding, especially on the left and right. */
                padding: 3px 10px 3px 10px;
        }

        .delicious-even {
                /* the background color for the even-numbered links is white */
                background-color:#FFF;

                /* The padding should be the same as the odd links, to be consistent */
                padding: 3px 10px 3px 10px;
        }

/* ***********************************************************************************
This is the title of the feed.  The default is "My Delicous Bookmarks. The title can
be changed in the form on delicious, or by changing the words on the included script
in the body of the document.  Leave this off if you do not want your users to be able
to link to the delicious bookmarks at the delicious web site.
*********************************************************************************** */
        .delicious-posts h2 {
                /* The title needs some padding on the left to look good */
                padding-left: 10px;
                font-family: Verdana, Geneva, sans-serif;
                border-bottom: black solid 1px;
                font-size: 14px;
        }

/* ***********************************************************************************
This styles the link of the title.  The link goes to the Delicious user's page
*********************************************************************************** */
        .delicious-posts h2 a {
                text-decoration: none;
                color: #33C;
        }

/* ***********************************************************************************
This styles the links in the feed.
*********************************************************************************** */
         a.delicious-link, a:visited.delicious-link {
                color: #009;
                font-family: Verdana, Geneva, sans-serif;
                font-size: 12px;
                text-decoration: none;

        }

/* ***********************************************************************************
This underlines the links when they are hovered over.
```

```
************************************************************************************ */
        a:hover.delicious-link  {
                text-decoration: underline;
        }

/* ************************************************************************************
This styles the description of the links.
************************************************************************************ */
        p.delicious-extended {



                font-family:
        Verdana, Geneva, sans-
        serif;
                color: #333;
                font-size: 11px;
                padding-left: 20px;
        }

</style>

</head>

<body>

<script type="text/javascript"
src="http://feeds.delicious.com/
v2/js/siggles/database?title=My%
20Delicious%20Bookmarks&icon=rss
&count=5&sort=alpha&extended"></
script>

</body>
</html>
```



My Delicious Bookmarks

Religious Centers Directory
A database of religious centers in the United States

The Association of Religion Data Archives
The ARDA collects quantitative data sets currently not stored in a public archive and makes them available for free use at the site. The site is conveniently searchable.

THOMAS (Library of Congress)
Thomas includes the full text of many legislative publications such as The Congressional Record, bills and summaries, and more. A complete description is available at http://thomas.loc.gov/home/abt_thom.html

United States Patents and Trademark Office -- Search for Trademarks
The U.S. government's website which contains information on trademarks, how to apply for one, and TESS: the search system that contains more than 3 million pending, registered and dead federal trademarks.

USPTO Database
The patent database for the U.S. Patent and Trademark Office. The database contains over 3 million patents including design patents, utility patents, plant patents, etc.

Figure 3. An example of the finished styled feed using the Delicious javascript.

Appendix 2.

These instructions will set up PHP scripts that will work with a mysql database to display 1) a feed of the bookmarks 2) a suggestion box filled with relevant tags to filter the feed and 3) a search box. This is a sample portion of the Subject Guide Toolbox project from LSU. To see samples of the finished product and other tools from the Subject Guide Toolbox, visit these LSU subject guides:

- Anthropology Subject Guide - http://www.lib.lsu.edu/soc/anth/
- Art and Design Subject Guide - http://www.lib.lsu.edu/design/
- China Studies Subject Guide – http://www.lib.lsu.edu/soc/china/

Requirements

- PHP and MySQL

- Access to the root directory of your domain.

Step One: Create Delicious Yahoo! Accounts

Create a Delicious account using a Yahoo! login (this is the default for all new Delicious accounts). If you have an older Delicious account, then you can merge it with a Yahoo login by following these instructions.

- Create a new Yahoo! account at https://edit.yahoo.com/registration.
- Log into your old account at http://delicious.com.
- Click on "Settings" in the upper right hand corner of the Delicious page.
- Click on "Merge with Yahoo! Account" under "Account".
- Accept the terms of service and then click the "Agree and Start" button.
- Enter your Yahoo! login credentials.

Repeat this step for all of the Delicious accounts you would like to use.

Step Two: Obtain a Yahoo! API Key

You only need one key; you do not need a key for each account. To obtain the key follow these instructions.

- Point your internet browser to https://developer.apps.yahoo.com/dashboard/createKey.html
- Log in with one of your Yahoo! accounts.
- Name your application; this can be any name (e.g. <your library>licious).
- Select "Web-Based" under "Kind of Application".
- Type a brief description (e.g. An application that will collect <your library>'s Delicious bookmarks and save in a MySQL database.)
- Enter your websites' domain for "Application URL" and "Application Domain". This must be the root directory of your website.

  Example 1
  Library Homepage - http://www.university.edu/library/index.html
  Domain is - http://www.university.edu/
  Domain is NOT - http://www.university.edu/libary/

  Example 2
  Library Homepage – http://library.university.edu/index.html
  Domain is – http://library.university.edu/

- Leave "Favicon URL" blank.
- Select "This app requires access to private user data." Under "Access Scopes".
- In the list that appears change Delicious from "No Access Needed" to "Read/Write".
- Accept the Terms of Use.
- Click the "Get API Key" Button.

- Verify your domain. To do this you will need to save an empty file with the filename provided by Yahoo (e.g. htn--alfo8na.html) in your root directory of your domain. If you do not have access to your domain root directory, then you will need to contact your information technology department to have them create the file for you. This is a mandatory step.
- Once you have verified your domain, Yahoo! will provide you with the information that you will use to authenticate with Delicious. Save these items in a permanent secure location:
    - The username and password of the Yahoo! account you used to get the API key.
    - The Application ID
    - The Consumer Key
    - The Consumer Secret
    - The Application URL
- (If you lose this information you can obtain it again by logging into Yahoo! and pointing your browser to http://developer.yahoo.com/ and selecting "My Projects".)

Step Three: Setup LSUlicious MySQL Database

Create a MySQL database called "lsulicious" on your server. This table will store all of your Delicious bookmarks as well as your Yahoo! API access tokens

```
<?php

//Fill in your MySQL Username and Password Below.
$mysql_username = "<YOUR MYSQL USERNAME>";
$mysql_password = "<YOUR MYSQL PASSWORD>";

