Lecture 27

AJAX and Persistence Too

Course logo spider web photograph from Morguefile openstock photograph by Gabor Karpati, Hungary.
Example 1 – Table Cells

- Define a table on server – SQLite
- Edit cells on client
- Update table whenever a cell changes
- Puts a lot of pieces together:
  - HTML, CSS, PHP,
  - SQL,
  - JavaScript, AJAX through jQuery
Changes Persistent

Fix the Dogs

<table>
<thead>
<tr>
<th>Name</th>
<th>Creator</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snoopy</td>
<td>Charles Schultz</td>
<td>Charlie Brown</td>
</tr>
<tr>
<td>Gromit</td>
<td>Nick Parks</td>
<td>Wallace</td>
</tr>
<tr>
<td>Lassie</td>
<td>Eric Knight</td>
<td>Timmie</td>
</tr>
</tbody>
</table>

Paragraph used to write update messages.
Paragraph used to write update messages.

Cell id dogs0003creator now has value Eric Knight

Return from AJAX
Data: UPDATE dogs SET creator="Eric Knight" WHERE dog_id = 3
Status: success
Pay Attention To ...

- PHP Code to create HTML for table
- Note a CSS id encodes table cell

```html
<table id="dogtable">
  <tr>
    <th>Name</th>
    <th>Creator</th>
    <th>Person</th>
  </tr>
  <tr>
    <td><input type="text" id="dogs0001name" value="Snoopy"></td>
    <td><input type="text" id="dogs0001creator" value="Charli"></td>
    <td><input type="text" id="dogs0001person" value="Charli"></td>
  </tr>
  <tr>
    <td><input type="text" id="dogs0002name" value="Gromit"></td>
  </tr>
</table>
```

- Probably better ways in complex systems, but this is clear.
Also, onchange call back

- There is only one call back function
- Two arguments
  - The id of the input text field
  - The value of the field after change.

```html
<input type="text" value="Snoopy" onchange="newval(this.id, this.value)">
<input type="text" value="Charles Schultz" onchange="newval(this.id, this.value)">
<input type="text" value="Charlie Brown" onchange="newval(this.id, this.value)">
<input type="text" value="Gromit" onchange="newval(this.id, this.value)">
<input type="text" value="Nick Parks" onchange="newval(this.id, this.value)">
<input type="text" value="Wallace" onchange="newval(this.id, this.value)"/>
```
Simpler using jQuery

- Post functionality of jQuery.
- Cell input field id (cid) sent to the URL responsible for making change.
- So too is the new value.
- Bit tricky – think about “cid:cid”

```javascript
function newval(cid, val) {
    $('#outp1').text('Cell id ' + cid + ' now has value ' + val);
    $.post('lec19ex04up.php', { cid:cid, val:val },
        function(data,status){
            $('#outp2').html('Return from AJAX <br/>
Data: ' + data + '<br/>
Status: ' + status);
        });
}
```
<?php

if (!isset($_REQUEST['cid'])) {
    die("Wow, that is not right.");
}

$cid = $_REQUEST['cid'];
if (isset($_REQUEST['val'])) {
    $val = strip_tags($_REQUEST['val']);
}

try {
    $dbh = new PDO("sqlite:/lec19ex04.db");
} catch(PDOException $e) {
    echo $e->getMessage();
}

$row = intval(substr($cid, 4, 4));
$fld = substr($cid, 8);
$cmd = "UPDATE dogs SET $fld=\"$val\" WHERE dog_id = $row;"
$dbh->exec($cmd);
echo $cmd;
?>
SQL Inject Demonstrated

- Starting condition; timed reset

Fix the Dogs

This database may be edited through the table below.

The data is entirely refreshed if more than 1 minute elapses since the last page view.

<table>
<thead>
<tr>
<th>Name</th>
<th>Creator</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snoopy</td>
<td>Schultz</td>
<td>Charlie Brown</td>
</tr>
<tr>
<td>Gromit</td>
<td>Parks</td>
<td>Wallace</td>
</tr>
<tr>
<td>Lassie</td>
<td>Knight</td>
<td>Timmie</td>
</tr>
</tbody>
</table>

Messages from AJAX based updates will appear here.

Last database Update: 04/24/2017 07:07:40
SQL Inject Demonstrated

- Enter a Weasel with a Twist
SQL Inject Demonstrated

- Refresh the Page

Fix the Dogs

This database may be edited through the table below.

The data is entirely refreshed if more than 1 minute elapses since the last page view.

<table>
<thead>
<tr>
<th>Name</th>
<th>Creator</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snoopy</td>
<td>Schultz</td>
<td>Weasel</td>
</tr>
<tr>
<td>Gromit</td>
<td>Parks</td>
<td>Weasel</td>
</tr>
<tr>
<td>Lassie</td>
<td>Knight</td>
<td>Weasel</td>
</tr>
</tbody>
</table>

Messages from AJAX based updates will appear here.

Last database Update: 04/24/2017 07:08:37
Think About This

➢ What happened?

```php
* There is a nice example of a vulnerability using this code.
* If a user enters exactly the following text into a field
  * cows" --
* Then the qualifier specifying the specific dog is ignored
* and all dogs have this field set.
*/
$dbh->exec ( "UPDATE dogs SET $fld="$val" WHERE dog_id = $row" );
$dbh->exec ( "UPDATE dogs SET uptime="$tme" WHERE dog_id = $row" );
echo "Last database Update: ". lastUpdatedHTML ()
?>
```

What Next – A Fix?

Use PHP prepare and execute.