

# Web Focused Programming With PHP

May 20 2014

**Thomas Beebe**

*Advanced DataTools Corp*

([tom@advanceddatatools.com](mailto:tom@advanceddatatools.com))

***Advanced DataTools***

## Tom Beebe



Tom is a Senior Database Consultant and has been with Advanced DataTools for over 10 years. He has been working with Informix since college and is currently the lead consultant for Networking, Unix System Administration and Web Development. Tom is Project Manager and lead developer on a variety of Web Development projects.

Contact Info:

[tom@advanceddatatools.com](mailto:tom@advanceddatatools.com)

[www.advanceddatatools.com](http://www.advanceddatatools.com)

703-256-0267 x 106

# About This Webcast Series

- Four Webcasts:
- March 18<sup>th</sup> – Getting started
- April 22<sup>nd</sup> – Data structures, objects, functions and database access.
- May 20<sup>th</sup> – Introduction to web focused programming. Creating simple and advanced forms
- June 17<sup>th</sup> – Putting it all together, assorted advanced topics.
- Make sure to register for them

# Schedule

- How does the tech actually work
- Writing simple web code
- What are forms, how do they work
- `$_GET`, `$_POST`, `$_REQUEST`, `$_FILE`
- Building PHP forms
- Connecting it to Informix
- `$_SESSION`
- Next Steps

# How Does A Web Call Work

- Browser makes a request 'get /index.html'
- Web server looks for the file in the web directory
- Returns a status code
  - 2xx Success
  - 3xx Redirect
  - 4xx Error
  - 5xx Server Error
- If it is a 200 it will read and send the index.html file and send it along
- Browser receives the HTML code and starts rendering it top down
- Follows and loads any links (js, css, etc) in order they are seen
- Displays the page

# Path Of A Web Script

- Browser sends a request
  - Get
- Returns status code
- 200 it will process
- Reads in the PHP code into memory.
- Compiles and executes the PHP code and merges it into the code.
- HTML code will be handled like .html files
- The executed code will be brought into apache memory
- Streamed the processed code out to the browser
- Browser displays the code

## Basic Web Code

```
<html><head></head><body>  
<p>The value of X is: <?php echo $x; ?></p>  
</body></html>
```

# Templating Web Code

```
$line = "<html><head></head><body>  
  The value of X is: $x  
  </body></html>";  
print "$line";
```



# Browser Vs Server

- Server code (PHP) is processed at time of execution. Once the page has been sent it is finished running.
- Once it is on the browser you need a browser based language to make anything happen.
  - (Javascript, jquery, etc)

# Forms

- Forms are embedded in HTML
- Input Types:
  - Text
  - Textarea
  - Radio
  - Checkbox
  - Select
  - Submit
  - Button
  - Hidden
  - File

# Forms

- Starts with `<form>` and ends with `</form>`
- Parameters
  - method (post/get)
  - Action (target of submission)
- Submit call sends it
- Each form field has to be uniquely named
- Two exceptions
  - Radio Buttons
  - Array of values (often checkboxes) use `[]` on the name

# What Actually Happens

- Browser requests webpage with form
- Sends it back over, displays in browser
- Filled out by user, submit is pressed
- Data is sent to the server along with a request for the target page (action attribute)
- Server runs code, sends the new page
- Code does whatever you tell it to with the submitted data

# Get vs Post

- Get passes data via URL line
  - Must use urlencode to pass it cleanly
  - Limit to length (around 2000 characters)
  - Cannot pass binary data
  - Will cleanly allow back and aid in browsing
- Post passes in a binary way
  - Can pass files
  - Don't need to urlencode
  - Back will not work

# Handling Data

- `$_POST` – handles post data
- `$_GET` – handles get data
- `$_REQUEST` – gets both
- `$_FILE` manages passed files
- All are associated arrays
  - `$user_id = $_REQUEST['user_id'];`

# Handling Form Data

- Use the right handler
- Do not trust anything from your form, sanitize everything
- Just available anywhere in the script
- It is that simple

# Sample

```
<form action='test.html' method='post'>  
<input type='text' size='20' name='first_name'>  
<select name='state'>  
  <option value='VA'>Virginia</option>  
  <option value='MD'>Maryland</option>  
</select>  
<input type='submit' value='Submit Form'>  
</form>
```



## Use It With Informix

- `$first_name = $_REQUEST['first_name'];`
- `$last_name = $_REQUEST['last_name'];`
- `$sqlline = "insert into users (first_name, last_name) values (?, ?)";`
- `$sth = $dbh->prepare($sqlline);`
- `$sth->execute(array($first_name, $last_name));`

# **\$\_SESSION**

- Maintains variables for a user across web page visits
- Stores a cookie on the user browser
- Matching file in the temp directory
  - Temp file stores session variables
- Global `$_SESSION` variable
- Functions to make it work:
  - `session_name('name');`
  - `session_start();`

## Example

```
' session_name('test_session');  
' session_start();  
' $_SESSION['first_name'] = $_REQUEST['first_name'];  
  
' print "First Name: ". $_SESSION['first_name'];
```

# Where To Go From Here

- Javascript/Jquery
  - What is AJAX
  - Database session storage
  - CSS

# Links

- <http://www.php.net>
- <http://framework.zend.com/>
- <http://www.zend.com/>
- <http://www.iiug.org/opensource>
- [http://pecl.php.net/package/PDO\\_INFORMIX](http://pecl.php.net/package/PDO_INFORMIX)
- [http://pecl.php.net/package/PDO\\_IBM](http://pecl.php.net/package/PDO_IBM)
- <http://www.openadmintool.org>

# Web Focused Programming With PHP

May 20 2014

**Thomas Beebe**

*Advanced DataTools Corp*

([tom@advanceddatatools.com](mailto:tom@advanceddatatools.com))

***Advanced DataTools***