$con = mysql_connect("localhost","$mysql_username","$mysql_password");
if (!$con)   {die('Could not connect: ' . mysql_error());}
if (mysql_query("CREATE DATABASE lsulicious",$con))
  {echo "Database created";}
else  {echo "Error creating database: " . mysql_error();}
mysql_close($con);

?>
```

Then create the following tables in the lsulicious database.

- Table: Oauth

Column Name	**Data Type**
(Primary Key) ID	INTEGER - Not Null & Auto Increment
nonce	VARCHAR(45)
request_token	VARCHAR(45)
request_token_secret	VARCHAR(45)
duser	VARCHAR(45)
verifier	VARCHAR(45)

PHP/MySQL Code to create this table

```
<?php

//Fill in your MySQL Username and Password Below.
$mysql_username = "YOUR MYSQL USERNAME";
$mysql_password = "YOUR MYSQL PASSWORD";


//  MYSQL CONNECTION CODE
//Connects to mysql.
$con = mysql_connect("localhost","$mysql_username","$mysql_password");

//If the connection fails, give mysql error.
If (!$con) {die(‘Could Not Connect: ‘ . mysql_error());}

//Connect to the mysql database.  If the connection fails, give mysl error.
Mysql_select_db(lsulicious, $con) or die("Could Not Open lsulicious " .
mysql_error());
}

$q = CREATE TABLE oauth
(
ID int NOT NULL AUTO_INCREMENT
PRIMARY KEY(ID),
nonce VARCHAR(45),
request_token VARCHAR(45),
request_token_secret VARCHAR(45),
duser VARCHAR(45),
verifier VARCHAR(45)
)
mysql_query($sql,$con);

?>
```

- Table: <delicious username>
  Make a table for each delicious account with these fields. Name each table with the delicious username.

Column Name	**Data Type**
ID	INTEGER - Not Null & Auto Increment
url	TEXT
description	TEXT
extended	TEXT
date	DATETIME
hash	VARCHAR(255)

PHP/MySQL Code to create this table

```
<?php

//Fill in your MySQL Username, Password and Delicious Username Below.
$mysql_username = "YOUR MYSQL USERNAME";
$mysql_password = "YOUR MYSQL PASSWORD";
```

```
$delicious_username = "YOUR DELICIOUS USERNAME"


//  MYSQL CONNECTION CODE
//Connects to mysql.
$con = mysql_connect("localhost","$mysql_username","$mysql_password");

//If the connection fails, give mysql error.
If (!$con) {die('Could Not Connect: ' . mysql_error());}

//Connect to the mysql database.  If the connection fails, give mysl error.
Mysql_select_db(lsulicious, $con) or die("Could Not Open lsulicious " .
mysql_error());
}

$q = CREATE TABLE $delicious_username
(
ID int NOT NULL AUTO_INCREMENT
PRIMARY KEY(ID),
url TEXT,
description TEXT,
extended TEXT,
date TEXT,
hash VARCHAR(255)
)
mysql_query($sql,$con);

?>
```

Step Four: Setup PHP Pages

Create the PHP pages that will run LSUlicious and the Subject Guide Toolbox.

- Create a PHP page called "sgt.php". This is the Subject Guide Toolbox file, which will contain all of the functions used to display the feed, tag menu and search box on your final page.
- Create a PHP page called "lsulicious.php". This is the file that connects to Delicious and copies the bookmarks to a MySQL database.
- Create a PHP page for each delicious account named with the delicious username (i.e. "yourdelicioususername.php"). These are the files that will be connected with the CRON.

Step Five: Setup the Subject Guide Toolbox

Open the "sgt.php" page in your editor program. Paste this code at the beginning of the document:

```
<?php
//
// Code created by Angela VandenBroek, LSU Libraries
//

//////// BEGIN VARIABLES - THIS IS THE ONLY PORTION THAT REQUIRES ANY EDITING OR INPUT

//Replace USERNAME with the username for your mysql database.
```

```
$mysql_username = "USERNAME";

// Replace PASSWORD with the password for your mysql database.
$mysql_password =  "PASSWORD";

// Replace DATABASE with the name of your mysql database.
$mysql_database = "DATABASE";

//Replace DELICIOUSICON with the url where your Delicious icon image is saved.
$delicious_icon = "DELICIOUSICON";

//Replace RSSICON with the url where your RSS icon image is saved.
$rss_icon = "RSSICON";
```

Replace the capitalized letters with the appropriate information. Then paste the code below in the document under the variables listed above. This code does not require any editing.

```
///////////////////////////////////////////// DO NOT EDIT CODE BELOW THIS POINT

////////// BEGIN CONNECTION FUNCTION
Connect() {
global $mysql_username;
global $mysql_password;
global $database;

//  MYSQL CONNECTION CODE
//Connects to mysql.
$con = mysql_connect("localhost","$mysql_username","$mysql_password");

//If the connection fails, give mysql error.
If (!$con) {die(‘Could Not Connect: ‘ . mysql_error());}
//Connect to the mysql database.  If the connection fails, give mysl error.
Mysql_select_db($mysql_database, $con) or die("Could Not Open $mysql_database" .
mysql_error());
}

//BEGIN FEED OF DELICIOUS BOOKMARKS FUNCTION
Feed() {

// CAPTURE USER TAG CHOICES FROM URL
$tagitem = $_GET["tagitem"];
$tagitem2 = $_GET["tagitem2"];
$tagitem3 = $_GET["tagitem3"];
$tagitem4 = $_GET["tagitem4"];
$tagitem5 = $_GET["tagitem5"];
$searchbox = $_GET["searchbox"];

//The mysql_real_escape_string function clears special characters from the search box
entry to prevent errors.
$escapedsb = mysql_real_escape_string($searchbox);

//CAPTURE DISPLAY PREFERENCE VARIABLES
//These variables are defined in the code snippet that is on the main page.
//Each variable customizes the way the bookmark feed will display.
global $delicious_icon;
global $rss_icon;
global $duser;
global $feedloc;
global $tagsdisplay;
global $descdisplay;
global $basetag;

