

CS 312

Internet Concepts

Web Programming/Scripting: PHP and AJAX Refresher

Dr. Michele Weigle

Department of Computer Science

Old Dominion University

mweigle@cs.odu.edu

<http://www.cs.odu.edu/~mweigle/CS312-F11>

1

Outline

- ◆ PHP refresher
 - » PHP and forms
 - » PHP and XML

- ◆ AJAX refresher
 - » XMLHttpRequest
 - ❖ sendXHR()
 - » Data flow

2

PHP Refresher

- ◆ PHP scripts are executed on the web server
- ◆ An HTTP request is sent in order to execute the script
 - » directly - user navigates to a .php URL
 - » indirectly - client-side code (such as JavaScript or HTML form) issues an HTTP request
- ◆ The output of the script is returned in the HTTP response
 - » directly - PHP code is embedded in page and result is returned to client (using .php URL)
 - » indirectly - using echo statements in PHP script


3

PHP and Forms

- ◆ Use a separate PHP file (not embedded in HTML)
- ◆ HTTP request sent to PHP script when user presses Submit button

```
<form action="action.php" method="post">  
  <p>Your name: <input type="text" name="name" /> </p>  
  <p>Your age: <input type="text" name="age" /> </p>  
  <p><input type="submit" /> </p>  
</form>
```

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form.html>



Your name:

Your age:

<http://us.php.net/tut.php>

4

PHP - post vs. get

- ◆ These are related to the HTTP request method used
- ◆ post
 - » parameters are transferred in the entity body
- ◆ get
 - » parameters are transferred in the URL

5

HTTP Request Message Refresher

- » Request line
 - » Optional header lines
 - » Present only for some methods (e.g., POST)
- ```
method <SP> path <SP> version <CR><LF>
header field name ":" value <CR><LF>
 ⋮
header field name ":" value <CR><LF>
<CR><LF>
entity body
```

6

# HTTP Response Message

## Refresher

---

- » Status line
- » Optional header lines
- » Requested object, error message, etc.

```
version <SP> code <SP> phrase <CR><LF>
header field name ":" value <CR><LF>
:
header field name ":" value <CR><LF>
<CR><LF>
entity body
```

7

## PHP and Forms

### Using post

---

```
<form action="action.php" method="post">
 <p>Your name: <input type="text" name="name" /> </p>
 <p>Your age: <input type="text" name="age" /> </p>
 <p><input type="submit" /> </p>
</form>
```

Your name:

Your age:

# PHP and Forms

## Using post

Hi <?php echo (\$\_POST['name']); ?>.  
You are <?php echo (int)\$\_POST['age']; ?> years old.

<http://www.cs.odu.edu/~mweigle/cs312/php/action.php>

- ◆ \$\_POST
  - » when "post" method is used
- ◆ \$\_GET
  - » when "get" method is used
- ◆ \$\_REQUEST
  - » when either "get" or "post" is used

[http://www.w3schools.com/PHP/php\\_get.asp](http://www.w3schools.com/PHP/php_get.asp)

9

# PHP and Forms

## Using post

Your name:

Your age:

Hi John Smith. You are 21 years old.

The screenshot shows the Chrome Developer Tools interface. The 'Network' tab is active, displaying a list of requests. The selected request is 'action.php' with a status of 200 OK. The 'Headers' pane is expanded, showing the 'Request Headers' section. The 'POST' method is highlighted, and the 'Host' is 'www.cs.odu.edu'. The 'Form Data' section is also expanded, showing the submitted data: 'name: John+Smith' and 'age: 21'. The 'Response Headers' section shows the server's response, including the date, server type, and content type.

Chrome > View > Developer > Developer Tools

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form.html>

10

# PHP and Forms

## HTTP Response Body

Hi John Smith. You are 21 years old.

