Application.cfc Reference

by Michael Dinowitz

This is based on Raymond Camden's Application.cfc reference

http://ray.camdenfamily.com/downloads/app.pdf

```
<cfcomponent output=”false”>
<!--- Application/Session management variables --->
  <cfset this.name = “ApplicationName”>
  <cfset this.applicationTimeout = createTimeSpan(0,2,0,0)>
  <cfset this.clientManagement = true>
  <cfset this.clientStorage = “registry”>
  <cfset this.loginStorage = “session”>
  <cfset this.sessionManagement = true>
  <cfset this.sessionTimeout = createTimeSpan(0,0,20,0)>
  <cfset this.setClientCookies = true>
  <cfset this.setDomainCookies = false>
    <cfset this.scriptProtect = false>
<!--- Memory variable start methods --->
  <cffunction name=”onApplicationStart” output=”false”>
      <cfreturn true>
  </cffunction>
  <cffunction name=”onSessionStart” returnType=”void” output=”false”>
    </cffunction>
  <!--- Page processing methods --->
    <cffunction name=”onRequestStart” output=”false”>
      <cfargument name=”thePage” required=”true”>
      <cfreturn true>
    </cffunction>
    <cffunction name=”onRequest” returnType=”void”>
      <cfargument name=”thePage” required=”true”>
      <cfinclude template=”#arguments.thePage#”>
    </cffunction>
    <cffunction name=”onRequestEnd” returnType=”void” output=”false”>
    <cfargument name=”thePage” required=”true”>
  </cffunction>
<!--- Error handling --->
  <cffunction name=”onError” returnType=”void” output=”false”>
      <cfargument name=”exception” required=”true”>
      <cfargument name=”eventname” required=”true”>
  </cffunction>
<!--- memory variable timeout methods --->
    <cffunction name=”onSessionEnd” returnType=”void” output=”false”>
      <cfargument name=”sessionScope” required=”true”>
      <cfargument name=”appScope” required=”false”>
  </cffunction>
  <cffunction name=”onApplicationEnd” returnType=”void” output=”false”>
      <cfargument name=”applicationScope” required=”true”>
    </cffunction>
</cfcomponent>
```

If no name is specified, the ***Application*** scope will contain a reference to the 'server object' which contains all applications as well as information about the server (not the *Server* scope).

When the onApplicationStart method is first run, it is single-threaded. This ensures that it will only run once, but if the code within the method takes a long time or hangs, it will cause the entire application to hang until the method finishes or times out.

The onRequest method overrides the requested page and allows an alternate to be loaded in its place.

If the onRequest and onError methods are both defined and the requested template uses a CFABORT tag, onError will treat it as an error.

The onError() method takes precedence over every other error handler other than CFTRY/ CFCATCH. If you want the onError method to output data, the output attribute has to be changed.

This is a reference to the current application scope. Changes made to this argument will be reflected in the current application scope.

PROFESSIONAL COLDFUSION CONTENT FOR THE COLDFUSION PROFESSIONAL

The Fusion Authority

Quarterly Update

Summer 2006

Why You Should Use Application.cfc
by the CF Jedi Master, Raymond Camden

The Why, When and How of Flash Forms
Matt Woodward's Cool Tricks

An Honest Look at Integrated Reporting
Kay Smoljak Investigates

Let Asynchronous Processing Save You Time and Money
Doug Boude's Experience

What's Hot? What's Not?
What's on the Guru's minds?
by Ben Forta ⁎ Michael Dinowitz ⁎ Hal Helms
Sean Corfield ⁎ Raymond Camden ⁎ Jared Rypka-Hauer

and more...

ColdFusion MX 7 Features you need to know

SUMMER 2006
ISSN 1932-0264
9 771932 026000
00
62>