

Using Isapi / Apache rewriting to mask URL strings, for cosmetics and security

Posted At : 22 December 2009 22:44 | Posted By : Shaun McCran

Related Categories: Software Architecture, Security, Development

One of the more recent additions to my Coldfusion frameworks is masking the more ugly URL's using Isapi rewrite. In this article I'll be using Helicon's Isapi ReWrite, but Apache re write works in much the same way.

Usually in your Coldfusion frameworks, most other technologies as well, you are passing around a variable or two to control the page content, and more often than not it is in the url. It never looks particularly clean if your URL has a long name value query string behind it, like this:

```
http://www.mysite.com/index.cfm?variable1=pageName&location=england&value=7
```

Cosmetic reasons

So for two reasons URL rewriting seems like a good idea.

Firstly to mask those ugly URLs with a url rewriter. On a basic level this will re write specified request to the URL you tell it to, taking your ugly list of name value pairs and changing it into a user friend URL. If you are pitching this to a client this looks a lot more professional.

Security reasons

Secondly there is an added security benefit here. The URL gives a lot away about a website, like what the code base is, and is potentially a window on the internal workings of a website. Take a normal FuseBox application for example. The normal URL might be:

```
www.mysite.com/index.cfm?fuseaction=controller.action&othervalues=values
```

From here it is very easy to start messing around with the controller names, trying to dig out an 'admin' controller, or other common function controller. Similarly adding values to pages where it is obvious a Query has been fired is an easy way of testing of the developer is using 'cfQueryParam', with potentially disastrous results.

Along the same lines it is quite simple to inject form values into the URL (like this <http://www.mccran.co.uk/index.cfm/2009/7/30/Cross-site-Script-hacking-using-the-GET-method>). By masking the URL and the values you make it considerably more difficult to do this, after all if you can see or get to the URL, how can you fool around with it?

So far I am implementing a rewrite script that will rewrite URLs into friendly strings, here is a modified version of the .htaccess file I'm using.

```
# Helicon ISAPI_Rewrite configuration file
# Version 3.1.0.68

RewriteEngine on
RewriteBase /wwwroot/

#generic
RewriteRule requestID/(.*)/(.*)/ index.cfm?decryptURL=$1&params=$2

# site pages
RewriteRule home(/)? index.cfm?go=controller.home
RewriteRule contact(/)? index.cfm?go=controller.contact
RewriteRule login(/)? index.cfm?go=controller.login
RewriteRule privacy(/)? index.cfm?go=controller.privacy
RewriteRule about(/)? index.cfm?go=controller.about
RewriteRule faqs(/)? index.cfm?go=controller.faqs
RewriteRule search(/)? index.cfm?go=controller.search
```

This code starts off by turning the rewriteEngine on, then setting the rewriteBase, this is typically your webroot, or the root of the site the file is for. Then it rewrites any URL params to the URL string.

The main part of the code is where we set individual rewriteRule's for each URL. The first example (home) looks for any URL requests to the 'home' string, and re writes this to the URL in the regular expression (index.cfm?go=controller.home). Pretty straight forward really.

There is a lot more you can do with this, and hopefully I'll get to explore rewriting in more depth in the future.