Your name:

Your age:

Submit Query

Elements Resources Network Scripts Timeline Profiles Audits

Name Path

action.php  
/~mweigle/cs312/php

Headers Preview **Response** Cookies Timing

1 Hi John Smith.

2 You are 21 years old.

3

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form.html>

11

# PHP and Forms

## Using get

```
<form action="action-get.php" method="get">
 <p>Your name: <input type="text" name="name" /> </p>
 <p>Your age: <input type="text" name="age" /> </p>
 <p><input type="submit" /> </p>
</form>
```

Your name:

Your age:

Submit Query

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form-get.html>

12

# PHP and Forms

## Using get

```
Hi <?php echo ($_GET['name']); ?>.
You are <?php echo (int)$_GET['age']; ?> years old.
```

<http://www.cs.odu.edu/~mweigle/cs312/php/action-get.php>

- ◆ \$\_POST
  - » when "post" method is used
- ◆ \$\_GET
  - » when "get" method is used
- ◆ \$\_REQUEST
  - » when either "get" or "post" is used

[http://www.w3schools.com/PHP/php\\_get.asp](http://www.w3schools.com/PHP/php_get.asp)

13

# PHP and Forms

## Using get

Hi John Smith. You are 21 years old.

Your name:

Your age:

The screenshot shows a web browser window with the URL <http://www.cs.odu.edu/~mweigle/cs312/php/action-get.php>. The page displays the text "Hi John Smith. You are 21 years old." and a form with two input fields: "Your name: John Smith" and "Your age: 21", followed by a "Submit Query" button. The browser's developer tools are open, showing the Network tab. The selected request is a GET request to the same URL. The Request Headers section is expanded, showing the request method (GET), status code (200 OK), and various headers including Host, Connection, User-Agent, Accept, Referer, Accept-Encoding, Accept-Language, and Accept-Charset. The Query String Parameters section is also expanded, showing the parameters "name=John+Smith" and "age=21". The Response Headers section is expanded, showing the response status (HTTP/1.1 200 OK), Date, Server, X-Powered-By, Content-Length, Keep-Alive, Connection, and Content-Type.

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form-get.html>

14

# PHP and Forms

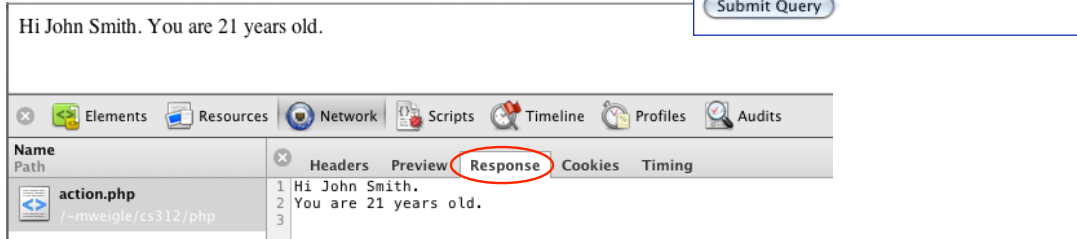
## HTTP Response Body

Hi John Smith. You are 21 years old.

Your name:

Your age:

Submit Query



Exactly the same as php-form.html (that uses post)

Only difference is in how the HTTP request is formed and how the PHP script gets its parameters.

<http://www.cs.odu.edu/~mweigle/cs312/php/php-form-get.html>

15

## Outline

- ◆ PHP refresher
  - » PHP and forms
  - » PHP and XML
- ◆ AJAX refresher
  - » XMLHttpRequest
    - ❖ sendXHR()
  - » Data flow

16



# PHP and XML

## SimpleXML

---

- ◆ Must know the layout of the XML file
- ◆ Load XML file
  - » `$xml = simplexml_load_file ("url of xml file");`

<http://php.net/manual/en/book.simplexml.php>

17

## Simple XML

### Functions

---

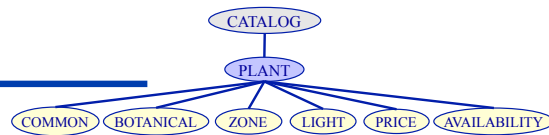
- ◆ Get the name of the first element
  - » `$xml->getName()`
- ◆ Loop through each child
  - » `foreach ($xml->children() as $child)`
    - ❖ to access label: `$child->getName()`
    - ❖ to access data: `$child`
- ◆ To get children of child
  - » `$child->children()`

18

# Simple XML

## Example

```
<?php
$xml = simplexml_load_file("plant_catalog.xml");
// access each plant
foreach ($xml->children() as $plant) {
 // access each element of the plant
 foreach ($plant->children() as $child) {
 // print label: data
 echo $child->getName() . ": " . $child . "
";
 }
}
?>
```



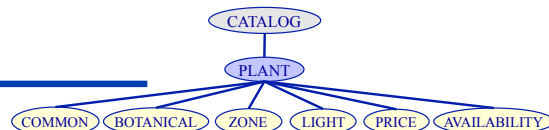
<http://www.cs.odu.edu/~mweigle/cs312/php/plants.php>

