Connecting Select form fields based on data selections Pt 3

Posted At : 20 August 2009 10:10 | Posted By : Shaun McCran
Related Categories: Coldfusion, Javascript, RIA, Json, AJAX

I've been exploring various methods of using JQuery to populate Select form fields. In the previous article I used an intermediary file to act as a handler for the JSON returned object. In this last article I have removed the extra handler template, interfacing directly with the CFC.

Firstly there are some small changes to the JQuery url call.

```
<scr/ipt src="jquery-1.2.3.js" type="text/javascript"></script>
<scr/ipt src="select-chain.js" type="text/javascript" charset="utf-8"></script>
<scr/ipt type="text/javascript">
jQuery.noConflict();
jQuery(function () {
var type = jQuery('#series');
var sp = jQuery('#issueno');
var selectedSeries = jQuery('#series').val();

type.selectChain({
target: sp,
url: 'jquerySeries.cfc?method=getIssueNos',
type: 'post',
data: { ajax: true, returnformat: 'plain', series: selectedSeries }
}).trigger('change');

});
</script>
```

Now we are directly referencing the CFC. Simply append the function name as a url parameter. In the "DATA" element we place the data we are passing in the form of key value pairs. This will build all the values into the JQuery url, so the final URL would be:

```
ue rySeries.cfc?method=getIssueNos&ajax=true&returnformat=plain&series=selectedSeri
```

At this point I was getting an parse error, until I found the 'returnFormat' value. Coldfusion will return WDDX encoded packets by default, problem was my handler was looking for Json. Adding this value will stop this.

Next we have the same form as before, there are no changes at all: (here for completeness really)

```
<cfset variables.series = createObject("component","jquerySeries").getSeries()>
<form action="" method="post">
<select name="series" id="series">
<cfoutput query="variables.series">
<option value="#variables.series.intId#">#variables.series.varName#</optio
</cfoutput>
</select>
<select name="issueno" id="issueno">
<option></option>
</select>
<button type="submit">Submit</button>
</form>
```

Lastly there are one or two small changes to our CFC function.

```
name="getIssueNos" access="remote" output="false" hint="returns related series iss
<cfargument name="series" type="numeric" required="false" hint="Id of the pul
<cfset var result = "">

<cfquery name="result" datasource="#application.ds#">
SELECT [intIssueNo]
FROM [dbo].[tbl_cn_series_issueNos]
where intSeriesId = <cfqueryparam value="#arguments.series#" cfsqltype="c
</cfquery>

<cfsetting showdebugoutput="false">
<cfset ojson = createObject("component","cfjson")>
<cfset theresults = ojson.encode(listToArray(valuelist(result.intIssueN
<cfsetting enablecfoutputonly="true">
<cfreturn theresults>
</cffunction>
```

The access method has changed, so we add an "access=remote" value to expose the function as a service. Then we serialize the query result into JSON, and return it. This seems like a much more succinct way of doing this.