

Listening for CFTREE data loaded events

Posted At : May 28, 2010 6:50 PM | Posted By : Nathan Mische

Related Categories: JavaScript, ColdFusion

Last week Ray Camden asked if it was possible to listen for data loaded events with CFTREE. I had done some work with CFTREE in the past and I knew something like this should be possible. After some quick investigation I put together a quick and dirty example to show how you could be notified when new nodes are loaded for a tree.

```
<html>
<head>
  <title>Example: ColdFusion.Tree.loadNodes</title>
  <script language="JavaScript">
    ColdFusion.Tree.loadNodes = (function (old) {
      return function (nodesArray, params) {
        if (typeof old == 'function') old.apply(this,arguments);
        myCallBack();
      };
    }) (ColdFusion.Tree.loadNodes);

    function myCallBack() {
      //console.log("hello");
      alert("hello");
    }
  </script>
</head>
<body>
  <h1>Example: Calling a method after nodes load</h1>
  <cform name="testform">
    <cftree name="t1" format="html">
      <cftreeitem bind="cfc:makeTree.getNodes({cftreeitemvalue},{cftreeitempath})"
    </cftree>
  </cform>
</body>
</html>
```

Ray pointed out that this would only work if you had one tree on a page because all trees would fire the same callback function. With a little bit more work I put together the following, which allows you to add listeners to all trees or target listeners to specific trees.

```
<html>
<head>
  <title>Example: ColdFusion.Tree.loadNodes</title>
  <script language="JavaScript">
    // add an object to hold our listeners
    ColdFusion.Tree.loadNodesListeners = {};

    // add a convenience method to add a listener
    ColdFusion.Tree.addLoadNodesListener = function(listener,treeid) {
```

```

var key = treeid ? treeid":_all_";
if (!this.loadNodesListeners[key])
    this.loadNodesListeners[key] = [];
this.loadNodesListeners[key].push(listener);
};

// override loadNodes to call our listeners;
ColdFusion.Tree.loadNodes = (function (old) {
    return function (nodesArray, params) {
        if (typeof old == 'function') old.apply(this,arguments);
        //call listeners not assigned to a specific treeid
        if (ColdFusion.Tree.loadNodesListeners._all_) {
            for (var i = 0; i < ColdFusion.Tree.loadNodesListeners._all_.length; ++i) {
                var listener = ColdFusion.Tree.loadNodesListeners._all_[i];
                listener.apply(this, arguments);
            }
        }
        //get treeid and call treeid specific listeners
        var treeid = params.treeid;
        if (ColdFusion.Tree.loadNodesListeners[treeid]) {
            for (var i = 0; i < ColdFusion.Tree.loadNodesListeners[treeid].length; ++i) {
                var listener = ColdFusion.Tree.loadNodesListeners[treeid][i];
                listener.apply(this,arguments);
            }
        }
    };
})(ColdFusion.Tree.loadNodes);

```

```

function treeListener(){
    //console.log"tree listener";
    alert"tree listener";
}

```

```

function treeOneListener() {
    //console.log"tree one listener";
    alert"tree one listener";
}

```

```

function treeTwoListener() {
    //console.log"tree two listener";
    alert"tree two listener";
}

```

```
function init() {
    ColdFusion.Tree.addLoadNodesListener(treeListener);
    ColdFusion.Tree.addLoadNodesListener(treeOneListener);
    ColdFusion.Tree.addLoadNodesListener(treeTwoListener);
}
```

```
ColdFusion.Event.registerOnLoad(init);
```

```
</script>
</head>
<body>
  <h1>Example: Calling a method after nodes loaded</h1>
  <h2>Tree 1</h2>
  <cfform name="testform">
    <cftree name="t1" format="html">
      <cftreeitem bind="cfc:makeTree.getNodes({cftreeitemvalue},{cftreeitempath})" />
    </cftree>
  </cfform>
  <h2>Tree 2</h2>
  <cfform name="testform">
    <cftree name="t2" format="html">
      <cftreeitem bind="cfc:makeTree.getNodes({cftreeitemvalue},{cftreeitempath})" />
    </cftree>
  </cfform>
</body>
</html>
```

Both of these examples call the following CFC to load dummy data. The CFC has a Sleep() call to delay the response so you can see the callbacks only fire after we get results back from the server.

```
<cfcomponent>
  <cffunction name="getNodes" returnType="array" output="no" access="remote">
    <cfargument name="nodeitemid" required="true" />
    <cfargument name="nodeitempath" required="true" />

    <cfset var nodeArray = ArrayNew() />
    <cfset var element = StructNew() />
    <cfset var i = "" />

    <!--- the initial value of the top level is the empty string --->
    <cfif nodeitemid IS "">
```

```

        <cfset nodeitemid =0>
    </cfif>

<!--- sleep so we can see the server working --->
<cfset Sleep(1000) />

<!--- create a array with elements defining the child nodes --->
<cfloop from="1" to="#RandRange(1,4)#" index="i">
    <cfset StructClear(element) />
    <cfset element.value = "#nodeitemid#.#i#" />
    <cfset element.display = "Node #element.value#" />
    <cfset element.expand = "false" />
    <cfset element.href = "index.cfm" />
    <cfset element.leafnode = "false" />
    <cfset element.target = "_blank" />
    <cfset nodeArray[i] = Duplicate(element) />
</cfloop>
<cfreturn nodeArray />
</cffunction>
</cfcomponent>

```

Hopefully this will help if you need to do something with a CFTREE after data is loaded from the server.