Combining JQuery Datatable with drag and drop functions

Posted At : 15 May 2011 01:10 | Posted By : Shaun McCran
Related Categories: JQuery, Javascript, AJAX

The datatables JQuery plugin (http://www.datatables.net/) is an incredibly powerful way of displaying tabular data and allowing the user to manipulate it without leaving the current view.

The plugin already features paging and sorting, but what if you want to extend the plugin with some other non default behaviours like drag and drop? Using a previous example (http://www.mccran.co.uk/index.cfm /2010/4/29/JQuery-Datatables-plugin-example-using-a-server-side-data-request-coldfusion) as a starting point I thought I'd try and integrate drag and drop with the datatable object.

There is an example of the finished JQuery datatable with added drag and drop here .

I won't go into how the datatable gets the data through an AJAX request as this is handled in the article above.

Firstly add the JQuery UI JavaScript file, as this will be handling the interactions.

```
t" language="javascript" src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.1
```

Next we will amend the existing datatable code and add in the block below. Using the 'sortable' UI function we can specify that each row will be a draggable object. (Any DOM elements specified as 'sortable' automatically inherit the 'draggable' functions properties.)

Using the 'start' and 'update' events we can calculate the starting and ending position of our dragged object, correcting the value for zero based

JavaScript Arrays. In the 'update' event I am also grabbing the table cell value as it is the record id in the example code. In this way I can pass this through to a remote CFC and amend the record in the database.

Lastly there is a simple ajax request to a data handler object, and a function to handle the response back from the object.

```javascript
var start;
var end;

$("#displayData tbody").sortable({
    cursor: "move",
    start:function(event, ui){
        // 0 based array, add one
        start = ui.item.prevAll().length + 1;
    },

    update: function(event, ui) {
        // 0 based array, add one
        end = ui.item.prevAll().length + 1;
        alert('Start: ' + start + ' End: ' + end);
        var id = ui.item.context.children[0].innerHTML;
        alert(id);

        $.getJSON('dao.cfc', {
            method:'methodName',
            returnFormat:'JSON',
            name:value
        },

        // handle the response
        function(data){
            if(data.intSuccess == 1){
                // success
            } else {
                // error
            }
        });
    }
    // end of drag
});
```

In the following example I have also created two areas on the page that will accept a dragged object. One is labelled 'duplicate' the other 'delete', these have events attached to them so as to demonstrate that you could easily add graphical tools to duplicate and delete record through

dragging them into these elements. Just imagine the 'delete' box as a bin.

[Example JQuery datatable with added drag and drop](#) .