CF Application configuration

New Zealand CFUG

Kai Koenig, 05/07/2007

Ventego creative

Agenda

- Fundamentals
- Loading and managing config data
  - .ini
  - XML
  - DI/IoC
- Concerns

Ventego creative

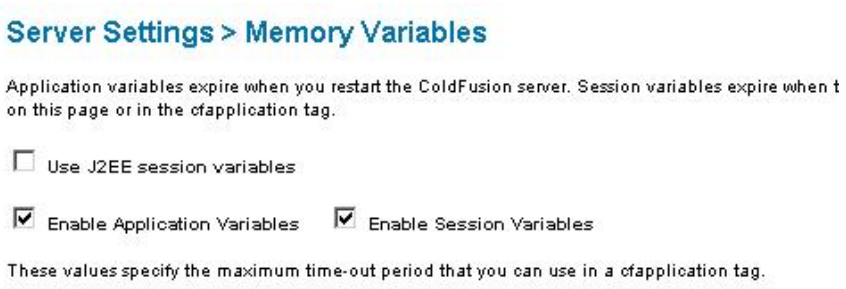
Fundamentals: Application scope

- The application scope is one of the so called “shared” scopes, i.e. variables in this scope are available to any CF page of an application.
- Often used to store application-wide configuration.
- Syntax:
  - <cfset application.datasource = “CRMdb”>
  - application.datasource = “CRMdb”;

Ventego creative

Fundamentals: Application scope

- Application variables have to be enabled in the ColdFusion administrator



Ventego creative

Fundamentals: <cfapplication>

- 2 slides ago I mentioned: “…to any CF page of an application.” – what does that mean?
- An application is defined by a <cfapplication> tag and its name attribute. All .cfm/.cfc files within its context are part of the application and share application scope variables.
- IMPORTANT: Provide the name attribute, otherwise you’re dealing with the “anonymous” application scope – chaos and mayhem might be the result ☺

Ventego creative

Fundamentals: <cfapplication>

- Usually placed in the Application.cfm or Application.cfc
- Pros and Cons of each have been discussed in a meeting a while ago
- BTW: application. is a scope – and as such a structure in CF.

Example 1

- Some simple experiments with application scope variables, the CF Administrator and CFDUMP ☺

Ventego creative

Application variables

- Difference between application variables and application scope variables:
  - Application scope variables are in a shrared scope
  - Application variables are local scope variables created in the Application.cfm (that's not what we're talking about here)

Common use cases

- A very common use case of the application scope is to store configuration data in there, i.e. data source names etc
- Another potential use:
  - Global variables
- The application scope can contain any variable type:
  - plain variables, structs, arrays, CFCs etc.

Ventego creative

A bit of theory: Singleton

- Singleton is a OO pattern
- Purpose: to ensure that exactly one instance of a class can ever exist in a system
- Basically – the application scope could play some sort of a Singleton role in the non-OO CF world
- Singletons are just sophisticated global variables anyway ☺

Ventego creative

Singletons in Java

```
public class Singleton
{
	// Private constructor suppresses generation of a (public) default
	    constructor
	private Singleton() {}

	private static class SingletonHolder
	{
	  private final static Singleton INSTANCE = new Singleton();
	}

	public static Singleton getInstance()
	{
	  return SingletonHolder.INSTANCE;
	}
}
```

How to store/load config data

- Windows-style .ini files
- XML
- Dependency injection (ColdSpring – no, we don’t cover that ☺)

Ventego creative

Windows-style .ini files

- Easy to use in CF:
  - getProfileSections
  - getProfileString

Ventego creative

Example 2

- Load date from an .ini file and store it in the application scope

XML

- XML handling in CF is reasonably well supported:
  - CFXML
  - XML function library
- But why to reinvent the wheel, let's just use configCFC (http://code.google.com/p/configcfc/)

Ventego creative

Example 3

- Working with configCFC and using XML files for configuration data
- Did you notice? This example also solved the issue of updating the config data and avoids to reload them all the time.

Ventego creative

Some general thoughts

- Configuration data in different environments… dev, test, staging, production
  - CVS/SVN is your friend here, also look into using ANT build scripts to automatically create and deploy config files for different environments during deployments
- Locking – CFLOCK is your friend when it comes to race conditions related to application scope variables!

Ventego creative

Dependency injection

- This is more about configuration in a wider and more abstract sense
- Dependency injection does NOT inject dependencies, it's a way to avoid dependencies and make your software systems (in OO-terms) more flexible
- In the CF world: ColdSpring is the #1 framework that supports DI and Inversion of Control (to kick in another buzzword ☺)

Ventego creative

Contact

Kai Koenig
Director, Software Solutions Architect
Ventego Creative Ltd, Wellington

email:
kai@ventego-creative.co.nz

mobile:
+64 (0)21 928 365

blog:
http://www.bloginblack.de

Ventego creative