<http://www.cs.odu.edu/~mweigle/cs312/php/plants.php.txt>

19

# SimpleXML

## XPath



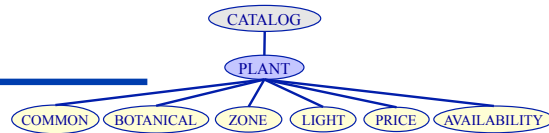
- ◆ After loading the xml file with SimpleXML, use XPath to find particular elements in the XML.
- ◆ Find all PLANT nodes whose common names are Bloodroot
  - » `$plant = $xml->xpath("//PLANT [COMMON='Bloodroot']");`
    - ❖ // - selects nodes in the document from the current node that match the selection no matter where they are (can be current node, root, children, or parent)
    - ❖ returns an array
- ◆ Access the price of the first plant that matches
  - » `$price = (string) $plant[0]->PRICE;`

[http://www.w3schools.com/XPath/xpath\\_syntax.asp](http://www.w3schools.com/XPath/xpath_syntax.asp)

20

# XPath

## Example



- ◆ Find all PLANT nodes with "Mostly Shady" light

```
$plant = $xml->xpath ("//PLANT [LIGHT='Mostly Shady']");
```

- ◆ Display all plants that match

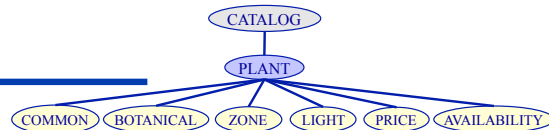
```
$numPlants = count ($plant);
for ($i=0; $i<$numPlants; $i++) {
 echo "<p>"
 foreach ($plant[$i]->children() as $child) {
 echo $child->getName() . ": " . $child . "
";
 }
 echo "</p>";
}
```

<http://www.cs.odu.edu/~mweigle/cs312/php/plants-xpath.php>  
<http://www.cs.odu.edu/~mweigle/cs312/php/plants-xpath.php.txt>

21

# Xpath

## More Examples



- ◆ Find all PLANT nodes

```
$plant = $xml->xpath ("//PLANT");
```

- ◆ Find all PLANT nodes where ZONE > 2

```
$plant = $xml->xpath ("//PLANT [ZONE > 2]");
```

- ◆ Find all PLANT nodes that are available in Feb

```
$plant = $xml->xpath ("//PLANT [substring (AVAILABILITY, 1, 2)='02']");
```

<http://www.cs.odu.edu/~mweigle/cs312/php/plants-xpath-more.php>

<http://www.cs.odu.edu/~mweigle/cs312/php/plants-xpath-more.php.txt>

String functions: [http://www.w3schools.com/XPath/xpath\\_functions.asp#string](http://www.w3schools.com/XPath/xpath_functions.asp#string)

22

# PHP and Xpath

## Questions

---

XPath is used in PHP to search/parse XML files

- ◆ Where are the XPath commands executed? Client or server? Why?
- ◆ Explain the data flow when using XPath. Is the XML document sent to the client?

23

## Outline

---

- ◆ PHP refresher
  - » PHP and forms
  - » PHP and XML
- ◆ AJAX refresher
  - » XMLHttpRequest
    - ❖ sendXHR()
  - » Data flow

24

# What is AJAX?

---



- ◆ Asynchronous JavaScript And XML
- ◆ Uses JavaScript to send and receive data between a web browser and web server
- ◆ Uses the JavaScript XMLHttpRequest (XHR) object for data transfer

<http://www.w3schools.com/ajax/>

25

## AJAX

### Remember XMLHttpRequest (XHR)?

---

```
function sendXHR(url)
{
 if (xmlhttp == null) {
 xmlhttp = new XMLHttpRequest();
 }
 if (xmlhttp == null) {
 alert("Your browser does not support XMLHttpRequest.");
 return;
 }
 xmlhttp.onreadystatechange = stateChange;
 xmlhttp.open("GET", url, true);
 xmlhttp.send(null);
}
```

requests the document found at the given URL

xmlhttp is the variable that allows us to make the request

stateChange function is called when the state of the XMLHttpRequest object changes

open() sets up an HTTP request for the URL

send() sends the request to the server

<http://www.cs.odu.edu/~mweigle/cs312/xml/xmlHelperFns.js>

26

# AJAX

## Remember XMLHttpRequest (XHR)?

---

```
function stateChange()
{
 if (xmlhttp.readyState == 4) {
 if (xmlhttp.status == 200) {
 // HTTP response code (200 is OK)
 doStuff(); // you must write this function
 } else {
 alert ("Problem retrieving data: " + xmlhttp.statusText);
 }
 }
}
```

