Using Multi-Word-Actions in ColdBox 2.6

Posted At : November 11, 2008 5:06PM | Posted By : Matt Quackenbush
Related Categories: ColdBox, ColdFusion

The other day Jason Durham posed a question to me:

> How can I go about using multi-word actions, using hyphens as separators, in my ColdBox handlers?

After my initial "Why on earth would you want to do that?" response, Jason pointed out the SEO benefits of doing so, which makes perfect sense. (That's when I thought to myself, "Hmmm, why didn't *I* think of that?")

A ColdBox URL looks like this:

Non-SES URL

```
http: //www.mydomain.com/index.cfm?event=handler.action
```

SES URL

```
http: //www.mydomain.com/index.cfm/handler/action
```

Typically speaking, when the action is multiple words, say 'list products', one would simply camel-case it, and it would become 'listProducts'. Jason's idea was to separate the words with a hyphen (-), giving search engines something more meaningful to index. So 'list products' would become 'list-products'. ColdFusion does not allow hyphens in method names, but thankfully, ColdBox allows for a ridiculously simple solution to the problem.

onMissingAction()

New in ColdBox 2.6 is onMissingAction() . Much like onMissingMethod() in CF8, onMissingAction() is an action that you can place into your handlers that will fire when an invalid action is requested. So let's take a look at how to use onMissingAction() to implement our need for multi-word actions.

```
<cffunction name= "onMissingAction" hint= "I am fired when an action is called on me that doesn't exist"
returntype= "void" output= "no" access= "public" >
<cfargument name= "Event" hint= "The event object" required= "yes" type= "coldbox.system.beans.
requestContext" />
<cfargument name= "missingAction" hint= "The name of the requested action" required= " true " type= "string"
/>
<cfscript>
var theRealAction = replace (missingAction, "-" , "" , "all" );
if ( structKeyExists(this, theRealAction) ) {
runEvent( event.getCurrentHandler() & "." & theRealAction );
} else {
runEvent( replace (getSetting( "onInvalidEvent" ), "/" , "." , "all" ) );
}
</cfscript>
</cffunction>
```

So what does this little snippet do? It's pretty straight forward:

1. The invalid action is captured by onMissingAction() [Line 1]
2. We create a variable named 'theRealAction' by stripping all hyphens from the captured invalid action [Line 6]
3. We check to see if the handler has a method named the same as our 'theRealAction' variable [Line 8]
    1. If yes, we run the event [Line 9]
    2. If no, we grab the onInvalidEvent setting and run that event [Line 11]

One note to make about the onInvalidEvent setting. If you are using SES URLs in ColdBox, this setting needs to use a slash (/) instead of a dot (.). Knowing this, in Step 3.2 we're replacing slashes with dots, since the runEvent() method requires the dot.

With our onMissingAction() action now in place, when the browser requests our list-products URL...

```
1. http: //www.mydomain.com/index.cfm/catalog/list-products
```

... the catalog.listProducts() event is the one that ultimately fires.

Happy SEO with your ColdBox-powered SES URLs. :-)