

Lecture 9

PHP 3 – PHP Include and Managing State



Topics



- PHP Includes
- Managing 'state' between client and server
 - hidden fields
 - query strings
 - sessions

- Facilitates the reuse of PHP code at the files level
- Useful for including recurring functionality or content e.g. menus

PHP include example

```
<!DOCTYPE html>
<html lang="en">
<head>
    ...
</head>
<body>
    <?php
        include_once ("php_menu.html");
    ?>
    <!-- Web page starts here -->
    <h1>Input checking using input values</h1>
    ...
</html>
```

`include_once` ensures that the code is only included once

Whatever text is in the file `php_menu.html` will be inserted at this point

Here file is named `.html` could be `php_menu.inc`

PHP include and require



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
<?php
```

```
require ("php_menu.html");
```

```
?>
```

```
<!-- Web page starts here -->
```

```
<h1>Input checking using input values</h1>
```

```
...
```

```
</html>
```

Same as include but will
produce a fatal error if the file
is missing

Name file .php if it needs
to be processed



Topics



- PHP Includes
- Managing 'state' between client and server
 - ➡ —hidden fields
 - query strings
 - sessions



Techniques for **maintaining state** information with PHP include:

- Hidden form fields
- Query strings
- Sessions

Understanding State Information

- HTTP was originally designed to be **stateless** – *Web browsers store no persistent data about a visit to a Web site*
- We need techniques to **maintaining state**: *i.e. store persistent information about Web site visits, that can be passed backwards and forwards between the client and the server.*
- We have previously used Web Storage and Cookies to store information locally on the client
- Information about individual visits to a Web site also needs to be maintained on the server



Some reasons why a web application may need to **maintain state** information:

- Temporarily store information as a user navigates through a multi-page form
- Allow a user to create bookmarks for returning to specific locations within a Web site
- Customize individual Web pages based on user preferences
- Provide shopping carts that store order information
- Store user IDs and passwords
- Use counters to keep track of how many times a user has visited a site



Using Hidden Form Fields to Save State



- Use hidden form fields to temporarily store data that needs to be sent to a server that a user does not need to see
- Examples include the result of a calculation
- Create hidden form fields with the `<input />` element using `type="hidden"`

```
<input type="hidden"  
      name="..." value="..." />
```

Both **name** and **value** attributes are needed.



Using Hidden Form Fields to Save State



- When submitted to a PHP script, access the values submitted from the form with the **`$_GET[]`** and **`$_POST[]`** Superglobals
- Pass the form values from one PHP script to another PHP script, by storing the *name-values* in **`input`** elements with **`type="hidden"`**.

Using Hidden Form Fields to Save State



```
<form action="toolLoans.php" method="get">
. . .
<p>
<input type="hidden" name="toolID"
        value="<?php echo $tool_id ?>" />
<input type="submit" value="Hire Tool" />
</p>
</form>
```

Note: The hidden value will be visible if you “view page source” on the client.

- PHP Includes
- Managing 'state' between client and server
 - hidden fields
 - query strings
 - sessions



Using Query Strings to Save State

- A **query string** is a set of name=value pairs *appended to a target URL*
- A **query string** consists of a single text string containing one or more pieces of information
- Any forms that are submitted with the **GET** method automatically add a question mark (?) and append the **query string** to the URL of the server-side script



Using Query Strings to Save State

- To pass information from one Web page to another using a query string,
 - add a question mark (?) immediately after the URL
 - followed by the query string containing the information in name=value pairs, and
 - separate the name=value pairs within the query string by ampersands (&)

```
<a href="details.php?firstName=John&lastName=Smith
&occupation=singer&userid=jfodhhreiowhy823y843">Jo
hn Smith</a>
```



Using Query Strings to Save State

- To pass query string information from one PHP script to another PHP script, echo the values in the first script

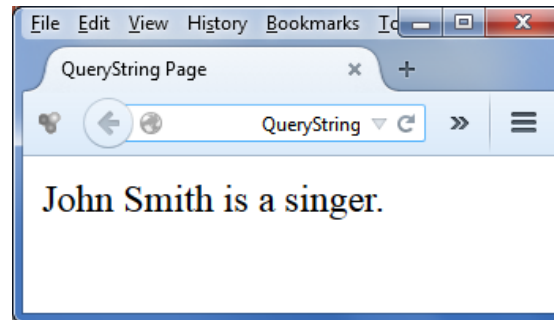
```
<a href="details.php?firstName=<?php echo $fname; ?>
&lastName=<?php echo $lname; ?>
&occupation=<?php echo $occ; ?>">
<?php echo $fname, $lname; ?></a>
```

Note: The values will be visible in the query string.

Using Query Strings to Save State



