Utility Functions: REFindAll() and REFindAllNoCase()

Posted At : September 25, 2009 12:29 PM | Posted By : Jon Hartmann

Related Categories: Utility Function

Its been a while since I released a utility function, so today I bring you a very closely related pair that solve a very simple problem thats missing with ColdFusion's Regular Expression functions. Given a regular expression in ColdFusion, you can find out if there is a match for that regular expression in a given string, and where it is, but you can't find out the location of ALL of the values that match the regular expression.

So now I've created REFindAll() and REFindAllNoCase() to handle that task. They each return an array of structures containing the pos and len values, just like the array you get from calling REFind() and REFindNoCase(), but unlike those functions, this gets you all occurrences in a single array.

```
<cffunction name="REFindAll" output="false" returntype="array">
    <cfargument name="pattern" type="string" required="true" />
    <cfargument name="string" type="string" required="true" />

    <cfset var local = StructNew() />

    <cfset local.result = ArrayNew(1) />

    <cfset local.end = Len(arguments.string) />
    <cfset local.current = 1 />

    <cfloop condition="local.current lt local.end">
        <cfset local.match = REFind(arguments.pattern, arguments.string, local.current, "true") />

        <cfif local.match.pos[1] gt 0>
            <cfset local.temp = StructNew() />
            <cfset local.temp.pos = local.match.pos[1] />
            <cfset local.temp.len = local.match.len[1] />

            <cfset ArrayAppend(local.result, local.temp) />
            <cfset local.current = local.match.pos[1] + local.match.len[1] />
        <cfelse>
            <cfset local.current = local.end />
        </cfif>
    </cfloop>

    <cfreturn local.result />
</cffunction>

<cffunction name="REFindAllNoCase" output="false" returntype="array">
    <cfargument name="pattern" type="string" required="true" />
    <cfargument name="string" type="string" required="true" />

    <cfset var local = StructNew() />

    <cfset local.result = ArrayNew(1) />

    <cfset local.end = Len(arguments.string) />
    <cfset local.current = 1 />

    <cfloop condition="local.current lt local.end">
        <cfset local.match = REFindNoCase(arguments.pattern, arguments.string, local.current, "true") />

        <cfif local.match.pos[1] gt 0>
            <cfset local.temp = StructNew() />
            <cfset local.temp.pos = local.match.pos[1] />
            <cfset local.temp.len = local.match.len[1] />
```

```
            <cfset ArrayAppend(local.result, local.temp) />
            <cfset local.current = local.match.pos[1] + local.match.len[1] />
        <cfelse>
            <cfset local.current = local.end />
        </cfif>
    </cfloop>

    <cfreturn local.result />
</cffunction>
```

Here is the example:

```
<cfset test = "asdf asdf qwer rety vhfgh cxv" />
<cfdump var="#REFindAllNoCase("[a-z]+", test)#">
```

Which produces:

array

1	**struct**	
	LEN	4
	POS	1
2	**struct**	
	LEN	4
	POS	6
3	**struct**	
	LEN	4
	POS	11
4	**struct**	
	LEN	4
	POS	16
5	**struct**	
	LEN	5
	POS	21
6	**struct**	
	LEN	3
	POS	27

Check back tomorrow for the next natural step with this function.