// CREATE MYSQL QUERY BASED ON USER SELECTED TAGS AND OTHER SETTINGS
if ($basetag == "" && $searchbox == "" && $tagitem == "") {
$query = "SELECT * FROM $duser ORDER BY Description";
}
elseif (
$basetag != "" && $searcbox == "" && $tagitem == "") {$query = "SELECT * FROM $duser
where (tags like '% $basetag %' || tags like '$basetag %' || tags like '% $basetag' ||
tags='$basetag') ORDER BY Description";
}
else {
$query = "SELECT * FROM $duser where ";
if ($basetag != "") {$query .= "(tags like '% $basetag %' || tags like '$basetag %' ||
tags like '% $basetag' || tags='$basetag')";}
if ($basetag != "" && $searchbox != "") {$query .= " && (tags like '%$escapedsb%' ||
description like '%$escapedsb%' || extended like '%$escapedsb%' || url like
'%$escapedsb%')";}
elseif ($searchbox != "") {$query .= "(tags like '%$escapedsb%' || description like
'%$escapedsb%' || extended like '%$escapedsb%' || url like '%$escapedsb%')";}
if ($tagitem != "" && $basetag != "" || $searchbox != "") {$query .= " && (tags like
'% $tagitem %' || tags like '$tagitem %' || tags like '% $tagitem' ||
tags='$tagitem')";}
elseif ($tagitem != "") {$query .= "(tags like '% $tagitem %' || tags like '$tagitem
%' || tags like '% $tagitem' || tags='$tagitem')";}
if ($tagitem2 != "") {$query .= " && (tags like '% $tagitem2 %' || tags like
'$tagitem2 %' || tags like '% $tagitem2' || tags='$tagitem2')";}
if ($tagitem3 != "") {$query .= " && (tags like '% $tagitem3 %' || tags like
'$tagitem3 %' || tags like '% $tagitem3' || tags='$tagitem3')";}
if ($tagitem4 != "") {$query .= " && (tags like '% $tagitem4 %' || tags like
'$tagitem4 %' || tags like '% $tagitem4' || tags='$tagitem4')";}
if ($tagitem5 != "") {$query .= " && (tags like '% $tagitem5 %' || tags like
'$tagitem5 %' || tags like '% $tagitem5' || tags='$tagitem5')";}
$query .= " ORDER BY Description";
}
$result = mysql_query($query);

//CREATE LINK TO DELICIOUS AND RSS WITH TAGS AND SEARCH SELECTED
if ($searchbox != "") {
$deliciouslink = "http://delicious.com/search?p=";
$sb = explode(" ",$escapedsb);
foreach ($sb as $sbkeyword) {$deliciouslink .= "+".$sbkeyword;}
$deliciouslink .= "&chk=&context=userposts|".$duser."|";
if ($basetag != "") {$deliciouslink .= "$basetag";}
if ($tagitem != "" && $basetag != "") {$deliciouslink .= "+$tagitem";}
elseif ($tagitem != "") {$deliciouslink .= "$tagitem";}
if ($tagitem2 != "") {$deliciouslink .= "+$tagitem2";}
if ($tagitem3 != "") {$deliciouslink .= "+$tagitem3";}
if ($tagitem4 != "") {$deliciouslink .= "+$tagitem4";}
if ($tagitem5 != "") {$deliciouslink .= "+$tagitem5";}
$deliciouslink .= "&fr=del_icio_us&lc=";
}
else {
$deliciouslink = "http://delicious.com/".$duser."/";
if ($basetag != "") {$deliciouslink .= "$basetag";}
if ($tagitem != "" && ($basetag != "")) {$deliciouslink .= "+$tagitem";}
elseif ($tagitem != "") {$deliciouslink .= "$tagitem";}
if ($tagitem2 != "") {$deliciouslink .= "+$tagitem2";}
if ($tagitem3 != "") {$deliciouslink .= "+$tagitem3";}
if ($tagitem4 != "") {$deliciouslink .= "+$tagitem4";}
if ($tagitem5 != "") {$deliciouslink .= "+$tagitem5";}
```

```
}

$rsslink = "http://feeds.delicious.com/v2/rss/".$duser."/";
if ($basetag != "") {$rsslink .= "$basetag";}
if ($tagitem != "" && ($basetag != "")) {$rsslink .= "+$tagitem";}
elseif ($tagitem != "") {$rsslink .= "$tagitem";}
if ($tagitem2 != "") {$rsslink .= "+$tagitem2";}
if ($tagitem3 != "") {$rsslink .= "+$tagitem3";}
if ($tagitem4 != "") {$rsslink .= "+$tagitem4";}
if ($tagitem5 != "") {$rsslink .= "+$tagitem5";}

//CREATE CURRENTLY FILTERING WITH DISPLAY
if ($searchbox == "" && $tagitem == "") {$currentdisplay = "<em>No Tags
Selected</em>";}
else {
if ($searchbox != "") {
$currentdisplay = "<p class=\"tag list\">Current Search: $searchbox <span
class=\"deletetag\"><a href=\"".$feedloc;
if ($tagitem != "") {$currentdisplay .= "tagitem=$tagitem";}
if ($tagitem2 != "") {$currentdisplay .= "&tagitem2=$tagitem2";}
if ($tagitem3 != "") {$currentdisplay .= "&tagitem3=$tagitem3";}
if ($tagitem4 != "") {$currentdisplay .= "&tagitem4=$tagitem4";}
if ($tagitem5 != "") {$currentdisplay .= "&tagitem5=$tagitem5";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
if ($tagitem != "") {
$currentdisplay .= "<p class=\"tag list\">$tagitem <span class=\"deletetag\"><a
href=\"".$feedloc;
if ($searchbox != "") {$currentdisplay .= "searchbox=$searchbox";}
if ($tagitem2 != "") {$currentdisplay .= "&tagitem=$tagitem2";}
if ($tagitem3 != "") {$currentdisplay .= "&tagitem2=$tagitem3";}
if ($tagitem4 != "") {$currentdisplay .= "&tagitem3=$tagitem4";}
if ($tagitem5 != "") {$currentdisplay .= "&tagitem4=$tagitem5";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
if ($tagitem2 != "") {
$currentdisplay .= "<p class=\"tag list\">$tagitem2 <span class=\"deletetag\"><a
href=\"".$feedloc;
if ($searchbox != "") {$currentdisplay .= "searchbox=$searchbox";}
if ($tagitem != "") {$currentdisplay .= "tagitem=$tagitem";}
if ($tagitem3 != "") {$currentdisplay .= "&tagitem2=$tagitem3";}
if ($tagitem4 != "") {$currentdisplay .= "&tagitem3=$tagitem4";}
if ($tagitem5 != "") {$currentdisplay .= "&tagitem4=$tagitem5";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
if ($tagitem3 != "") {
$currentdisplay .= "<p class=\"tag list\">$tagitem3 <span class=\"deletetag\"><a
href=\"".$feedloc;
if ($searchbox != "") {$currentdisplay .= "searchbox=$searchbox";}
if ($tagitem != "") {$currentdisplay .= "tagitem=$tagitem";}
if ($tagitem2 != "") {$currentdisplay .= "&tagitem2=$tagitem2";}
if ($tagitem4 != "") {$currentdisplay .= "&tagitem3=$tagitem4";}
if ($tagitem5 != "") {$currentdisplay .= "&tagitem4=$tagitem5";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
if ($tagitem4 != "") {
$currentdisplay .= "<p class=\"tag list\">$tagitem4 <span class=\"deletetag\"><a
href=\"".$feedloc;
if ($searchbox != "") {$currentdisplay .= "searchbox=$searchbox";}
if ($tagitem != "") {$currentdisplay .= "tagitem=$tagitem";}
if ($tagitem2 != "") {$currentdisplay .= "&tagitem2=$tagitem2";}
if ($tagitem3 != "") {$currentdisplay .= "&tagitem3=$tagitem3";}
if ($tagitem5 != "") {$currentdisplay .= "&tagitem4=$tagitem5";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
if ($tagitem5 != "") {
$currentdisplay .= "<p class=\"tag list\">$tagitem5 <span class=\"deletetag\"><a
href=\"".$feedloc;
if ($tagitem != "") {$currentdisplay .= "tagitem=$tagitem";}
if ($tagitem2 != "") {$currentdisplay .= "&tagitem2=$tagitem2";}
if ($tagitem3 != "") {$currentdisplay .= "&tagitem3=$tagitem3";}
if ($tagitem4 != "") {$currentdisplay .= "&tagitem4=$tagitem4";}
$currentdisplay .= "\" title=\"Remove Tag\">[X]</a></span></p>";
}
}

