Webservices, Proxies, Rest, File Uploads, Security

CS174

Chris Pollett

Nov. 3, 2008.

Outline

- Web Services
- REST
- JSON Example
- More PHP

Web Services

- One important use of AJAX and PHP is to allow you to write web services.
- A *web service* is a programming interface which can be invoked over HTTP.
- The first attempts to standardize such services made use of things like WSDL, SOAP, UDDI, etc. XML languages which tended to violate the KISS (keep it simple stupid) principle.
- Some simpler web service interfaces have been written by major companies using XML-RPC, JSON-RPC and REST.
- A XML-RPC document is a XML document which specifies a *remote procedure call*, i.e., which function of which object to invoke on some server; or it specifies the response of such a call. It later evolved into the more complicated and still evolving SOAP (Simple Object Access protocal)
- JSON-RPC is like XML-RPC but uses JavaScript Object Notation -- basically, a snippet of javascript code for an object.

REST

- REST stands for *Representational State Transfer*. It is a technique for writing web services developed by Roy Fielding in 2000.
- The idea is that an application state/method is viewed a resource. Each resource has a URL. There is a well defined way to tack on to this URL a query to invoke the function and return results. For example, the Yahoo News Rest Service might be invoked with a line like:

http://search.yahooapis.com/NewsSearchService/V1/newsSearch?appid=YahooDemo&query=madonna&results=2&language=en

JSON

- Stands for Javascript Object Notation.
- It is commonly used for sending data when REST is used by Javascript can immediately use the data.
- Primitive types in JSON are written as you expect:

  12.3 -- an example Number

  “hi there” -- an example String

  true -- an example Boolean, other possibility false.
- Arrays are written in square brackets and comma separated:

  [1, 4, 9] .
- Objects are written in braces and the name value pair are separated by a colon:

  {“bob”: 29, “sally” : 35}
- JSON data can be assigned to an object with the syntax:

  myObj = eval(“(” + data + “)”);

Proxies

- Javascript function is only allowed to make requests back to the server from which it came.
- So if you have a page http://somewhere.com/index.html

  and you would like the Javascript on it to make use of the Yahoo! Rest API, how do you do it?
- You need to use a proxy on your server which passes the request onto Yahoo!
- One example of a PHP script to do such proxy-ing can be found at:

http://developer.yahoo.com/javascript/samples/proxy/php_proxy_simple.txt

- To use such a proxy, you need to have PHP running on your machine.
- You could rename the above file proxy.php set its permissions so that is executable and put it somewhere you know under your document root.
- Then to access the Yahoo! service via the proxy you could do:

http://yourServer/proxy.php?yws_path=urlencodepath

- For example,

http://www.cs.sjsu.edu/faculty/pollett/test/proxy.php?yws_path=NewsSearchService%2FV1%2FnewsSearch%3Fappid%3DYahooDemo%26query%3Dmadonna%26results%3D2%26language%3Den

Example

- Looked at proxy code from Yahoo!
- It is anexample of using the PHP curl_init, curl_exec, curl_setopt, curl_close.

File Uploads

- We are now going to look at a couple of useful things PHP can do with regard to form processing.
- We have already seen that usually information sent from forms is provided to our PHP scripts in the global variables: $_REQUEST, $_POST, $_GET
- These variables though are not used to handle file uploads. Instead, the variable $_FILES is used.

Example

- Consider the form:

```
<form enctype="multipart/form-data" method="post" action="test_upload1.php" >
  <input type="hidden" name="MAX_FILE_SIZE" value="1000000" /><!-- The size is also
    controlled by php.ini -->
  <input type="file" name="docname" />
  <input type="submit" value="Upload" />
</form>
```

- When test_upload1.php is run, the global variable $_FILES[“docname”] will be set to something like:

```
Array(
    [name] => mystyles.css
    [type] => text/css
    [tmp_name] => /private/var/folders/k-/k-GHnyslGhyOhMqq80ZXgk+++TI/-Tmp-/phpcSLlhk
    [error] => 0    [size] => 157)
```

- Hence, we can then do a command like:

  move_uploaded_file($_FILES["docname"]["tmp_name"], "$where_we_want");

  to get the file where we would like.

Security

- We are now going to spend the rest of the lecture looking at various things vaguely connected with security.
- Often data from clients comes to your server via some form. For example, the file file upload processing we just did.
- One annoyance you will have to deal with is robots that find your web forms and upload garbage to your site, spamming you.
- One solution to this problem is to use Captcha's.

Captcha

- CAPTCHA stands for **C**ompletely **A**utomated **P**ublic **T**uring **T**est to tell **C**omputers and **H**umans **A**part.
- These were developed at Carnegie Mellon around 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford.
- The basic idea is that you put on your form an image of a distorted string.
- You hope the robot cannot decipher the string from the image so won't be able to fill out that portion of the form correctly:

  Please type the following text:

Making a Simple Captcha in PHP

- The previous CAPTCHA might be created with the code:

```
<p><b>Please type in the following text:</b>
<?php
    $md5 = md5(microtime() * mktime());
    $captcha_string = substr($md5,0,5);
     $captcha_img = imagecreatetruecolor(70, 40);
      $color = imagecolorallocate($captcha_img, 255, 0, 255);
      $line = imagecolorallocate($captcha_img,233,239,239);
      imagestring($captcha_img, 5, 10, 10, $captcha_string, $color);
      imageline($captcha_img,0,0,39,29,$line);
      imageline($captcha_img,40,0,64,29,$line);
      imageline($captcha_img,0,40,64,0,$line);  imagejpeg($captcha_img,
      "images/captcha.jpg",100); imagedestroy($captcha_img);
$_SESSION['key'] = md5($captcha_string);  ?>
<img src="images/captcha.jpg" alt="captcha" style="position:relative; top:15px;"/>
<input type="text" name="key" size="5" /></p>
```

Thumbnails

- The code for creating a captcha uses the image libraries which can also be useful for creating such things as thumbnails:

```
function createThumb($image, $base, $filename)    {
   $image = imagecreatefromjpeg($base.$filename);
   //or could have imagecreatefromgif, etc
   $size = getimagesize($base.$filename);
   $thumb = imagecreatetruecolor(50, 50);
   imagecopyresampled($thumb, $image, 0,0, 0,0, 50, 50,
   $size[0],$size[1]);
   imagejpeg( $thumb, $base."thumb.jpg", 100 );
   imagedestroy($image);
   imagedestroy($thumb);
   }
```

Sending a Mail Message

- It is often useful to collect a person's e-mail address with a form.
- By mailing, a person a special code that allows them to complete a registration process, one can verify that one has a real e-mail address of a real person.
- The simplest way to do this is to use the mail() command:

```
$message = "Here is a mail message";
mail("Someone@somewhere.com",
        "Here is the title",
      $message,
      "From: cpollett@somewhereelse.com");
```

- This could be combined with a captcha to try to reduce the risk of your site spamming other sites.