

Utility Functions: REExtractAll() and REExtractAllNoCase()

Posted At : September 25, 2009 12:40 PM | Posted By : Jon Hartmann

Related Categories: Utility Function

Since it had been so long since I released utility functions, I'm doing another double header today! Not that its really that hard since these two are exactly the same... but any way, continuing where yesterday's REFindAll() and REFindAllNoCase() left off, these two new functions actually return the matching values!

```
<cffunction name="REExtractAll" 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) />

  <!-- Get an array of all tag positions -->
  <cfset local.positions = REFindAll(arguments.pattern, arguments.string) />

  <!-- Loop over positions and get tags -->
  <cfloop from="1" to="#ArrayLen(local.positions)#" index="local.x">
    <cfset local.temp = local.positions[local.x] />
    <cfset ArrayAppend(local.result, Mid(arguments.string, local.temp.pos, local.temp.len)) />
  </cfloop>

  <cfreturn local.result />
</cffunction>

<cffunction name="REExtractAllNoCase" 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) />

  <!-- Get an array of all tag positions -->
  <cfset local.positions = REFindAllNoCase(arguments.pattern, arguments.string) />

  <!-- Loop over positions and get tags -->
  <cfloop from="1" to="#ArrayLen(local.positions)#" index="local.x">
    <cfset local.temp = local.positions[local.x] />
    <cfset ArrayAppend(local.result, Mid(arguments.string, local.temp.pos, local.temp.len)) />
  </cfloop>

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

And the example is:

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

Which nets you:

array	
1	asdf
2	asdf

3	qwer
4	rety
5	vhfgh
6	cxv

Home you enjoyed these functions; I've got a couple more based on these that are coming soon.

This function is close to my older PatternExtract() function, but its a little easier to use.

Update: As Adam Cameron rightly pointed out... this has already been done in CF8... *sigh* go use REMatch() and REMatchNoCase(), and leave me to my shame :)