



Intergral Information Solutions GmbH

Schickardstr 32 • D-71034 • Böblingen • Germany

FRAPI Examples in ColdFusion

Doc. Rev. 228, 22 January 2008

INTER **GRAL**
information solutions

Trademarks and Warranties

FusionReactor, the FusionReactor logotype, and the Integral logotype are trademarks of Intergral Information Solutions GmbH and may not be used without permission. Other trademarks are the property of their respective owners.

The FusionReactor software product, including the FusionReactor JDBC Driver Wrapper is commercial software and may not be redistributed except with the express written agreement of Intergral Information Solutions GmbH. The software may only be used in accordance with the appropriate FusionReactor license agreement.

To the fullest extent applicable by law, Intergral Information Solutions hereby disclaims all warranties, including but not limited to the warranty of merchantability and the warranty of fitness for a particular purpose.

Feedback

We welcome feedback on all our products and publications. Please e-mail them to support@fusion-reactor.com and we will address them as quickly as possible.

Published in Germany

Copyright © 2005-2008 Intergral Information Solutions GmbH

All Rights Reserved

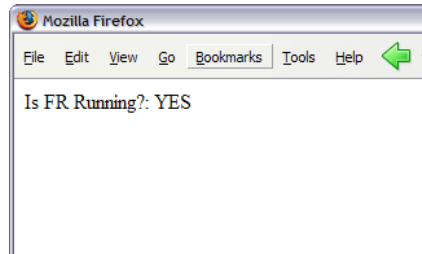
The FRAPI lets you call FusionReactor methods directly from your ColdFusion pages. JavaDocs are available for the FRAPI and are available from the FusionReactor site. Lets first start with some simple examples though...

This is one of the simplest FRAPI calls:

```
<cfset frapiClass = createObject("java", "com.intergral.fusionreactor.api.FRAPI")>
<cfset frapi = frapiClass.getInstance()>

<cfoutput>Is FR Running?: #frapi.isFusionReactorRunning()#</cfoutput>
```

This page simply outputs a YES/NO value depending on the running state of FusionReactor.



Now for something a little more interesting:

```
<cfset frapiClass = createObject("java", "com.intergral.fusionreactor.api.FRAPI")>
<cfset frapi = frapiClass.getInstance()>
<cfset dateClass = createObject("java", "java.util.Date")>

<!-- Loop over each running request -->

<cfset requestArray = frapi.getRunningRequests() />
<hr>
<cfloop index="i" from=1 to=#ArrayLen(requestArray)#>

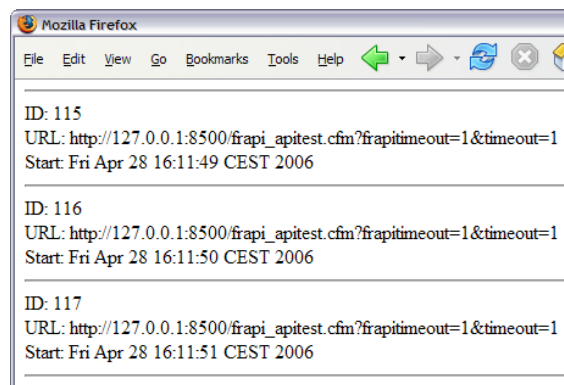
  <cfset dateObject = dateClass.init(requestArray[i].getStartTimeMillis())>
  <cfoutput>
    ID: #requestArray[i].getFusionRequestId()#<br/>
    URL: #requestArray[i].getRequestUrl()#<br/>
    Start: #dateObject.toString()#<br/>
  </cfoutput>
  <hr>
</cfloop>

<!-- Sleep for 5 seconds -->

<cfobject type="java" action="create" class="java.lang.Thread" name="Thread">

<cfscript>
  Thread.sleep( JavaCast( "long", 5000 ) );
</cfscript>
```

This bit of code will print out a list of all requests currently running. (We've added a wait at the bottom so we can run it several times and get a result something like this:



As well as getting information from FusionReactor, you can also use the FRAPI to change FusionReactor itself. The following code will change the timeout setting for Crash Protection for this request:

```
<cfset frapiClass = createObject("java", "com.intergral.fusionreactor.api.FRAPI")>
<cfset frapi = frapiClass.getInstance()>

<!--- Force Crash Protection to 10 seconds for this page --->
<cfset frapi.setPageTimeout( 10000 )>
```

We also have the ability to set markers in pages. Look at this example:

```
<cfset frapiClass = createObject("java", "com.intergral.fusionreactor.api.FRAPI")>
<cfset frapi = frapiClass.getInstance()>

<!--- Set a marker --->
<cfset frapi.trace( "Hello World!" )>
```

After this page has run, if we go into FusionReactor and look at the request details for this request we will see this marker listed along with the time at which it was set:

The screenshot shows the 'Request Detail' view in FusionReactor. At the top, there are tabs for 'Requests' and 'Request Detail', and a 'Back' button. A 'Refresh' button is also present. Below this is a table with the following data:

Finished	Ret.Code/IP	ID	URL/Parameters	Time(ms)	Memory(KB)
16:38:28.520 28-Apr-2006	200 127.0.0.1	5 web-0	http://127.0.0.1:8500/markerTest.cfm	109 0	Cur:(2%)5,858 Free:196,701

Below the table are several tabs: 'Main', 'Cookies', 'AMF', 'Headers', 'JDBC', and 'User Trace Markers'. The 'User Trace Markers' tab is selected, showing a section titled '0: FRAPI-UserTrace'. This section contains the following information:

- GenerationTime: 16:38:28.427 28-Apr-2006
- Message: Hello World!

The ability to set markers within your pages has many useful purposes including following the path of execution, measuring execution time or simply outputting debug information without changing the final output or needing to read through log files.

As we hope you can see, the FRAPI is a very powerful new tool which has a lot of potential. As ever, if you have any feedback or suggestions for this or any other feature within FusionReactor, please let us know and we will try our best to include them in upcoming builds.