CS2003
Advanced Internet Programming

Building Internet Applications
10 – Web Development with PHP (II)

Slides by Jason T. Jacques (http://www.cs.st-and.ac.uk/~jtj2)

Objectives

At the end of these lectures you should be able to:

- Implement a interactive PHP script
- Identify GET and POST requests
- Add state to HTTP interactions
- Create multi-page websites
- Send email messages

What we looked at last session

- Basic syntax
- Core functions
- File I/O
  - including structured data in CSV files
- Code reuse
- Object model

INTERACTIVITY

GET requests (I)

- GET is the typical HTTP request

```
GET /search?hl=en&q=php HTTP/1.1
Host: www.google.com
```

- Asks server for a resource, from a host
- Optionally includes a ‘query string’
  - delimited by a question mark (‘?’)
  - comprising field value pairs
    - delimited by a ampersand (‘&’)

GET requests (II)

- GET can be generated by links
  - tags
- or by forms
  - <form method="GET"> <input name="name" value="lilly"> </form>
- The query string is visible in the URL bar
- Accessed using the global $_GET array

```
<pre>
<?php

    /*
     * Example 18: GET requests
     * file: 18-get.php
     */

    if(isset($_GET["clicked"])) {
        echo "You clicked " . $_GET["clicked"];
    }

?>


<a href="18-get.php?clicked=link 1">link 1</a>
<a href="18-get.php?clicked=link 2">link 2</a>

<form method="GET">
<button type="submit"
        name="clicked"
        value="button 1">button 1</button>
<button type="submit"
        name="clicked"
        value="button 2">button 2</button>
</form>
```

POST requests (I)

- POST is used to send data to the server

```
POST /session HTTP/1.1
Host: mobile.twitter.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

username=jtjacques&password=guess
```

- Sends data to a resource on the host
- Can use a number of encodings
  - this example uses URL-encoding, like query strings

POST requests (II)

- POST is generated by forms
  - <form method="POST"> <input name="age" value="22"> </form>
- POST data is not show in the URL
- Accessed using the global $_POST array
  - can be used to upload files with the $_FILES array

```
<pre>
<?php

    /*
     * Example 19: POST requests
     * file: 19-post.php
     */

    if(isset($_POST["clicked"])) {
        echo "You clicked " . $_POST["clicked"];
    }

?>

<form method="POST">
<button type="submit"
        name="clicked"
        value="button 1">button 1</button>
<button type="submit"
        name="clicked"
        value="button 2">button 2</button>
<button type="submit"
        name="clicked"
        value="button 3">button 3</button>
</form>
```

Handling form data

- Form data is not automatically remembered between submissions
  - Problematic if user makes error on a long form
- Data should be validated and stored
  - or validated and pre-populated in the form on errors

```
<pre>
<?php

    /*
     * Example 20: Fields
     * file: 20-fields.php
     */

    $age = "";

    if( isset($_POST["age"])
            && $_POST["age"] * 1 > 0 ) {
                // * 1, coerces value to numeric
        $age = $_POST["age"];
    }

?>

<form method="POST">
Name: <input name="name">
Age : <input name="age"
        value="<?php echo $age; ?>">
<button type="submit">submit</button>
</form>
```

SESSIONS

Sessions (I)

- HTTP is a stateless protocol
  - each request has no association with prior ones
  - how do we know who is logged in and when?
- State can be added and managed using:
  - hidden form fields
    - but each request must be a form submit, no links
  - query string suffix
    - can leak session to others, looks messy
  - cookies
    - may not be enabled, not always available

Sessions (II)

- PHP can manage sessions
  - Session_start ( ) ;
- Must be started on every page
  - before any other output
- Accessed using the global $_SESSION array

Example: 21-sessions.php
21-sessions-secret.php

```php
<?php

    session_start();

    if(isset($_POST["login"])) {
        $_SESSION["authorised"] = true;
    }

    if(isset($_POST["logout"])) {
        $_SESSION["authorised"] = false;
    }

?>
<pre>

<form method="POST">
<button type="submit"
        name="login">login</button>
<button type="submit"
        name="logout">logout</button>
</form>

<a href="21-sessions-secret.php">see secrets</a>
```

```php
<?php

    session_start();

    if(isset($_SESSION["authorised"]) &&
        $_SESSION["authorised"] == true) {
        echo "The gold is buried at (7, 12)";
    } else {
        echo "I don't think you should be here";
    }

?>
<pre>
<a href="21-sessions.php">change session</a>
```

