Old Dominion University

CS 418/518
Web Programming
Fall 2012

Validating User Input, Handling and Avoiding Errors

Dr. Michele Weigle

http://www.cs.odu.edu/~mweigle/CS418-F12/

Outline

- Assigned Reading
  - Chapter 8 “Validating User Input"
  - Chapter 9 "Handling and Avoiding Errors"
- Server-Side Input Validation
  - simple strings
  - integers
  - formatted text
- Handling Errors
  - using Apache
  - handling PHP errors
  - exceptions

Validating User Input

- Why do we need input validation?
  - people make typo mistakes
  - people don't want to give out info unless you really need it (phone numbers, email address, SSNs)
  - people can be attackers
- We'll look at validating
  - simple string values
  - integer values
  - formatted text input

Why Server-Side Input Validation?

- Client-side input validation is nice but not a replacement for server-side validation
  - client-side security == no security
    - malicious or broken clients
  - client-side (Javascript) examples: http://www.codeave.com/javascript/category.asp?u_category=Forms
- Server-side input validation can be built using:
  - empty(), is_numeric(), is_string(), is_bool(), is_array(), is_object(), etc.
  - http://us2.php.net/manual/en/ref.var.php

Input Doesn't Validate, Now What?

- We'll look at three different ways to handle the validation errors:
  - generate the error on the same page (don't load a different URI)
  - load a separate URI that is just for handling errors
  - reload the same page with an error argument

Same URI to Handle Errors

```
…
if (empty($var1)) {
        $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
        $errors .= "$var2 should be a number";
}
// check all anticipated error conditions
…

if (empty($errors)) {
        // do interesting work
} else {
        internal_error_function($errors);
}

function internal_error_function ($errors) {
        // generate pretty HTML response
        // provide link to start over
}
```

Separate URI to Handle Errors

```
…
if (empty($var1)) {
        $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
        $errors .= "$var2 should be a number";
}
// check all anticipated error conditions
…

if (empty($errors)) {
        // do interesting work
} else {
        $errors = urlencode($errors);
        header("Location: http://foo.edu/error.php?error=$errors");
}
```


Encoding/Decoding URLs

- RFC-1738 requires "unsafe" and "reserved" characters to be encoded in URIs:
  - http://www.rfc-editor.org/rfc/rfc1738.txt
  - Reserved examples "/:", ":", "?"…
  - Unsafe examples [space], "%", "#"…
- PHP urlencode(), urldecode()
  - http://us2.php.net/manual/en/function.urlencode.php
  - http://us2.php.net/manual/en/function.urldecode.php
- More info
  http://www.blooberry.com/indexdot/html/topics/urlencoding.htm

Same URI with Error Argument

```
…
if (empty($var1)) {
        $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
        $errors .= "$var2 should be a number";
}
// check all anticipated error conditions
…

if (empty($errors)) {
        // do interesting work
} else {
        $errors = urlencode($errors);
        header("Location:".$_SERVER["REQUEST_URI"]."?errors=$errors";
}
```


Demo/Walkthrough Time

- Examples from Chapter 8
  https://mweigle418.cs.odu.edu/~mweigle/textbook/ch08.htm
- Checking for Empty Entries
  - index.php
  - movie-rev01.php
  - commit-rev01.php

Outline

- Server-Side Input Validation
  - simple strings
  - integers
  - formatted text
- Handling Errors
  - using Apache
  - handling PHP errors
  - exceptions

Formatted Text

- To do formatted text (e.g., date) validation, we'll use *regular expressions*
- DD-MM-YYYY format for date
  - ([0-9]{2})-([0-9]{2})-([0-9]{4})
  - 2 digits between 0-9
  - "-" character
  - 2 digits between 0-9
  - "-" character
  - 4 digits between 0-9

Which will match?
1-2-2012
1-02-2012
01-02-12
01-02-2012

Regular Expressions

Note: `ereg()` has been deprecated as of PHP 5.3 and will cause a warning!

```
if ( !preg_match("/([0-9]{2})-([0-9]{2})-([0-9]{4})/",
            $_POST['movie_release'], $reldatepart) ) {
        $error .= "Please+enter+a+date+with+the+dd-mm-yyyy+format";
}
```

if `$_POST['movie_release']` is `31-05-1969` then:

```
$reldatepart[0] = 31-05-1969
$reldatepart[1] = 31
$reldatepart[2] = 05
$reldatepart[3] = 1969
```

http://us2.php.net/regex
http://oreilly.com/catalog/9780596528126/

Date/Time

- Change the date string into a timestamp (to store in the database) using `mktime()`
- Takes as parameters: hour, min, seconds, month, day, year
- If valid date, returns the number of seconds since Jan 1, 1970
- If not a valid date (e.g., 99-99-9999), returns -1

```
$movie_release = mktime (0, 0, 0, $reldatepart['2'],
$reldatepart['1'], $reldatepart['3']);
```

http://us2.php.net/manual/en/ref.datetime.php

Same Thing with MySQL

- UNIX_TIMESTAMP()
  - takes YYYY-MM-DD HH:MM:SS format string
  - creates a timestamp

```
$reldate = $reldatepart['3'] . "-" .
           $reldatepart['2'] . "-" .
           $reldatepart['1'] . " 00:00:00";

$sql = "INSERT INTO movie_main (release_date) " .
       "VALUES (UNIX_TIMESTAMP('$reldate'))";
```