//RUN MYSQL LOOP TO CREATE LIST OF DELICIOUS BOOKMARKS

$RowColor = 0;
$rowcount = mysql_num_rows($result);
echo "<div class='sgtbox'>";
print "<h2>$rowcount Bookmarks Found with These Tags:<a href='$deliciouslink'><img
class='deliciousicon' src='$delicious_icon’ /></a><a href='$rsslink'><img
class='deliciousicon' src='$rss_icon' /></a></h2> $currentdisplay <br clear=\"all\"
/><br />";

while ($item = mysql_fetch_array($result)){
$RowColor = ($RowColor == "even") ? "odd" : "even";
echo "<div class='deliciousitem $RowColor'"; if ($descdisplay == "tooltip") {echo
"title=\"".$item["extended"]."\"";} elseif ($tagsdisplay == "tooltip") {echo
"title=\"".$item["tags"]."\"";} echo ">";
echo "<p class='delicioustitle'><a
href='".$item["url"]."'>".$item["description"]."</a></p>";
if ($descdisplay == "text") echo "<p
class='deliciousextended'>".$item["extended"]."</p>";
elseif ($descdisplay == "moreinfo") echo "<p class='deliciousextendedmore'> <span
class=\"moreinfo\">More Information</span> <br />".$item["extended"]."</p>";
else echo "";
if ($tagsdisplay == "text") echo "<p class='delicioustags'><strong>Tagged
With:</strong> ".$item["tags"]."</p>";
else echo "";
echo "</div>";
}
mysql_close($link);
echo "</div>";

}

////////// BEGIN SUGGESTION FEATURE FUNCTION
Suggest() {

// CAPTURE USER TAG CHOICES FROM URL
$tagitem = $_GET["tagitem"];
$tagitem2 = $_GET["tagitem2"];
$tagitem3 = $_GET["tagitem3"];
$tagitem4 = $_GET["tagitem4"];
$tagitem5 = $_GET["tagitem5"];
$searchbox = $_GET["searchbox"];
$escapedsb = mysql_real_escape_string($searchbox);

//CAPTURE DISPLAY PREFERENCE VARIABLES
global $duser;
global $dusertags;
global $feedloc;
global $basetag;

//FIND TAGS THAT ARE ASSOCIATED WITH CURRENTLY SELECTED TAGS
$searchbox_query = "(tags like '%$escapedsb%' || description like '%$escapedsb%' ||
extended like '%$escapedsb%' || url like '%$escapedsb%')";
$basetag_query = "(tags like '% $basetag %' || tags like '$basetag %' || tags like '%
$basetag' || tags='$basetag')";
$tagitem_query = "(tags like '% $tagitem %' || tags like '$tagitem %' || tags like '%
$tagitem' || tags='$tagitem')";
$tagitem2_query = "(tags like '% $tagitem2 %' || tags like '$tagitem2 %' || tags like
'% $tagitem2' || tags='$tagitem2')";
$tagitem3_query = "(tags like '% $tagitem3 %' || tags like '$tagitem3 %' || tags like
'% $tagitem3' || tags='$tagitem3')";
$tagitem4_query = "(tags like '% $tagitem4 %' || tags like '$tagitem4 %' || tags like
'% $tagitem4' || tags='$tagitem4')";
$tagitem5_query = "(tags like '% $tagitem5 %' || tags like '$tagitem5 %' || tags like
'% $tagitem5' || tags='$tagitem5')";



$sql = "SELECT tags FROM $duser";
if ($basetag != "" || (isset($_GET["searchbox"]) || isset($_GET["tagitem"]) ||
isset($_GET["tagitem2"]) || isset($_GET["tagitem3"]) || isset($_GET["tagitem4"]) ||
isset($_GET["tagitem5"]))) {$sql .= " where ";}
if ($basetag != "") {$sql .= "$basetag_query ";}
if ($basetag != "" && (isset($_GET["searchbox"]) || isset($_GET["tagitem"]) ||
isset($_GET["tagitem2"]) || isset($_GET["tagitem3"]) || isset($_GET["tagitem4"]) ||
isset($_GET["tagitem5"]))) {$sql .= "&& ";}
if (isset($_GET["searchbox"])) {$sql .= "$searchbox_query ";}
if (isset($_GET["searchbox"]) && (isset($_GET["tagitem"]) || isset($_GET["tagitem2"])
|| isset($_GET["tagitem3"]) || isset($_GET["tagitem4"]) || isset($_GET["tagitem5"])))
{$sql .= "&& ";}
if (isset($_GET["tagitem"])) {$sql .= "$tagitem_query ";}
if (isset($_GET["tagitem"]) && (isset($_GET["tagitem2"]) || isset($_GET["tagitem3"])
|| isset($_GET["tagitem4"]) || isset($_GET["tagitem5"]))) {$sql .= "&& ";}
if (isset($_GET["tagitem2"])) {$sql .= "$tagitem2_query ";}
if (isset($_GET["tagitem2"]) && (isset($_GET["tagitem3"]) || isset($_GET["tagitem4"])
|| isset($_GET["tagitem5"]))) {$sql .= "&& ";}
if (isset($_GET["tagitem3"])) {$sql .= "$tagitem3_query ";}
if (isset($_GET["tagitem3"]) && (isset($_GET["tagitem4"]) ||
isset($_GET["tagitem5"]))) {$sql .= "&& ";}
if (isset($_GET["tagitem4"])) {$sql .= "$tagitem4_query ";}
if (isset($_GET["tagitem4"]) && (isset($_GET["tagitem5"]))) {$sql .= "&& ";}
if (isset($_GET["tagitem4"])) {$sql .= "$tagitem5_query ";}

