# Connecting Select form fields based on data selections Pt 2

Posted At : 18 August 2009 10:15 | Posted By : Shaun McCran
Related Categories: Coldfusion, Javascript, RIA, Json, AJAX

I want to be able to dynamically change a second select field based on the value of the first select field.

Following on from the cfajaxproxy example I wrote last week, where I discovered I could not use ColdFusion 8 functionality on my live server, I have arrived at a variant solution.

In this example I am using a similar JQuery url request to process the output from a CFC. The first template is the form itself. This includes references to the JQuery libraries and the binding of the response to the form elements. It also builds the url request to the 'request_processor. cfm' file, which handles the CFC.

```
       <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');

                                 type.selectChain({
                                    target: sp,
                           url: 'request_processor.cfm',
                                    type: 'post',
                               data: { ajax: true }
                               }).trigger('change');

                                       });
                                  </script>
```

Next build a simple form, and populate the first select field with data from a method.

```
                        <cfif not structisempty(form)>
                              <cfdump var="#form#">
                                    <cfelse>
<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#</option>
                                       </cfoutput>
                                      </select>
                             <select name="issueno" id="issueno">
                                       <option></option>
                                         </select>
                            <button type="submit">Submit</button>
                                        </form>
                                      </cfif>
```

Next is the intermediately file that handles the data manipulation, this is triggered when the user chooses an option from the first select field. It passes through the selected value, and returns the query object. This is then serialised using the cfJson object, to return the data in Json. If you do anything like this remember to watch your debug output, it was destroying my Json response for a good ten minutes before I remembered to turn it off. Doh!

```
                        <cfsetting showdebugoutput="false">
                       <cfsetting enablecfoutputonly="true">
                      <cfparam name="form.series" default="8">
ıriables.issues = createObject("component","jquerySeries").getIssueNos(series=form
                   <cfset ojson = createObject("component","cfjson")>
fset theresults = ojson.encode(listToArray(valuelist(variables.issues.intIssueNo))
                        <cfoutput>#theresults#</cfoutput>
```

Finally a CFC that performs the database functions. This is a pretty straight forward CFC that performs two database queries, the second based on an id passed in from the first.

```
```

```
<cfcomponent>

    <cffunction name="getSeries" output="false" hint="Returns publication series
        <cfset var result = "">

        <cfquery name="result" datasource="#application.ds#">
            SELECT [intId],[varName]
              FROM [dbo].[table]
             Where intActive = '1'
               Order by varName
             </cfquery>

        <cfreturn result>
          </cffunction>

<cffunction name="getIssueNos" output="false" hint="returns related series issue n
    <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].[table]
      where intSeriesId = <cfqueryparam value="#arguments.series#" cfsqltype="c
             </cfquery>

        <cfreturn result>
          </cffunction>
</cfcomponent>
```

Once you have these elements hooked up you'll see that the response from the first select field changes the values in the second field. You can download a rar'd version of the code base here .

This works well, but I'm not massively happy about the 'remote_processor' file. I think I'll see if there is a way of directly calling the CFC, and moving the JSON serialisation into the functions.