Server-side Web Development 1: PHP finale

Backend Web Development

by: Salahuddin ElKazak

Classes & Objects contd.

OOP Finale

Lecture 4

Static

```php
class MathHelper {
    // Static property
    public static $pi = 3.14159;

    // Static method to calculate the area of a circle
    public static function calculateCircleArea($radius) {
        return self::$pi * ($radius * $radius);
    }
}
```

Inheritance

```php
// Parent class
class Vehicle {
    public $brand;
    public $model;
    public function __construct($brand, $model) {
        $this->brand = $brand;
        $this->model = $model;
    }
    public function start() {
        return "The vehicle is starting...";
    }
}
```

```php
// Child class inherits from Vehicle
class Car extends Vehicle {
    public $doors;
    public function __construct($brand, $model, $doors) {
        // Calling parent constructor
        parent::__construct($brand, $model);
        $this->doors = $doors;
    }
    public function start() {
        return "The car is starting with a roar!";
    }
    public function displayDetails() {
        return "Brand: $this->brand, Model: $this->model, Doors: $this->doors";
    }
}
```

$_GET and *$_POST* Superglobal Arrays

Lecture 4

Superglobal Arrays

Name	Description
$GLOBALS	Array for storing data that needs superglobal scope
$_COOKIES	Array of cookie data passed to page via HTTP request
$_ENV	Array of server environment data
$_FILES	Array of file items uploaded to the server
$_GET	Array of query string data passed to the server via the URL
$_POST	Array of query string data passed to the server via the HTTP header
$_REQUEST	Array containing the contents of $_GET, $_POST, and $_COOKIES
$_SESSION	Array that contains session data
$_SERVER	Array containing information about the request and the server

Determining if Any Data was Sent (Quick Lab Exercise)

```
<!DOCTYPE html>
<html>
<body>
<?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ( isset($_POST["uname"]) && isset($_POST["pass"]) ) {
      // handle the posted data.
      echo "handling user login now ...";
      echo "... here we could redirect or authenticate ";
      echo " and hide login form or something else";
    }
  }
?>
    <h1>Some page that has a login form</h1>
    <form action="samplePage.php" method="POST">
          Name <input type="text" name="uname"><br>
          Pass <input type="password" name="pass"><br>
          <input type="submit">
    </form>
  </body>
</html>
```

Determining if Any Data was Sent (Quick Lab Exercise Solution)

```
<!DOCTYPE html>
<html>
<body>
<?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ( isset($_POST["uname"]) && isset($_POST["pass"]) ) {
      // handle the posted data.
      $username = $_POST["uname"];
      $password = $_POST["pass"];
       $users = ["user1"=>"pass1", "testuser"=>"topsecret"];
       if (array_key_exists($username, $users) &&
$users[$username] == $password) {
              echo "Logged in!";
         } else {
              echo "Invalid Login";
       }
    }
  } ?>
    <h1>Some page that has a login form</h1>
    <form action="samplePage.php" method="POST">
          Name <input type="text" name="uname"><br>
          Pass <input type="password" name="pass"><br>
          <input type="submit">
    </form>
  </body>
</html>
```

Accessing Form Array Data

```
<form method="get">
  Please select days of the week you are free.<br>
  Monday <input type="checkbox" name="day" value="Monday"> <br>
  Tuesday <input type="checkbox" name="day" value="Tuesday"> <br>
  Wednesday <input type="checkbox" name="day" value="Wednesday"> <br>
  Thursday <input type="checkbox" name="day" value="Thursday"> <br>
  Friday <input type="checkbox" name="day" value="Friday"> <br>
  <input type="submit" value="Submit">
</form>
```

What happens if you try to access `$_GET['day']`?

What if you change the name to "day[]"?

Now try both with this code:

```
echo "You submitted " .
count($_GET['day']) . "values";
foreach ($_GET['day'] as $d) {
  echo $d . " <br>";
}
```

Using Query Strings

- Try the following simple html with query strings

```
<a href='page2.php?search=1234&color=yellow'>Yellow 1234</a><br/>

<a href='page2.php?search=mazda&color=red'>Red Mazda</a><br/>

<a href='page2.php?search=rose&color=blue'>Blue Rose</a><br/>
```

- Then build page2.php to grab the GET values "search" and "color" and handle them
- Can we use this to make google-search links?
  - Google uses the google.com/search url with the "q" variable as a query string

Sanitizing Query Strings

- Query strings automatically are converted into GET requests by the browser with data encoded as key=value pairs with & signs separating them
- A simple code injection is to inject "`<script>alert('this is a js injection in a query string');</script>`" in our previous example, this is a dangerous vulnerability!
- How to "sanitize" query strings (or any inputs from the user)?
  - Write a simple function that removes unexpected characters, e.g. you would not expect other than alphanumeric strings in a simple color query
  - You can use php filter functions (search php.net) or the ctype_alpha function (https://www.php.net/manual/en/function.ctype-alpha.php) or your own!

Working with the HTTP Header

Lecture 4

Redirect Using Location Header

```
header("Location: url.php");
```

- This sets the new location to a new url/page, you can use this to redirect the user to another page
  - Could be a 404 not found page in case they are looking for something that does not exist
  - Could be a 403 not permitted page if they are trying to access something they are not allowed to access
  - Could be a simple redirect to the mainpage on a failed login or timeout

Setting the Content-Type Header

- You can inform the browser (or any API caller otherwise) of what content-type to expect, accordingly the browser could behave differently, for example
  - if it's an xls file, it would download it as an excel file,
  - if it's a json file, it would be able to display it in a pretty-printed format,
  - if it's a pdf/img it could display and offer a download option

```
<?php
$books = array();
$books[] = ["title"=>"Basics of Web Design","year"=>2014,"pages"=>400];
$books[] = ["title"=>"Database Processing","year"=> 2012,"pages"=>630];
$books[] = ["title"=>"Development Economics","year"=>2014,"pages" => 760];
header('Content-Type: application/json');
echo json_encode($books, JSON_NUMERIC_CHECK);
?>
```