echo "<h2>Select a Tag to Narrow Results</h2>";
echo "<div class=\"suggestbox\"><br />";
$result = mysql_query($sql);

//CREATE ZERO RESULTS ERROR
$count = mysql_num_rows($result);
if ($count == 0) {
echo "Sorry, results cannot be narrowed further.  <a href=\"$feed_feedloc\">{{ Start
Over }}</a>";
}

//IF THERE ARE RESULTS THEN DISPLAY
else {
$tags = array();
```

```
while ($row = mysql_fetch_assoc($result)) {
$lines = explode(" ", $row['tags']);
foreach($lines as $lined) {
$tags[] = $lined;
}
}
$uniquelines = array_unique($tags);
sort($uniquelines);
$numtags = count($uniquelines);

if (isset($_GET["tagitem"], $_GET["tagitem2"], $_GET["tagitem3"], $_GET["tagitem4"],
$_GET["tagitem5"])) {
$numtags = $numtags-5;
}
elseif (isset($_GET["tagitem"], $_GET["tagitem2"], $_GET["tagitem3"],
$_GET["tagitem4"])) {
$numtags = $numtags-4;
}
elseif (isset($_GET["tagitem"], $_GET["tagitem2"], $_GET["tagitem3"])) {
$numtags = $numtags-3;
}
elseif (isset($_GET["tagitem"], $_GET["tagitem2"])) {
$numtags = $numtags-2;
}
elseif (isset($_GET["tagitem"])) {
$numtags = $numtags-1;
}


if ($numtags == 0) {
echo "Sorry, results cannot be narrowed further.  <a href=\"$feed_feedloc\">{{ Start
Over }}</a>";
}
else {
foreach($uniquelines as $line){
if ($line == $tagitem) echo "";
elseif ($line == $tagitem2) echo "";
elseif ($line == $tagitem3) echo "";
elseif ($line == $tagitem4) echo "";
elseif ($line == $tagitem5) echo "";
elseif ($line == $feed_basetag) echo "";
else {
$sqlcount = "SELECT * FROM $feed_duser WHERE (tags like '% $line %' || tags like
'$line %' || tags like '% $line' || tags = '$line')";
if ($feed_basetag != "") {$sqlcount .= "&& $basetag_query";}
if (isset($_GET["searchbox"])) {$sqlcount .= "&& $searchbox_query";}
if (isset($_GET["tagitem"])) {$sqlcount .= "&& $tagitem_query";}
if (isset($_GET["tagitem2"])) {$sqlcount .= "&& $tagitem2_query";}
if (isset($_GET["tagitem3"])) {$sqlcount .= "&& $tagitem3_query";}
if (isset($_GET["tagitem4"])) {$sqlcount .= "&& $tagitem4_query";}
if (isset($_GET["tagitem5"])) {$sqlcount .= "&& $tagitem5_query";}
$resultcount = mysql_query($sqlcount);
$tagcount = mysql_num_rows($resultcount);
echo "<span class=\"tag\"><a href=\"".$feed_feedloc;
if (isset($_GET["searchbox"])) {echo "searchbox=".$_GET["searchbox"]."&";}
if (!isset($_GET["tagitem"])) {echo "tagitem=".$line;}
else {echo "tagitem=".$_GET["tagitem"];
if (!isset($_GET["tagitem2"])) {echo "&tagitem2=".$line;}
else {echo "&tagitem2=".$_GET["tagitem2"];
if (!isset($_GET["tagitem3"])) {echo "&tagitem3=".$line;}
else {echo "&tagitem3=".$_GET["tagitem3"];
if (!isset($_GET["tagitem4"])) {echo "&tagitem4=".$line;}
else {echo "&tagitem4=".$_GET["tagitem4"];
if (!isset($_GET["tagitem5"])) {echo "&tagitem5=".$line;}
else {echo "&tagitem5=".$_GET["tagitem5"];}
}}}}
echo "\">".$line."</a>";
echo " <span class=\"tagcount\">(".$tagcount.")</span></span>";
}
}
}}
echo "<br clear=\"all\" /><br /></div><br />";
}

function searchbox(){

	global $searchbox_feedloc;

	echo "<form class=\"center\" action=\"/$searchbox_feedloc#feed\" method=\"get\"
	name=\"delsearchbox\"><input  name=\"searchbox\" style=\"width:140px\"
	type=\"text\" />&nbsp;<input name=\"search\" type=\"submit\" value=\"Search\"
	/></form><br />";

}

?>
```

Step Six: Setup LSUlicious

Open the "lsulicious.php" file in your editor program. Paste the code below in the document. Under "Connection" replace "FILEPATH" with the file path to the sgt.php file. Under "Setup the Variables" fill in the capitalized words the consumer key and share secret provided by Yahoo! in step two above. Also under "Setup the Variables", fill in the file path to the lsulicious.php file.

