07. AJAX Database Example.

7.1. AJAX Database

AJAX can be used for interactive communication with a database.

The following example will demonstrate how a web page can fetch information from a database with AJAX:

Example:

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
  border : 1px solid black;
  border-collapse: collapse;
}
th,td {
  padding: 5px;
}
</style>
<body>

<h2>The XMLHttpRequest Object</h2>

<form action="">
<select name="customers" onchange="showCustomer(this.value)">
<option value="">Select a customer:</option>
<option value="ALFKI">Alfreds Futterkiste</option>
<option value="NORTS ">North/South</option>
<option value="WOLZA">Wolski Zajazd</option>
</select>
</form>
<br>
<div id="txtHint">Customer info will be listed here...</div>

<script>
function showCustomer(str) {
  var xhttp;
  if (str == "") {
    document.getElementById("txtHint").innerHTML = "";
    return;
  }
  xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("txtHint").innerHTML =
    this.responseText;
        }
    };
    xhttp.open("GET", "getcustomer.php?q="+str, true);
    xhttp.send();
}
</script>

</body>
</html>
```

7.2. Example Explained - The showCustomer() Function

When a user selects a customer in the dropdown list above, a function called "showCustomer()" is executed. The function is triggered by the "onchange" event:

Example: showCustomer:

```
function showCustomer(str) {
  var xhttp;
  if (str == "") {
    document.getElementById("txtHint").innerHTML = "";
    return;
  }
  xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
    document.getElementById("txtHint").innerHTML =
  this.responseText;
    }
  };
  xhttp.open("GET", "getcustomer.php?q="+str, true);
  xhttp.send();
}
```

The *showCustomer*() function does the following:

- Check if a customer is selected
- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a file on the server
- Notice that a parameter (q) is added to the URL (with the content of the dropdown list)

7.3. The AJAX Server Page

The page on the server called by the JavaScript above is an PHP file called "getcustomer.php".

The server file could easily be rewritten in PHP, or some other server languages (ASP, JSP, CFM).

The source code in "getcustomer.php" runs a query against a database, and returns the result in an HTML table:

Example: PHP File

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}

table, td, th {
    border: 1px solid black;
    padding: 5px;
}

th {text-align: left;}
</style>
</head>
<body>

<?php
$q = intval($_GET['q']);

$con = mysqli_connect('localhost','peter','abc123','my_db');
if (!$con) {
    die('Could not connect: ' . mysqli_error($con));
}

mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = '".$q."'";
$result = mysqli_query($con,$sql);

echo "<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
<th>Hometown</th>
<th>Job</th>
```

```
        </tr>";
        while($row = mysqli_fetch_array($result)) {
            echo "<tr>";
            echo "<td>" . $row['FirstName'] . "</td>";
            echo "<td>" . $row['LastName'] . "</td>";
            echo "<td>" . $row['Age'] . "</td>";
            echo "<td>" . $row['Hometown'] . "</td>";
            echo "<td>" . $row['Job'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        mysqli_close($con);
        ?>
        </body>
        </html>
```

Explanation: When the query is sent from the JavaScript to the PHP file, the following happens:

1. PHP opens a connection to a MySQL server
2. The correct person is found
3. An HTML table is created, filled with data, and sent back to the "txtHint" placeholder