

## JQuery datatable with added FancyBox and auto refresh

Posted At : 19 May 2011 22:11 | Posted By : Shaun McCran  
Related Categories: JQuery, Coldfusion, Javascript, RIA

Recently I've had a few questions about how to use the datatables plugin to display data, and also integrate a pop up or lightbox function. The pop up would have an edit form in it, and when the user completes the edit the light box goes away and the datatable refreshes.

This is relatively straightforward to do, but there are several key concepts to getting everything working correctly, so I've broken it down into a few chunks.

You can see an example of a [datatable with added FancyBox here](#) .

Firstly we cannot use the standard method of calling a datatable. We need to load it into a JavaScript function rather than just using the document.ready() method. This is because we need to check if it exists, and we need to be able to refresh it in code, and we can't do that unless it has a reference.

Below is the code to create the datatable as a JavaScript function. Notice that what it does is check if a reference exists 'dTable' and if it doesn't it runs the datatable code. If it does exist then it runs the dTable.fnDraw() method instead, which causes the datatable to refresh itself. All the other datatable code remains the same.

```
</script type="text/javascript" charset="utf-8">
function getData()
{
    if (typeof dTable == 'undefined') {

        dTable = $('#displayData').dataTable( {
            // All the rest of the datatable code goes here
        });
    }
    else
    {
        dTable.fnDraw();
    }
}
```

```
}  
</script>
```

Next I want to change the JSON from the remote cfc. Instead of passing the id value back in the JSON response I want to change it into a link for my pop up event. In the loop that creates the JSON I find the id value, and switch it out for a hyperlink, and give it a class of pop-up. I am also adding the id in as the element id, so that I can pick it up and pass it to our pop up window.

Be careful with your quoted values at this point, quotes may interfere with your JSON.

```
<cfif variables.index EQ 'id'>  
<a class=pop-up id=#rResult[variables.i][rResult.currentRow]#>Edit</a>  
<cfelse>  
#rResult[variables.i][rResult.currentRow]#  
</cfif>
```

So now the datatable will have a link in it as the first element. The next part of this puzzle is attaching a JQuery selector to the 'pop-up' class on our link. As the links are loaded in dynamically we cannot just give them a regular selector. We need to bind the event using the live() JQuery event method. This is a really powerful JQuery method, I won't go into depth about what it does here, but Ben Nadel has written a great entry about it here: <http://www.bennadel.com/blog/1751-jQuery-Live-Method-And-Event-Bubbling.htm> , I'd advise reading it and making sure you are clear on what its doing.

```
<s/crript type="text/javascript">  
$(document).ready(function(){  
  
    // pop up handler
```

```

$( ".pop-up" ).live(
    "click",
    function( event ){

        // get the id of the item clicked on
        var id = $(this).attr('id');

        $.fancybox({
            'overlayOpacity'      : 0.5,
            'href'                : 'blank.cfm?id=' + id,
            'onClosed'            : function() {
                getData();
                alert('data table reloaded');
            }
        });

        // block the href actually working
        return( false );

    }

);

// load the datatable
getData();
});
</script>

```

You can see above we are looking out for a click on any elements that exist, or are created with a class of 'pop-up'. Then we attach the fancybox plugin to them. I'm also reading the id value and using it in the href url call. In this way we can pass in the id of the record clicked on.

The fancybox plugin has a 'onClosed' option, which allows you to specify code to run when the lightbox closes through any method. in the example above I am running the getData() function, which as the datatable already exists will simply reload it.

The very last line is where you tell the page to load the datatable, as it is no longer within a document.ready() it will not auto load.

You can see an example of a [datatable with added FancyBox here](#) .