```
<?
///////////////
//Connection.//
///////////////

	//Connect to the Subject Guide Toolbox to use the connect function.
	include("FILEPATH/sgt.php");

	//Connect to the lsulicious MySQL database.
	connect();

/////////////////////////
//Setup the Variables. //
/////////////////////////

//Create a random string for the nonce variable.
	function getUniqueCode($length = "10") {
		$code = md5(uniqid(rand(), true));
		if ($length != "") return substr($code, 0, $length);
		else return $code;
	}
	$nonce = getUniqueCode();

//Fill in the variables below with your consumer key, shared secret, and the filepath
//for this file.
```

```
       $consumer_key = "YOUR CONSUMER KEY";
       $secret = "YOUR SHARED SECRET";
       $callback = "FILE PATH"."LSUlicious.php?nonce=".$nonce;

//Get the current time in seconds since the Unix Epoch (January 1, 1970 00:00:00 GMT)
       $timestamp = time();

//Get the Delicious Username (Provided by the CRON PHP files) and check for previous
//authentications.
       $duser = $_GET["duser"];
       $q = "SELECT * FROM oauth WHERE duser='$duser'";
       $result = mysql_query($q);
       $ifduser = mysql_num_rows($result);

///////////////////////////////////////////////////////////////////////////////////////
//If this delicious user has already been logged in then skip login and request
//credentials.
///////////////////////////////////////////////////////////////////////////////////////

if($ifduser == 1 && $duser != "") {

///////////Steps refer to the OAUTH Delicious Instructions
(http://delicious.com/help/oauthapi)
       //STEP 4 [Request Access Token] -----------------------------------------------

       //Collect the neccesary variables and construct the API URL
       while($row = mysql_fetch_array($result)) {

              $verifier = $row["verifier"];
              $request_token = $row["request_token"];

              $url = "https://api.login.yahoo.com/oauth/v2/get_token?";
              $url.= "oauth_consumer_key=".$consumer_key;
              $url.= "&oauth_signature_method=PLAINTEXT";
              $url.= "&oauth_version=1.0";
              $url.= "&xoauth_lang_pref=en-us";
              $url.= "&oauth_verifier=".$verifier;
              $url.= "&oauth_token=".$request_token;
              $url.= "&oauth_timestamp=".$timestamp;
              $url.= "&oauth_nonce=".nonce;
              $url.= "&oauth_signature=".$secret."%26".$row["request_token_secret"];

              $duser = $row["duser"];
       }
       //Make the API Call to Yahoo!
              $ch = curl_init();
              curl_setopt($ch, CURLOPT_URL, $url);
              curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
              $response = curl_exec($ch);
              curl_close($ch);

       //Seperate the variables received from Yahoo! and store in variables.
              $responses = explode("&", $response);

              foreach($responses as $item) {
                     $items = explode("=", $item);
                     if ($items[0] == "oauth_token") {
                            $oauth_token = $items[1];
                     }
                     if ($items[0] == "oauth_token_secret") {
                            $oauth_token_secret = $items[1];
                     }
                     if ($items[0] == "oauth_expires_in") {
                            $oauth_expires_in = $items[1];
                     }
                     if ($items[0] == "oauth_session_handle") {
                            $oauth_session_handle = $items[1];
                     }
                     if ($items[0] == "oauth_authorization_expires_in") {
                            $oauth_authorization_expires_in = $items[1];
                     }
                     if ($items[0] == "xoauth_yahoo_guid") {
                            $xoauth_yahoo_guid = $items[1];
                     }
              }

       //STEP 5 [Build the Base String] ----------------------------------------------

       //Collect the necessary variables and construct the basestring (URL and
       //Parameters)
              $apiurl = "http://api.del.icio.us/v2/posts/all";
              $params = "oauth_consumer_key=".$consumer_key;
              $params .= "&oauth_nonce=".$nonce;
              $params .= "&oauth_signature_method=HMAC-SHA1";
              $params .= "&oauth_timestamp=".$timestamp;
              $params .= "&oauth_token=".$oauth_token;
              $params .= "&oauth_version=1.0";

       //URL Encode the URL and Parameters
              $urlencoded_apiurl = urlencode($apiurl);
              $urlencoded_params = urlencode($params);

       //Combine the GET method, Encoded URL and Encoded Parameters.
              $basestring = "GET&".$urlencoded_apiurl."&".$urlencoded_params;

       //STEP 6 [Generate the Signature] ---------------------------------------------

       //Create the HMAC Signature using functions from the PHP library.
              $signature = base64_encode(hash_hmac("sha1", $basestring,
              $secret."&".$oauth_token_secret, true));

       //STEP 7 [Create Request to Delicious] ----------------------------------------

       //Construct the Delicious API URL.
              $url = "http://api.del.icio.us/v2/posts/all";

       //Create the header array with appropriate variables and HMAC signature.
              $headers = array(
                     "Authorization: OAuth
                     realm=\"yahooapis.com\",
                     oauth_consumer_key=\"".$consumer_key."\",
                     oauth_nonce=\"".$nonce."\",
                     oauth_signature_method=\"HMAC-SHA1\",
                     oauth_timestamp=\"".$timestamp."\",
                     oauth_token=\"".$oauth_token."\",
                     oauth_version=\"1.0\",
                     oauth_signature=\"".$signature."\""
              );

       //Make API Call to Delicious.
              $ch = curl_init();
              curl_setopt($ch, CURLOPT_URL, $url);
```

```
              curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
              curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
              $response = curl_exec($ch);
              curl_close($ch);
              print_r($response);

       //STEP 8 [Explode and Add to MySQL Database] ----------------------------------

       //Delete old data from MySQL database.
              $q = "TRUNCATE $duser";
              mysql_query($q);

       //Seperate the information received from Delicious and store in MySQL database.
              $responses = explode("<post href=\"", $response);
              foreach ($responses as $key => $item) {
                            if ($key == 0) {}
                            else {
                            $items = explode("\" hash=\"", $item);
                            $delicious_url = $items[0];
                            $items = explode("\" description=\"", $items[1]);
                            $delicious_hash = $items[0];
                            $items = explode("\" tag=\"", $items[1]);
                            $delicious_description = $items[0];
                            $items = explode("\" time=\"", $items[1]);
                            $delicious_tags = $items[0];
                            $items = explode("\" extended=\"", $items[1]);
                            $delicious_date = $items[0];
                            $items = explode("\" meta=\"", $items[1]);
                            $delicious_extended = $items[0];


                            $q = "INSERT INTO $duser (description, url, extended, tags,
                            date, hash) VALUES ('".$delicious_description."',
                            '".$delicious_url."', '".$delicious_extended."',
                            '".$delicious_tags."', '".$delicious_date."',
                            '".$delicious_hash."')";

