JQuery Sortable Drag and Drop lists and a server side AJAX save

Posted At : 17 June 2011 07:19 | Posted By : Shaun McCran
Related Categories: JQuery, Javascript, RIA

I was recently asked this question about creating and using elements with the JQuery drag and drop functions:

```
say a grocery list) using JQuery and PHP to process and save the list. I found yo
```

This article explains how you can use a few JQuery interactions to create a sortable drag and drop list and save its state to a server side processing template.

Firstly you must include a series of JQuery script files. These are the core files and a collection of interaction manager files, things like mouse.js, sortable.js, draggable.js and droppable.js.

```
="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.

<s/cript src="jquery.ui.core.js"></script>
<s/cript src="jquery.ui.widget.js"></script>

<s/cript src="jquery.ui.mouse.js"></script>
<s/cript src="jquery.ui.draggable.js"></script>
<s/cript src="jquery.ui.droppable.js"></script>
<s/cript src="jquery.ui.sortable.js"></script>
```

Next create two div elements. The first of which holds our list of items. This is a html list that holds each item. I've also given each item in the list a unique id as this will be used as our item reference later on, to save back to the server.

The second div element is a sortable list. This list will hold our dragged items.

```
          <div class="shop-container">
                          <ul>
            <li class="draggable" id="bread">
  Bread <img src="bread.png" class="img-float">
                      </li>
            <li class="draggable" id="milk">
   Milk <img src="milk.jpg" class="img-float">
                      </li>
           <li class="draggable" id="cheese">
 Cheese <img src="cheese.gif" class="img-float">
                      </li>
           <li class="draggable" id="oranges">
Oranges <img src="orange.gif" class="img-float">
                      </li>
                          </ul>

                     </div>

    <ul id="sortable" class="sortable"></ul>
```

This will give us a list and an empty area with no interaction at all. To allow the list items to be dragged and the sortable area to accept and hold them we need to write a function that uses their selectors as identifiers for the JQuery interactions.

The code below has a selector of 'draggable' to enable the list items to be dragged around. It also uses the 'connectToSortable' value to pair it with the 'sortable' div element. Only by doing this do we allow the two div's to interact.

The first function uses the 'sortable' class to enabling sorting and dropping on the corresponding div. It also has an 'update' event that fires whenever its internal list is updated. This creates an Array of the list items and fires them through an AJAX request to a server side template where you can process them however you want.

For demo purposes I am also alerting() the list item values.

```
        <s/cript>
          $(function() {
$( "#sortable" ).sortable({ revert: true,

         update: function( event, ui ) {

var basketItems = $(this).sortable('toArray').toStrin
                   alert(basketItems);
  $.get('updatebasket.cfm', {basketItems:basketItems}

                        }

                   });

      $( ".draggable" ).draggable({
         connectToSortable: "#sortable",
                 helper: "clone",
                revert: "invalid"
                   });

    $( "ul, li" ).disableSelection();
               });
            </script>
```

It seems like a pretty flexible way of creating a shopping basket or sorting system but I've yet to find anywhere to use it in a production environment.

There is a demo of this working here: http://www.mccran.co.uk/examples/jquery-multi-drag-drop/