```
echo "{$_GET['firstName']} {$_GET['lastName']}",  
      "is a {$_GET['occupation']}. ";
```



Output of the contents of a query string

Topics



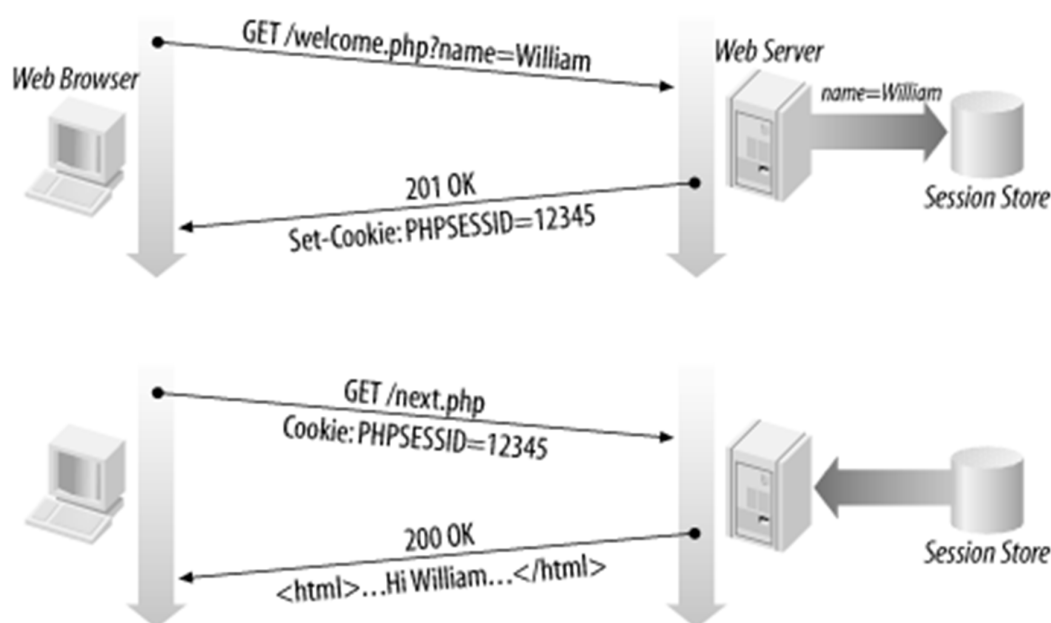
- PHP Includes
- Managing 'state' between client and server
 - hidden fields
 - query strings
 - sessions



Using Sessions to Save State

- A **session** refers to a period of activity when a PHP script stores *state information on a Web server*
- **Sessions** allow you to maintain state information *even when clients disable cookies in their Web browsers*

Session interaction



Starting a Session



```
<?php
session_start();

...
?>
<p><a href='<?php echo
    "occupation.php?PHPSESSID="
    . session_id() ?>'>Occupation</a></p>
```



Starting a Session



- The **session_start()** function starts a new session or continues an existing one
- The **session_start()** function generates a unique session ID to identify the session
- A **session ID** is a random alphanumeric string that looks something like:

7f39d7dd020773f115d753c71290e11f

- The **session_start()** function creates a text file on the Web server that is the same name as the session ID, preceded by **sess_**



Starting a Session (continued)



- Session ID text files are stored in the Web server directory specified by the **`session.save_path`** directive in your `php.ini` configuration file
- The **`session_start()`** function does not accept any arguments, nor does it return a value that you can use in your script

```
<?php
session_start() ;
...
```



Starting a Session (continued)



- You must call the **`session_start()`** function ***before*** you send the Web browser any output
- If a client's Web browser is configured to accept cookies, the session ID is assigned to a temporary cookie named `PHPSESSID`
- Pass the session ID as a query string or hidden form field to any Web pages that are called as part of the current session



Working with Session Variables



- Session state information is accessed using the **\$_SESSION** superglobal
- When the **session_start()** function is called, PHP either initializes a new **\$_SESSION** superglobal or retrieves any variables for the current session (based on the session ID) into the **\$_SESSION** superglobal



Working with Session Variables (continued)



```
<?php
session_set_cookie_params(3600);
session_start();
$_SESSION['firstName'] = "John";
$_SESSION['lastName'] = "Smith";
$_SESSION['occupation'] = "singer";
?>
<p><a href='<?php echo "Occupation.php?"
. session_id() ?>'>Occupation</a></p>
```

Sets the “lifetime” argument to 3600 seconds



Working with Session Variables (continued)



- Use the **isset()** function to ensure that a session variable is set before you attempt to use it

```
<?php
```

```
session_start();  
if (isset($_SESSION['firstName']) &&  
    isset($_SESSION['lastName'])  
    && isset($_SESSION['occupation']))  
    echo "<p>" . $_SESSION['firstName'] . " "  
        . $_SESSION['lastName'] . " is a "  
        . $_SESSION['occupation'] . "</p>";
```

```
?>
```



Deleting a Session (continued)



```
<?php
```

```
session_start();  
$_SESSION = array();  
session_destroy();
```

```
?>
```

Step 1

Step 2: Use the array() construct to reinitialize the \$_SESSION superglobal

Step 3: Delete the session

This is the code often used for a “Log-out” script, or the code that is included in a “Registration” / “Log In” page, so that it deletes any existing user sessions whenever a user opens it.



PHP Syntax Checking



- <http://phpcodechecker.com/>