Escaping HTML

```
<?php
$orig = "I'll \"walk\" the <b>dog</b> now";

$a = htmlentities($orig);

$b = html_entity_decode($a);

echo $a; // I'll &quot;walk&quot; the &lt;b&gt;dog&lt;/b&gt; now

echo $b; // I'll "walk" the <b>dog</b> now

?>
```

See:
http://www.w3schools.com/PHP/func_string_htmlentities.asp
http://php.net/manual/en/function.htmlentities.php
http://us2.php.net/manual/en/function.html-entity-decode.php
http://www.w3schools.com/PHP/func_string_html_entity_decode.asp
Also: http://en.wikipedia.org/wiki/Lightweight_markup_language

Demo/Walkthrough Time

- Examples from Chapter 8
  https://mweigle418.cs.odu.edu/~mweigle/textbook/ch08.htm
- Checking for Format Errors
  - movie-rev02.php
  - commit-rev02.php

Outline

- Server-Side Input Validation
  - simple strings
  - integers
  - formatted text
- Handling Errors
  - using Apache
  - handling PHP errors
  - exceptions

Apache httpd.conf

```
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
ErrorDocument 404 /error.php?404
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
…
```

If you don't have access to httpd.conf (we don't), then you can put the `ErrorDocument` directives in `~/cs418_html/.htaccess`

Demo/Walkthrough Time

- Examples from Chapter 9

  https://mweigle418.cs.odu.edu/~mweigle/textbook/ch09.htm

- Apache's ErrorDocument Directive
  - .htaccess
  - error-rev01.php

Send Email on Error

- PHP syntax
  - mail($to,$subject,$body,$headers)
- body
  - can be HTML formatted
- headers
  - MIME-Version: 1.0\r\n
  - Content-type: text/html; charset=iso-8859-1\r\n
  - From: Apache Error <host@yourdomain.com>\r\n


```
Received: from mweigle418.cs.odu.edu (128.82.4.129) by mail1.cs.odu.edu
(128.82.4.99) with Microsoft SMTP Server id 14.1.218.12; Fri, 14 Sep 2012 20:10:24
-0400
Received: by mweigle418.cs.odu.edu (Postfix, from userid 33)id AA48E9FBA2; Fri, 14
Sep 2012 20:09:08 -0400 (EDT)
To: Administrator <mweigle@cs.odu.edu>
Subject: Apache Error Generation
X-PHP-Originating-Script: 280:error-rev02.php
Content-Type: text/html; charset="iso-8859-1"
From: Apache Error <host@yourdomain.com>
Message-ID: 20120915000908.AA48E9FBA2@mweigle418.cs.odu.edu
Date: Fri, 14 Sep 2012 20:09:08 -0400
Return-Path: www-data@cs.odu.edu
...
Received-SPF: None (mail1.cs.odu.edu: host@yourdomain.com does not designate
permitted sender hosts)
MIME-Version: 1.0

<html><head><meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1"><title>Apache Error</title></head><body>Error occurred on
<b>Friday, September 14, 2012</b> at <b>20:9:8:2012</b><br>Error received was a
<b>404</b> error.<br>The page that generated the error was: <b>/~mweigle/noexist</
b><br>The generated error message was:<h1>&quot;Page Not Found&quot; Error Page -
(Error Code 404)</h1>The page you are looking for cannot be found<br><a
href="mailto:sysadmin@localhost.com">Contact</a> the system administrator if you
feel this to be in error</body></html>
```

Outline

- Server-Side Input Validation
  - simple strings
  - integers
  - formatted text
- Handling Errors
  - using Apache
  - handling PHP errors
  - exceptions

Up Next:
Project 1 demos,
Project 2 assignment

Demo/Walkthrough Time

- Examples from Chapter 9

  https://mweigle418.cs.odu.edu/~mweigle/textbook/ch09.htm

- Creating Error E-mail
  - error-rev02.php
- Generating PHP Errors
  - snippet01.php, snippet02.php, snippet03.php
- Creating a Custom Error Handler
  - custom_error-rev01.php, custom_error-rev02.php
- Creating a Full-Featured Error Page
  - feature_error.php
- Exceptions
  - exceptions-rev01.php, exceptions-rev02.php, exceptions-rev03.php