REDIRECTION

Redirection

- Send users to the correct page
  - on login, go directly to what was requested
- PHP makes use of HTTP headers for this
  - Location: <URL>
- Sent using header ( ) ;
  - must be sent before content

Example: 22-redirects.php
22-redirect-secret.php

```php
<?php

    session_start();

    if(isset($_POST["login"])) {
        $_SESSION["authorised2"] = true;
        header("Location: 22-redirect-secret.php");
        exit();
    }

?>

<form method="POST">
<button type="submit"
        name="login">login</button>
</form>
```

```php
<?php

    session_start();

    if(isset($_GET["logout"])) {
        $_SESSION["authorised2"] = false;
    }

    if(isset($_SESSION["authorised2"]) == false ||
        $_SESSION["authorised2"] == false) {
        header("Location: 22-redirect.php");
        exit();
    }

?>
<pre>
The gold is buried at (7, 12)
<a href="?logout">logout</a>
```

OTHER SERVICES

Email

- PHP can send email
  - mail ($to , $subject , $content , [$headers]) ;
- The server must be configured correctly
  - School hosts are setup correctly
  - School hosts do not send email to domains other than @st-andrews.ac.uk

```
<pre>
<?php

    /*
     * Example 23: Email
     * file: 23-email.php
     */

    if(isset($_POST["email"])) {
        mail(   $_POST["email"],                // to
                "Example 22: Email",            // subject
                "This message was sent from " . // message
                    $_SERVER["SERVER_NAME"],
                "From: email@example.com"       // header
            );
    }

?>
<form method="POST">
Send to: <input name="email">

<button type="submit">send message</button>
</form>
```

Databases

- PHP is often used with MySQL databases
- Connect. Query. Process.
  - mysqli_connect ($host , $usr , $pwd , $db)
  - mysqli_query ($link , $query) ;
  - mysqli_fetch_assoc ($result) ;
- Remember to clean any user provided data
  - mysqli_real_escape_string ($link , $string) ;

```
<pre>
<?php

    /*
     * Example 24: Databases
     * file: 24-database.php
     * note: this will fail to connect
     */

    $link = mysqli_connect( $host, $user, $pass, $db );

    $bad_data = "' OR 1=1 -- in ur codes, sql injecting";
        // bad user, trying to break into your server!

    $good_data = mysqli_real_escape_string($bad_data);
        // nice, safe data :)

    $query = "SELECT * from `user` where password = '" .
            $good_data . "'";

    $result = mysqli_query($link, $query);

    $array =  mysqli_fetch_assoc($result);

    print_r($array)

?>
```

SECURITY

Security

- Never, ever, trust the user
  - Some users will **try** to break your system
- Always filter user input
  - Expect id=x to be a number? check it
    - e.g. intval ($data) ;
- Don't just echo user data to the browser
  - Vulnerable to XSS (cross site scripting)
    - use htmlentities ($data) ; or strip_tags ($data) ;
- POST is not security
  - Data is sent 'over the wire' in the clear
    - use HTTP**S** for sensitive data where possible

QUIZ

Quiz

What variable should be used to access the data submitted with this form?

```
<form method="POST">
    <input name="car">
    <button type="submit">submit</button>
</form>
```

A: $_GET["car"]

B: $_POST["car"]

C: $submit

D: $car

Quiz

What variable should be used to access the data submitted with this form?

```
<form method="POST">
    <input name="car">
    <button type="submit">submit</button>
</form>
```

Method is POST

A: $_GET["car"]

B: $_POST["car"]

C: $submit

D: $car

Quiz

What variable is used to access the data sent in these query strings?

```
<a href="?brand=coca-cola">coca-cola</a>
<a href="?brand=pepsi">pepsi</a>
```

A: $_GET["brand"]
B: $_GET["coca-cola"]
C: $_GET["pepsi"]
D: $brand

Quiz

What variable is used to access the data sent in these query strings?

```
<a href="?brand=coca-cola">coca-cola</a>
<a href="?brand=pepsi">pepsi</a>
```

Data is declared:
key=value

A: $_GET["brand"]

B: $_GET["coca-cola"]

C: $_GET["pepsi"]

D: $brand

Quiz

Does this code work correctly?

```
<pre>
<?php
    session_start();

    if(isset($_SESSION["authorised"]) &&
        $_SESSION["authorised"] == true) {
        echo "Hello, Boss!";
    }
?>
```

Yes

No

Quiz

Does this code work correctly?

```
<pre>
<?php
    session_start();

    if(isset($_SESSION["authorised"]) &&
       $_SESSION["authorised"] == true) {
       echo "Hello, Boss!";
    }
?>
```

There is output before session_start();

Yes

No