### readyState

0 - request is not initialized  
1 - request has been set up  
2 - request has been sent  
3 - request is in process  
4 - request is complete

<http://www.cs.odu.edu/~mweigle/cs312/xml/xmlHelperFns.js>

27

## XMLHttpRequest and sendXHR()

---

- ◆ XMLHttpRequest
  - » built-in JavaScript object for contacting the server from inside JavaScript
- ◆ sendXHR()
  - » function that I wrote as a wrapper for creating an XMLHttpRequest and accessing a specific file on the server
- ◆ When to use sendXHR()?
  - » anytime you need to access a file or execute a script on the server from JavaScript
  - » you do not have to be using XML to use sendXHR()
  - » you do not have to be calling a PHP script to use sendXHR()

28

# Uses of sendXHR()

## From Our Previous Examples

---

- ◆ Change a webpage without reloading
  - » <http://www.cs.odu.edu/~mweigle/cs312/xml/text-noreload.html>
- ◆ View HTTP response header
  - » <http://www.cs.odu.edu/~mweigle/cs312/xml/get-header.html>
- ◆ AJAX type-ahead suggestion box
  - » uses JavaScript to send user input to PHP script for processing (without using Submit button) and display response to web page
  - » does not use XML
  - » <http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/suggest.html>
- ◆ AJAX drop-down box
  - » uses JavaScript to send user input to PHP script for processing (without using Submit button) and display response to web page
  - » PHP script on the server uses XML
  - » <http://www.cs.odu.edu/~mweigle/cs312/ajax/plants/plants.html>

29

## AJAX Suggest Example

### Type-Ahead Searching

---

- ◆ Like Google Suggest
  - » start typing in textbox and suggested words/phrases appear below

First Name: <input type="text" value="jo"/>
Suggestions: John, Joe, Joseph

<http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/suggest.html>

# AJAX Suggest Example

## Components

---

First Name:

Suggestions: John, Joe, Joseph

### ◆ HTML

- » provide form with a text box for user to enter a name
- » provide named element to display suggestions

### ◆ JavaScript (**client-side** operations)

- » `showHint()`
  - ❖ take data from HTML form and send to PHP script using `sendXHR`
- » `doStuff()`
  - ❖ receive reply from PHP script
  - ❖ display result on webpage in named HTML element

### ◆ PHP script (**server-side** operations)

- » create array of available names
- » search array for those starting with data sent by JavaScript
- » return the list of matching names

31

# AJAX Suggest Example

## HTML Setup

---

First Name:

Suggestions: John, Joe, Joseph

- ◆ Load JavaScript helper functions
- ◆ Setup for embedded JavaScript (or the functions can be put into an external JavaScript file)

```
<head>
<title>Ajax Suggest Example</title>
<script type="text/javascript" src="http://www.cs.odu.edu/~mweigle/cs312/xml/
xmlHelperFns.js">
</script>

<script type="text/javascript">
<!--
 put showHint() and doStuff() functions here
// -->
</script>
</head>
```

32



# AJAX Suggest Example

## HTML

First Name:

Suggestions: John, Joe, Joseph

- ◆ Provide form with a text box for user to enter a name
- ◆ Provide named element to display suggestions

```
<body>
```

```
<form action ="">
```

First Name:

*this.value is the text currently in the box*

```
<input type="text" id="txt1" onkeyup="showHint(this.value)" />
```

```
</form>
```

```
<p>Suggestions: </p>
```

```
</body>
```

33

# AJAX Suggest Example

## JavaScript

First Name:

Suggestions: John, Joe, Joseph

```
var xmlhttp = null; // must be outside function to be global
```

```
function showHint(str)
```

send data from HTML form to PHP script

```
{
```

```
 // check to see if textbox is blank
```

```
 if (str.length==0) {
```

```
 document.getElementById("txtHint").innerHTML="";
```

```
 return;
```

```
 }
```

```
 // build request to run PHP script on server
```

```
 var url = "gethint.php";
```