                            mysql_query($q);
                            }
              }
}

/////////////////////////////////////////////////////////////////////////////////////
//If this delicious user has NOT already been logged in then login and request
//credentials.
/////////////////////////////////////////////////////////////////////////////////////

elseif(!$_GET["oauth_token"] && !$_GET["oath_verifier"]) {

       //Save the current nonce and Delicious username to be used later.
              $q = "INSERT INTO oauth (nonce, duser) VALUES ('".$nonce."',
'".$duser."')";
              mysql_query($q);

       //STEP 2 [Get Request Token] ------------------------------------------------

       //Collect the neccesary variables and construct the API URL
              $url = "https://api.login.yahoo.com/oauth/v2/get_request_token?";
              $url.= "oauth_nonce=".$nonce;
              $url.= "&oauth_timestamp=".$timestamp;
              $url.= "&oauth_consumer_key=".$consumer_key;
```

```
                $url.= "&oauth_signature_method=plaintext";
                $url.= "&oauth_signature=".$secret."%26";
                $url.= "&oauth_version=1.0";
                $url.= "&xoauth_lang_pref=en-us";
                $url.= "&oauth_callback=".$callback;

        //Make the API Call to Yahoo!
                $ch = curl_init();
                curl_setopt($ch, CURLOPT_URL, $url);
                curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                $response = curl_exec($ch);
                curl_close($ch);

        //Seperate the variables received from Yahoo! and store in variables.
                $responses = explode("&", $response);

                foreach($responses as $item) {
                        $items = explode("=", $item);
                        if ($items[0] == "oauth_token") {
                                $oauth_token = $items[1];
                        }
                        if ($items[0] == "oauth_token_secret") {
                                $oauth_token_secret = $items[1];
                        }
                        if ($items[0] == "xoauth_request_auth_url") {
                                $xoauth_request_auth_url = urldecode($items[1]);
                        }
                }
        //Save the variables in the MySQL database to be used later.
                $q = "UPDATE oauth SET request_token='".$oauth_token."',
request_token_secret='".$oauth_token_secret."' WHERE nonce='".$nonce."'";
                mysql_query($q);

        //STEP 3 [Send User to Yahoo! Login] ------------------------------------------

        //Construct the link to Yahoo! for logging in.
                echo "<p><a href=\"";
                echo $xoauth_request_auth_url;
                echo "&oauth_nonce=".$nonce;
                echo "&oauth_timestamp=".$timestamp;
                echo "&oauth_consumer_key=".$consumer_key;
                echo "&oauth_signature_method=plaintext&oauth_signature=".$secret."%26";
                echo "&oauth_version=1.0&xoauth_lang_pref=en-
us&oauth_callback=".$callback;
                echo "\">Log In</a></p>";
        }

        else {

        //STEP 4 [Request Access Token] -----------------------------------------------

        //Collect variables sent from Yahoo!
                $verifier = $_GET["oauth_verifier"];
                $request_token = $_GET["oauth_token"];
                $nonce_constant = $_GET["nonce"];

        //Save the variables in MySQL database for later use.
                $q = "UPDATE oauth SET verifier='$verifier' WHERE
nonce='$nonce_constant'";
                mysql_query($q);
```

```
       //Get the Request Token from the MySQL database. Then construct the API URL.
              $q = "SELECT * FROM oauth WHERE nonce='".$nonce_constant."' &&
request_token='".$request_token."'";
              $get_rts = mysql_query($q);

              while($row = mysql_fetch_array($get_rts)) {

              $url = "https://api.login.yahoo.com/oauth/v2/get_token?";
              $url.= "oauth_consumer_key=".$consumer_key;
              $url.= "&oauth_signature_method=PLAINTEXT";
              $url.= "&oauth_version=1.0";
              $url.= "&xoauth_lang_pref=en-us";
              $url.= "&oauth_verifier=".$verifier;
              $url.= "&oauth_token=".$request_token;
              $url.= "&oauth_timestamp=".$timestamp;
              $url.= "&oauth_nonce=".nonce;
              $url.= "&oauth_signature=".$secret."%26".$row["request_token_secret"];

       //Get the Delicious User from MySQL database.
              $duser = $row["duser"];

              }
       //Make the API Call to Yahoo!
              $ch = curl_init();
              curl_setopt($ch, CURLOPT_URL, $url);
              curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
              $response = curl_exec($ch);
              curl_close($ch);

       //Seperate the variables received from Yahoo! and store in variables.
              $responses = explode("&", $response);

              foreach($responses as $item) {
                     $items = explode("=", $item);
                     if ($items[0] == "oauth_token") {
                            $oauth_token = $items[1];
                     }
                     if ($items[0] == "oauth_token_secret") {
                            $oauth_token_secret = $items[1];
                     }
                     if ($items[0] == "oauth_expires_in") {
                            $oauth_expires_in = $items[1];
                     }
                     if ($items[0] == "oauth_session_handle") {
                            $oauth_session_handle = $items[1];
                     }
                     if ($items[0] == "oauth_authorization_expires_in") {
                            $oauth_authorization_expires_in = $items[1];
                     }
                     if ($items[0] == "xoauth_yahoo_guid") {
                            $xoauth_yahoo_guid = $items[1];
                     }
              }

              //STEP 5 [Build the Base String] ----------------------------------------

              //Collect the neccesary variables and construct the basestring (URL and
              //Parameters)
                     $apiurl = "http://api.del.icio.us/v2/posts/all";
                     $params = "oauth_consumer_key=".$consumer_key;
                     $params .= "&oauth_nonce=".$nonce;
```

```
                   $params .= "&oauth_signature_method=HMAC-SHA1";
                   $params .= "&oauth_timestamp=".$timestamp;
                   $params .= "&oauth_token=".$oauth_token;
                   $params .= "&oauth_version=1.0";

             //URL Encode the URL and Parameters
                   $urlencoded_apiurl = urlencode($apiurl);
                   $urlencoded_params = urlencode($params);

             //Combine the GET method, Encoded URL and Encoded Parameters.
                   $basestring = "GET&".$urlencoded_apiurl."&".$urlencoded_params;

             //STEP 6 [Generate the Signature] ---------------------------------------

             //Create the HMAC Signature using functions from the PHP library.
                   $signature = base64_encode(hash_hmac("sha1", $basestring,
$secret."&".$oauth_token_secret, true));

             //STEP 7 [Create Request to Delicious] ----------------------------------

             //Construct the Delicious API URL.
                   $url = "http://api.del.icio.us/v2/posts/all";

