Mashups: Remixing the Web

H79.2802

ASSIGNMENT #2: SERVER-SIDE DATA PROCESSING
Due October 6, 2010 (in lecture)

	Reflection		Ideation		Exercise		Bonus Challenge

Time to Process Form Input with PHP (*12 Points*)

Time Magazine offers an archive of its entire collection of magazine covers going back to 1923. You can search and browse the covers at this URL:

http://www.time.com/time/coversearch

After a bit of browsing through the covers, you may notice that the URL for each cover image has a consistent format, based on the date of the issue. For example, the cover for September 30, 1929 can be found at the following URL:

http://img.timeinc.net/time/magazine/archive/covers/1929/1101290930_400.jpg

There is one issue per week, and all of the issue dates are Mondays. Based on this information, when given an arbitrary date, you should be able to construct a URL that retrieves the magazine cover for the weekly issue that was current on that date.

1. Take a look at the following HTML page:
   http://webremix.org/assignments/dateform.html
   It is a simple form allowing the user to specify a date in *mm/dd/yyyy* format. Enter a date and click the button. The form calls `getcover.php`, which displays the Time Magazine cover for the date specified.
2. Your goal is to write your own script mimicking the behavior of `getcover.php`. Given a date string in *mm/dd/yyyy* format, it should:
   a. Verify that the date is valid; if invalid, display an error message.
   b. Make sure that the date is within the range of dates included in the archive.
   c. Find the date of the current issue, which will always be the Monday preceding the date specified.
   d. Construct a URL to retrieve the cover image for that issue.
   e. Output HTML to display the cover image.

You will most likely want to take advantage of PHP date parsing and string formatting functions such as `strtotime`, `date`, and `printf`; refer to the official PHP documentation to learn more about these functions.

3. To verify that your script is working correctly, download `dateform.html`, and modify it to call your version of `getcover.php`.

4. Include a link to your working script in your assignment hand-in. Also include a printout of your source code.

This Just In: Calling Web APIs with PHP (*16 Points*)

The New York Times Article Search API lets you search New York Times articles from 1981 to today, retrieving headlines, abstracts, lead paragraphs, and links to associated multimedia. You can read the detailed documentation for the API here:
http://developer.nytimes.com/docs/article_search_api

In this exercise, you will use the New York Times API to retrieve historical front-page headlines for a specified date.

1. In order to use the API, you will need to register an API key. Begin this exercise by requesting a key using the following instructions:
http://developer.nytimes.com/docs/reference/keys

2. Once you have registered a key, peruse the API documentation to learn how to make search queries. Formulate a query to retrieve the title, word count, and byline of all of the articles written in the year 2005 that contain the word "university" in the title. Enter the query into your browser, and take a look at the formatted response that comes back. Include the query URL (with the API key omitted) in your assignment hand-in.

3. As with the previous exercise, you should begin with a simple form:
http://webremix.org/assignments/dateform.html

Your goal: given a date, retrieve the front-page headlines for that date. You can specify that you want only front page articles using the `page_facet` request parameter. Display the headlines ordered by word count, from longest article to shortest article. Request small images as well, using the `small_image` and `small_image_url` request parameters; if the article includes a small image, display the image alongside the headline.

The Article Search API returns results in the JSON structured data format. You can parse the results using the PHP function `json_decode`, which converts a JSON string into a PHP object (or to an associated array, if you prefer). Once you have formulated a request

string, you can call the PHP function `file_get_contents` to retrieve a response to the request, and parse the response with `json_decode`.

You may wish to refer to the PHP documentation to learn about different methods for sorting arrays, such as `usort` and `array_multisort`. You can view a working example for reference here:
http://webremix.org/assignments/getheadlines.php

If your script is working correctly, it should return the same articles as this example. Include your source code and a link to your working script in your assignment hand-in.

Climbing the Charts: More PHP Data Processing (*16 Points*)

Billboard (http://www.billboard.com) is a music publication. Since 1945, Billboard has published charts tracking the best-selling and most-played songs and albums. Billboard offers an API that gives access to the full history of these charts. In this exercise, you will retrieve a list of Top Ten Singles for a given date.