*adds parameter 'q' to URL*

```
 url = url + "?q=" + str;
```

*adds random number to prevent using cached file*

```
 url = url + "&sid=" + Math.random(); // avoid browser cache
```

```
 sendXHR(url);
```

sendXHR() in xmlHelperFns.js

```
}
```

34

# AJAX Suggest Example

## JavaScript

First Name:

Suggestions: John, Joe, Joseph

- ◆ Receive reply from PHP script
- ◆ Display result in named HTML element

```
function doStuff()
{
 document.getElementById("txtHint").innerHTML = xmlhttp.responseText;
}
```

responseText holds everything printed (echo) by the PHP script

35

# AJAX Suggest Example

## PHP (gethint.php)

First Name:

Suggestions: John, Joe, Joseph

```
<?php
// Fill up array with names
$names = array ("Anna", "Brittany", "Diana", ...); rest of names omitted for space

// get the q parameter from the URL
$q = $_GET["q"];

// lookup all hints from array if length of q > 0
if (strlen ($q) > 0) {
 $hint = "";
 for ($i=0; $i<count($names); $i++) {
 if (strtolower($q) == strtolower(substr($names[$i], 0, strlen($q)))) {
 if ($hint == "") {
 $hint = $names[$i];
 } else {
 $hint = $hint . " , " . $names[$i];
 }
 }
 }
}
```

create array of available names

search for names that start with the data from the URL

- *count(\$arr)* returns number of elements in \$arr
- *strtolower(\$str)* returns the lowercase version of \$str
- *substr(\$str, \$start, \$end)* returns characters in positions \$start to \$end from \$str

36

# AJAX Suggest Example

## PHP (gethint.php *continued*)

First Name:

Suggestions: John, Joe, Joseph

```
// Set output to "no suggestion" if no hint was found
// or to correct value
if ($hint == "") {
 $response = "no suggestion";
} else {
 $response = $hint;
}

// output the response
echo $response;

return list of names

?>
```

<http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/suggest.html>

<http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/gethint.php.txt>

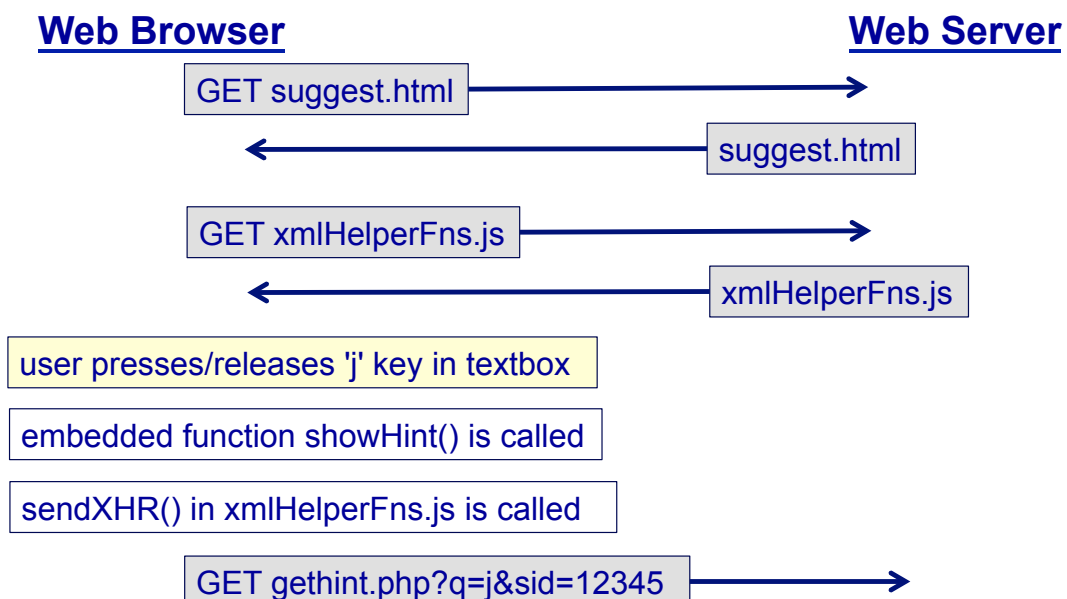
37

# AJAX Suggest Example

## Data Flow

First Name:

Suggestions: John, Joe, Joseph



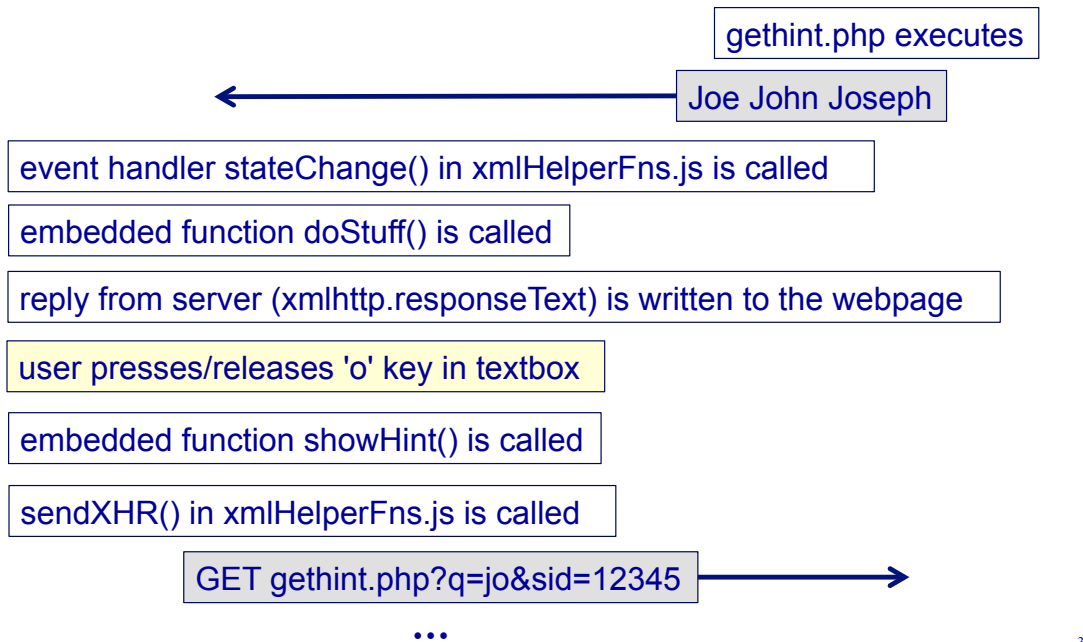
38

# AJAX Suggest Example

## Data Flow (continued)

### Web Browser

### Web Server



39

# AJAX Suggest Example

## Data Trace

First Name:

Suggestions: John, Joe, Joseph





Name Path	Meth...	Status Text	Type	Initiator	Size Conten	Time Latency	Timeline	1.51s	2.27s	3.03s	3.78s
suggest.html /~mweigle/cs312/ajax	GET	304 Not Mod	text/...	Other	219B 1.07KB	26ms 25ms					
xmlHelperFns.js /~mweigle/cs312/xml	GET	304 Not Mod	appli...	suggest.html: Parser	218B 1.60KB	32ms 27ms					
gethint.php /~mweigle/cs312/ajax	GET	200 OK	text/...	xmlHelperFns: Script	270B 19B	33ms 33ms					
gethint.php /~mweigle/cs312/ajax	GET	200 OK	text/...	xmlHelperFns: Script	270B 19B	36ms 35ms					

Chrome > View > Developer > Developer Tools

40

# AJAX Suggest Example

## Data Trace

Name	Path	Headers	Preview	Response	Cookies	Timing
	suggest.html	Request URL: http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/gethint.php?q=j&sid=0.2973918674979359 Request Method: GET Status Code: 200 OK				
	xmlHelperFns.js	/~mweigle/cs312/xml				
	gethint.php	/~mweigle/cs312/ajax/sugg				
	gethint.php	/~mweigle/cs312/ajax/sugg				

Request Headers view parsed

GET /~mweigle/cs312/ajax/suggest/gethint.php?q=j&sid=0.2973918674979359 HTTP/1.1

Host: www.cs.odu.edu

Connection: keep-alive

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_7\_2) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2

Accept: \*/\*

Referer: http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/suggest.html

Accept-Encoding: gzip, deflate, sdch

Accept-Language: en-US,en;q=0.8

Accept-Charset: ISO-8859-1, utf-8;q=0.7,\*;q=0.3

Cookie: txtMainTab=Timeline; \_\_utma=35744766.580760710.1270506004.1319999346.1321654293.39; \_\_utmz=35744766.1315504338.37.17.utmcsr=cs.odu.edu|utmccn=(referral)|utmcmd=referral|utmcct=/~mweigle/CS795-F11/Sched; author=mweigle; \_\_utma=256360345.1284928110.1285499037.1321824230.1322006984.250; \_\_utmz=256360345.1319416285.241.76.utmcsr=oducs-networking.blogspot.com|utmccn=(referral)|utmcmd=referral|utmcct=/