             //Create the header array with appropriate variables and HMAC signature.
                   $headers = array(
                          "Authorization: OAuth
                          realm=\"yahooapis.com\",
                          oauth_consumer_key=\"".$consumer_key."\",
                          oauth_nonce=\"".$nonce."\",
                          oauth_signature_method=\"HMAC-SHA1\",
                          oauth_timestamp=\"".$timestamp."\",
                          oauth_token=\"".$oauth_token."\",
                          oauth_version=\"1.0\",
                          oauth_signature=\"".$signature."\""
                   );

             //Make API Call to Delicious.
                   $ch = curl_init();
                   curl_setopt($ch, CURLOPT_URL, $url);
                   curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
                   curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                   $response = curl_exec($ch);
                   curl_close($ch);
                   print_r($response);

             //STEP 8 [Explode and Add to MySQL Database] ----------------------------
             //Delete old data from MySQL database.
                   $q = "TRUNCATE $duser";
                   mysql_query($q);

             //Seperate the information received from Delicious and store in MySQL
             //database.
                   $responses = explode("<post href=\"", $response);
                   foreach ($responses as $key => $item) {
                                 if ($key == 0) {}
                                 else {
                                 $items = explode("\" hash=\"", $item);
                                 $delicious_url = $items[0];
                                 $items = explode("\" description=\"", $items[1]);
                                 $delicious_hash = $items[0];
                                 $items = explode("\" tag=\"", $items[1]);
```

```
                                  $delicious_description = $items[0];
                                  $items = explode("\" time=\"", $items[1]);
                                  $delicious_tags = $items[0];
                                  $items = explode("\" extended=\"", $items[1]);
                                  $delicious_date = $items[0];
                                  $items = explode("\" meta=\"", $items[1]);
                                  $delicious_extended = $items[0];


$q = "INSERT INTO $duser (description, url, extended, tags, date, hash) VALUES
('".$delicious_description."', '".$delicious_url."', '".$delicious_extended."',
'".$delicious_tags."', '".$delicious_date."', '".$delicious_hash."')";

$sql = mysql_query($q);
if ($sql) {echo "Successful";}
                                  }
                    }
}

?>
```

Step Eight: Get Initial Yahoo! Credentials

Because of the increased security with Yahoo! logins, each initial login will have to be manual . After that, LSUlicious will automatically get the credentials.
To manually log in each account, open each of the PHP files named for a delicious username in your browser. A "log in" link will appear. Click the link and enter in the username and password for that delicious account. You will be rerouted back to the lsulicious.php page. If you see the word "Successful", then the log in is complete.

Step Seven: Setup CRON

In each PHP file named for a delicious username (created in step four), paste the code below in the document. Replace "FILEPATH" with the file path to the lsulicious.php file and "DELICIOUS_USERNAME" with the delicious username associated with that file.

```
<?php header( 'Location: FILEPATH/lsulicious.php?duser=DELICIOUS_USERNAME' ) ; ?>
```

Once you have all of these files set up, contact your systems administrator to set up a CRON to run these PHP files on a regular schedule (generally nightly or weekly.) Do not run the CRON too frequently, as this can lead to being banned from the Delicious servers.

Step Eight: Create Final Product

Create a PHP page based on the templates you use for all other pages or with a new custom design. Then place the code below in the location you would like the search box, tag suggestion menu and feed to appear. Replace all of the capitalized words with the appropriate information.

To remove any of the three tools,   place two slashes "//" before the function call. For example, to suppress the tag suggestion box replace "suggest();" with "//suggest();".

```
<?
//Connect to the Subject Guide Toolbox to use the connect function.
include("FILEPATH/sgt.php");

//Replace DELICIOUSUSERNAME with the delicious username.
$duser = "DELICIOUSUSERNAME";

//Replace URLOFFEED with the URL of this page.
$feedloc = "URLOFFEED";

//Replace TAGSDISPLAY with either "text", "tooltip" or "none" to choose how tags will
display.
$tagsdisplay = "TAGSDISPLAY";

//Replace DESCRIPTIONDISPLAY with "text", "tooltip", "none" or "moreinfo" to choose
how the descriptions will display.
$descdisplay = "DESCRIPTIONDISPLAY";

//Replace BASETAG with a single tag name to filter all functions on page by that tag
without notifying the user.  This is good for using one delicious accounts to run
multiple pages based on tag structures.
$basetag = "BASETAG";

connect();
searchbox();
suggest();
feed();

?>
```

Step Nine: Style the Feed and Tag Suggestion Box

Paste the code below into your CSS file or in the header of your document surrounded by <style> tags.

```
/*The classes deliciousitem, odd and even controls the style for  each bookmark
displayed in the feed of resources*/
.deliciousitem {
margin-right:5px;
margin-left:5px;
padding:5px 5px 5px 5px;
border-top:1px solid #ccc;
}
.odd {background:#fafafa;}
.odd:hover {background:# fafafa;} /* change this color to add a hover effect */
.even {background:#fff;}
.even:hover {background:# fff;} /* change this color to add a hover effect */

.delicioustitle {  /* This controls the style of the bookmarks’ title. */
font-size:14px;
margin:0px;
line-height:150%;
padding:0px;
}
```

```
.deliciousextended { /* This controls the style of the bookmarks’ description. */
font-size:12px;
line-height:150%;
}

/* The classes deliciousextendedmore and moreinfo create the "moreinfo" style for
descriptions. */
.deliciousextendedmore {
font-size:12px;
line-height:150%;
height:6px;
width:90px;
overflow:hidden;
}
.deliciousextendedmore:hover {
height:auto;
width:auto;
}
.moreinfo {
font-size:9px;
color:#43267A;
}

.tagcount {font-size:10px;}

/* The classes suggest and suggestbox style the suggestion feature */
.suggest {text-align:center;margin:0px 0px 15px 0px;padding:0px;}
.suggestbox {background:#f5f5f5;padding:0px 2px 0px 2px;border-top:#43267A 1px
solid;border-bottom:#43267A 1px solid;}

/* The class tag creates the style for the list of tags in the suggestion box. */
.tag {
display:block;
float:left;
background:#fff;
height:20px;
margin:2px 2px 1px 2px;
padding:1px 16px 1px 5px;
width:215px;
border:1px solid #c1c1c1;
}

.delicioustags {font-size:9px;margin-top:10px;}
.deliciousbuttons {width:225px;padding-left:5px;}
.deliciousicon {padding-left:5px;padding-right:5px;}
.deletetag {font-weight:normal;font-size:10px;}
```