1. As in the previous exercise, you must begin by requesting an API key. Sign up for a key here:
   http://developer.billboard.com/member/register

   Once you have received your API key, study the documentation for the Billboard API to learn how to query for charts:
   http://developer.billboard.com/docs

2. For this exercise, we will be using the "Billboard Hot 100" chart, which has a chart ID of 379. Formulate a query to return the first 50 items of the Hot 100 Chart for the week of August 16, 2008, in XML format. Include the query URL (with the API key omitted) in your assignment hand-in.

3. Now that you have learned how to retrieve a portion of the Hot 100 chart, write a script that retrieves the Top Ten singles on the Hot 100 Chart for the date specified. As in the previous exercises, begin with the simple date form:
   http://webremix.org/assignments/dateform.html

   Construct a PHP script that reads the date specified by this form, and displays the Top Ten list for the date specified.

   Although this task sounds like it should be easy, the limitations of the Billboard API make it a bit of a challenge:
   • Requests to the Billboard API specify a date range, and the result will include all of the charts in that range. Charts are released weekly, so to get the current chart,

request the charts from a one-week span including the date specified; this will give you the chart that was current on that date.

- Note that the results are returned in order of release date, not in order of their ranking in the chart. After retrieving the chart, you must re-sort the entries by chart position.
- Finally, you may notice that the Billboard API limits the number of results returned to 50. Since the results are sorted in order of date, not chart position, the first 50 results may not include the 10 top songs. To get around this limitation, make two queries: one query for chart items 1-50 chart items, and another query for chart items 51-100. Merge the two result sets, sort the songs by chart position, and extract and display the top ten songs.
- If you choose to request the data in JSON format, you can read it with `json_decode`, as before. If you request the data in XML format, you can use `simplexml_load_string` to parse the XML data into a PHP object.

4. You can verify that your solution is working correctly by comparing it against the results from this reference implementation:
   http://webremix.org/assignments/getchart.php

   Include your source code and a link to your working script in your assignment hand-in.

Put It Together in Style (*9 Points*)

Create a single PHP script that takes a date as input, and displays a page with all of the elements you pulled from external websites in the previous exercise: a Time Magazine cover, a set of New York Times headlines, and the Billboard Top Ten list.

Use HTML tags to structure the content, and add CSS styles to position the elements and give them a compelling visual style. Include the URL to this page in your assignment hand-in.

Jazz It Up with Album Covers (*10 Points*)

The last.fm API gives you access to a wealth of information about musical artists, albums, and songs, as well as music preferences of users of the online radio service. You can read about the full range of available data in the API documentation:

http://www.last.fm/api

One particularly handy thing you can do with the last.fm API is retrieve images of artists and album covers. For example, the following query returns an XML-formatted document

containing various information about Cher's *Believe* album, including links to cover images of different sizes:

[http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=b25b959554ed76058ac2 20b7b2e0a026&artist=Cher&album=Believe](http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=b25b959554ed76058ac220b7b2e0a026&artist=Cher&album=Believe)

In the last exercise, we retrieved a Billboard Top Ten list, but it looks a little drab – let's liven it up with some pictures. Use the last.fm API to retrieve album cover images for the tracks in the list. Rather than trying to look up the album containing each track, you can simply specify the song title as the album name, since most of the Billboard Top Ten titles are released as singles. If you are unable to locate an album cover for a particular song, try querying last.fm for images of the artist instead. Modify your script to display the images you retrieve alongside the Billboard track listing.

Adding an Additional Data Source *(15 Points)*

Find another website that offers historical data. Using the data from this website, add an additional element to the page you created in the previous exercises. You are welcome to choose any site you like, but here are a few possible ideas for things to incorporate:

- Wikipedia has lists of notable events that occurred every day. For example, the events that occurred in August 2003 are on the page:
  http://en.wikipedia.org/wiki/August_2003
  Find and display an event that occurred on the date requested.
- Various finance websites, such as Xignite and StrikeIron, offer historical data about financial markets. Display information about the performance of a set of stocks or an index on the requested date.
- Find a website with historical weather data, such as the NSSL Historical Weather Data Archives, and look up and display the weather in a particular city on the date requested.
- Find an image archive that allows you to query for photos taken on a particular date. Display historical photos from the date requested.