

Building Intelligent sessions into your framework

Posted At : 20 May 2010 11:52 | Posted By : Shaun McCran
Related Categories: Development, Coldfusion, Best practices

Have you ever been logged into an application and had your session timeout, then when you log back in you are returned to a different place altogether?

This article deals with a way to mark where a user was in your application and return them to that location when they log back in. It also allows a user to deep link into an application. IE when they have a bookmarked link instead of being passed to the home page, they are passed through to their original destination.

This relies quite a lot on knowing the processing order of your application well. IE what order your templates are run in and more specifically what order any functions are run in. You could do this effectively in your 'onRequestStart' function, but in this example I am using a fuseBox framework.

In my controller CFC's I have a preFuseAction. This function runs before any of the other controller functions, making it an ideal place to put any granular security or user handling functions.

```
<cffunction name="prefuseaction">
    <cfargument name="myFusebox" />
    <cfargument name="event" />

<cfset cookie.requestedTemplate = 'index.cfm?' & cgi.query_string>

    <!--- check that user is logged in --->
    <cfif NOT isdefined('session.loggedIn')>
    <!--- kick the user to somewhere else --->
    </cfif>
```

In the code above before I do anything I store the requested URL as a cookie value. This happens before I check for security. In this way it is stored whether a user is delivered to the requested template, or to the login screen. Then I run a check on a session value to see if I should continue onwards.

In this way we always have a cookie value of the last page a user requested. Whether they got there or not.

Next we need to use the cookie value in our login script.

```
                <cfif variables.login>
<cfif isDefined('cookie.requestedTemplate') and len(cookie.requestedTemplate)>
                <cflocation url="#cookie.requestedTemplate#" addtoken="false">
                    </cfif>

                <cflocation url="#myself#" addtoken="false">
                    </cfif>
```

In the code here we are checking that we have successfully logged in, and if we have we check if the cookie value exists, and if there is a value there.

If there is a value we pass the user to the last location they requested prior to login, otherwise we just pass them to the logged in home page.

This works for deep linking as well because the preFuseAction function will run when a user uses a bookmark, so the bookmark location will be stored as a cookie value.