Query String Parameters view decoded

q: j

sid: 0.2973918674979359

Response Headers view parsed

HTTP/1.1 200 OK

Date: Sun, 27 Nov 2011 19:15:54 GMT

Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod\_ssl/2.2.17 OpenSSL/0.9.8q

X-Powered-By: PHP/5.3.5

Content-Length: 19

Keep-Alive: timeout=5, max=98

Connection: Keep-Alive

Content-Type: text/html





41

# AJAX Suggest Example

## Data Trace

First Name:

Suggestions: John, Joe, Joseph

Name	Path	Headers	Preview	Response	Cookies	Timing
	suggest.html	1 John, Joe, Joseph				
	xmlHelperFns.js	2				
	gethint.php	/~mweigle/cs312/ajax/sugg				
	gethint.php	/~mweigle/cs312/ajax/sugg				

42

# AJAX Suggest Example

## Data Trace

The screenshot shows the Network tab of a web browser. The left pane lists the following resources:

- suggest.html (/~mweigle/cs312/ajax/sugg)
- xmlHelperFns.js (/~mweigle/cs312/xml)
- gethint.php (/~mweigle/cs312/ajax/sugg)
- gethint.php (/~mweigle/cs312/ajax/sugg)

The right pane shows the details for the selected resource, **suggest.html**. The Request tab is active, displaying the following information:

- Request URL: `http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/gethint.php?q=jo&sid=0.48900638381019235`
- Request Method: GET
- Status Code: 200 OK
- Request Headers** (view parsed):
  - GET /~mweigle/cs312/ajax/suggest/gethint.php?q=jo&sid=0.48900638381019235 HTTP/1.1
  - Host: www.cs.odu.edu
  - Connection: keep-alive
  - User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_7\_2) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2
  - Accept: \*/\*
  - Referer: http://www.cs.odu.edu/~mweigle/cs312/ajax/suggest/suggest.html
  - Accept-Encoding: gzip, deflate, sdch
  - Accept-Language: en-US,en;q=0.8
  - Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.3
  - Cookie: txtMainTab=Timeline; \_\_utma=35744766.580760710.1270506004.1319999346.1321654293.39; \_\_utmz=35744766.1315504338.37.17.utmcsr=cs.odu.edu|utmccn=(referral)|utmcmd=referral|utmcct=/~mweigle/CS795-F11/Sched; author=mweigle; \_\_utma=256360345.1284928110.1285499037.1321824230.1322006984.250; \_\_utmz=256360345.1319416285.241.76.utmcsr=oducs-networking.blogspot.com|utmccn=(referral)|utmcmd=referral|utmcct=/
- Query String Parameters** (view decoded):
  - q: jo
  - sid: 0.48900638381019235
- Response Headers** (view parsed):
  - HTTP/1.1 200 OK
  - Date: Sun, 27 Nov 2011 19:15:55 GMT
  - Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod\_ssl/2.2.17 OpenSSL/0.9.8q
  - X-Powered-By: PHP/5.3.5
  - Content-Length: 19
  - Keep-Alive: timeout=5, max=97
  - Connection: Keep-Alive
  - Content-Type: text/html

43

# AJAX Suggest Example

## Data Trace

The screenshot shows a web browser displaying the **suggest.html** page. The form has "First Name: jo" and "Suggestions: John, Joe, Joseph". Below the page, the Network tab is open, showing the following resources:

- suggest.html (/~mweigle/cs312/ajax/sugg)
- xmlHelperFns.js (/~mweigle/cs312/xml)
- gethint.php (/~mweigle/cs312/ajax/sugg)
- gethint.php (/~mweigle/cs312/ajax/sugg)

The right pane shows the details for the selected resource, **suggest.html**. The Response tab is active, displaying the following information:

- 1 John, Joe, Joseph
- 2

Exactly the same as with q=j

44

# AJAX

## Questions

---

- ◆ When should XMLHttpRequest (or, with the helper functions, sendXHR) be used?
- ◆ In a webpage that uses AJAX, which parts are executed on the client and which on the server?
  - » use the type-ahead example
- ◆ How many total HTTP requests are generated in loading the type-ahead example and searching for the name 'john'?