PHP, JSON, REST

CS174

Chris Pollett

Oct 17, 2007.

Outline

- Web Services
- REST
- JSON Example
- More PHP

Web Services

- One important use of PHP is to allow you to write web services.
- For HW3, you are asked to write a web service, let's spend a moment to talk about them.
- 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 + “)”);

General Syntax

- Let's return to discussing PHP.
- Recall a section of PHP code is delimited with <?php … code here …?>
- The command include("filename.php"); can be used to include one document in another.
- All variables in PHP begin with a $ sign. Names of variables are otherwise like in Perl or other common programming languages.
- PHP can use either Perl, C++, or C comments: #, //, /* … */

Primitives

- PHP has four scalar types: Boolean, integer, double, and string.
- There are also two compound types: array and object and two special types: resource and NULL.
- As with Javascript and Perl, PHP is dynamically typed: it has no type definitions.
- An unassigned variable is sometimes called an unbound variable and has value NULL. This can be coerced to one of the other types depending on the context.
- To check if a variable is currently bound you can use the IsSet($variable_name) function. This returns TRUE or FALSE (one of the two Boolean values).
- You can also call error_reporting(15), to set PHP's error reporting level so that it prints unbound variables.

Integer, String, and Boolean Type

- Integers in PHP correspond to C long's (signed), so there size depends on the size of a long on a given machine.
- This is usually 32 bits.
- PHP's double type corresponds to C's double and literals follow the format for C (or Perl or Javascript) floating point literals: I.e., .345, 3E-7, etc.
- Characters in PHP are single bytes (No Unicode!).
- String literal are built up out of these characters.
- PHP distinguish between single quote and double quote literals in the same way that Perl does. So the variable in "Bob=$bob" is interpolated but in 'Bob=$bob' is not.
- There are only two values for Boolean's: TRUE or FALSE. As for coercion, the empty string or a "0" or a "0.0" is interpreted as FALSE. Otherwise strings evaluate to TRUE. Integers and double evaluate to TRUE as long as they aren't zero.

Arithmetic Operators,etc

- PHP supports the usual arithmetic operators: +, -, *, /, %, ++, --.
- PHP also does type coercion.
- Division should be assumed to output a double if any fractional value exists.
- Some useful predefined functions are: floor, ceil, round, srand, rand, abs, min, and max.
- String concatenation is done with a “.”
- Some useful string operations are strlen, strcmp, strpos, substr, chop, trim, ltrim, strtolower, strtoupper.
- Type conversion can be done using expressions like (int)$sum or intval($sum) or settype($sum, integer).
- You can also check the type with is_int, is_integer, is_long, etc.
- Assignment operators are like in C, Java, etc.

Output

- You can output text to be inserted into the page using either of the commands: print or echo
- For example,

  print “hi there”;

Control Statements

- PHP supports:
  - if, else if, else: For example,

    ```
    if($a) print “hello”;
    ```
  - switch case:

    ```
    switch($a)
    {
        case 5:
         echo “hello”;
    }
    ```
  - for : for($a = 0; $a<10; $a++){echo “hello $a”;}
  - while: while(!$var) { /*do something*/}
  - do while: do {/*do something */} while (!$var);

Arrays

- Arrays can be declared with the syntax:

  $a = array(“hi”, 1, 2);
- You can nest arrays:

  $b = array(“hi”, array(1,2), 2);
- You can dereference arrays as in most languages:

  echo $a[0];
- You can cycle over elements of an array using foreach:
  foreach($arr as $var){echo $var;}