

# Utility Functions: Java Regex Pattern Matching

Posted At : December 12, 2008 5:29 AM | Posted By : Jon Hartmann

Related Categories: Utility Function

Ok, so this time I have a trio of functions all to get at a key piece of Java regular expression matching: groups. Groups allow you to extract small snippets of a regular expression, and let you do a lot more with them then ColdFusion alone. Check out the whole bunch.

First up is a basic function to create a pattern. Patterns are reusable, and you can get instances of the Matcher class for a pattern as methods of the pattern itself, so a shortcut to the "constructor" makes sense.

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

    <cfreturn CreateObject("java", "java.util.regex.Pattern").compile(arguments.pattern) />
</cffunction>
```

Next up is a simple function to use a pattern against a string to decide if its a match. It does let you use either an existing pattern or to give it a new pattern string.

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

    <cfset var local = StructNew() />

    <cfif IsSimpleValue(arguments.pattern)>
        <cfset arguments.pattern = CreateObject("java", "java.util.regex.Pattern").compile(arguments.pattern) />
    </cfif>

    <cfset local.matcher = arguments.pattern.matcher(arguments.string) >

    <cfreturn local.matcher.matches() />
</cffunction>
```

Now the good stuff, this function actually extracts any groups that are in your pattern from a string, and returns them as an array.

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

    <cfset var returnValue = ArrayNew(1) />

    <cfif IsSimpleValue(arguments.pattern)>
        <cfset arguments.pattern = CreateObject("java", "java.util.regex.Pattern").compile(arguments.pattern) />
    </cfif>

    <cfset local.matcher = arguments.pattern.matcher(arguments.string)>

    <cfif local.matcher.matches()>
        <cfset local.groups = local.matcher.groupCount() />
        <cfloop from="1" to="#local.groups#" index="x">
            <cfset returnValue[x] = local.matcher.group(x) />
        </cfloop>
    </cfif>
</cffunction>
```

```
<cfreturn returnValue />  
</cffunction>
```