Kune - Enhancement #70
Implement some kind of "/etc/init.d/kune reload" to reload configuration

07/27/2011 01:42 PM - Vicente J. Ruiz Jurado

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Pablo Ojanguren	**% Done:**	100%
Category:	Server side	**Estimated time:**	0.00 hour
Target version:			
Resolution:		**Tags:**	

Description

Related
http://ykyuen.wordpress.com/2010/07/26/jetty-stop-a-jetty-server-by-command/
http://wiki.eclipse.org/Jetty/Howto/Secure_Termination
http://irc.codehaus.org/display/JETTY/How+to+gracefully+shutdown
http://ptrthomas.wordpress.com/2009/01/24/how-to-start-and-stop-jetty-revisited/
http://eureka.ykyuen.info/2010/07/26/jetty-stop-a-jetty-server-by-command/

Associated revisions

Revision a80e8a84 - 03/22/2013 04:00 PM - Pablo Ojanguren

Implemented #70 Reload server configuration via JMX

Revision 28356438 - 03/27/2013 09:51 PM - Vicente J. Ruiz Jurado

Some work with #520 #525 and following #70

Revision 5b3965e2 - 04/02/2013 06:53 PM - Vicente J. Ruiz Jurado

Added some mbean methods to SearchEngineServletFilter (related to #512 #286 and #70)

History

#1 - 08/16/2011 10:53 PM - Vicente J. Ruiz Jurado

Something similar to:
http://ptrthomas.wordpress.com/2009/01/24/how-to-start-and-stop-jetty-revisited/
but instead of stop, to reload the configuration.

#2 - 03/28/2012 01:24 AM - Vicente J. Ruiz Jurado

- Status changed from New to Closed

- Resolution set to fixed

#3 - 02/28/2013 08:03 PM - Vicente J. Ruiz Jurado

- % Done changed from 0 to 100

@pablojan suggest to use

http://en.wikipedia.org/wiki/Java_Management_Extensions

#4 - 03/06/2013 10:48 AM - Vicente J. Ruiz Jurado

- Status changed from Closed to New

- % Done changed from 100 to 0

- Resolution deleted (fixed)

Closed by error, sorry...

#5 - 03/06/2013 08:50 PM - Vicente J. Ruiz Jurado

- Assignee changed from Vicente J. Ruiz Jurado to Pablo Ojanguren

#6 - 03/08/2013 09:02 PM - Pablo Ojanguren

- File jmx-in-action-1.png added

Implemented initial version following these steps:

1) Create MBean interface for class to manage

KunePropertiesDefaultMBean

```
package cc.kune.core.server.properties;

/**
 * MBean interface for JMX management of server properties
 *
 */
public interface KunePropertiesDefaultMBean {

    String reload();
    String getProperty(String key);
    void setProperty(String key, String value);

}
```

2) Implement mbean interface new methods

KunePropertiesDefault

```
diff --git a/src/main/java/cc/kune/core/server/properties/KunePropertiesDefault.java
b/src/main/java/cc/kune/core/server/properties/KunePropertiesDefault.java
index aa8639f..ed79f6d 100644
--- a/src/main/java/cc/kune/core/server/properties/KunePropertiesDefault.java
+++ b/src/main/java/cc/kune/core/server/properties/KunePropertiesDefault.java
@@ -1,6 +1,6 @@
 /*
```

```
 *
- * Copyright (C) 2007-2012 The kune development team (see CREDITS for details)
+ * Copyright (C) 2007-2013 The kune development team (see CREDITS for details)
 * This file is part of kune.
 *
 * This program is free software: you can redistribute it and/or modify
@@ -35,7 +35,7 @@
 import com.google.inject.Singleton;

 @Singleton
-public class KunePropertiesDefault implements KuneProperties {
+public class KunePropertiesDefault implements KuneProperties, KunePropertiesDefaultMBean {
   public static final Log LOG = LogFactory.getLog(KunePropertiesDefault.class);
   private CompositeConfiguration config;
   private final String fileName;
@@ -124,4 +124,33 @@
     }
   }

+  @Override
+  public String reload() {
+
+      String resultInfoStr = "OK";
+
+      try {
+
+          loadConfiguration();
+
+      } catch (final ServerException e) {
+
+          resultInfoStr = e.getMessage();
+      }
+
+      return resultInfoStr;
+  }
+
+  @Override
+  public String getProperty(String key) {
+
+      return this.get(key);
+  }
+
+  @Override
+  public void setProperty(String key, String value) {
+
+      this.setProperty(key, value);
+  }
+
 }
```

3) Add mbean activation just after KuneProperties object is created

```
diff --git a/src/main/java/cc/kune/core/server/persist/DataSourceKunePersistModule.java
b/src/main/java/cc/kune/core/server/persist/DataSourceKunePersistModule.java
index 5b097bf..2ae371b 100644
--- a/src/main/java/cc/kune/core/server/persist/DataSourceKunePersistModule.java
+++ b/src/main/java/cc/kune/core/server/persist/DataSourceKunePersistModule.java
@@ -21,8 +21,15 @@

 import java.io.IOException;
 import java.io.InputStream;
+import java.lang.management.ManagementFactory;
 import java.util.Properties;

+import javax.management.InstanceAlreadyExistsException;
+import javax.management.MBeanRegistrationException;
+import javax.management.MBeanServer;
+import javax.management.MalformedObjectNameException;
+import javax.management.NotCompliantMBeanException;
+import javax.management.ObjectName;
 import javax.persistence.EntityManager;

 import org.apache.commons.configuration.SystemConfiguration;
@@ -220,6 +227,33 @@
     final SystemConfiguration sysConf = new SystemConfiguration();
     kuneConfig = settedProperties != null ? settedProperties : sysConf.getString("kune.server.config");
     kuneProperties = new KunePropertiesDefault(kuneConfig);
+
+    /* #70 Reload Configuration Properties via JMX */
+    MBeanServer mbeanServer = ManagementFactory.getPlatformMBeanServer();
+    ObjectName mbeanName = null;
+    try {
+
+        mbeanName = new ObjectName("cc.kune.mbeans:type=KuneProperties");
+
+    } catch (MalformedObjectNameException e) {
+        e.printStackTrace();
+    } catch (NullPointerException e) {
+        e.printStackTrace();
+    }
+
+    try {
+
+        mbeanServer.registerMBean(kuneProperties, mbeanName);
+
+    } catch (InstanceAlreadyExistsException e) {
+        e.printStackTrace();
+    } catch (MBeanRegistrationException e) {
+        e.printStackTrace();
+    } catch (NotCompliantMBeanException e) {
+        e.printStackTrace();
+    }
+
```

```
+
    log4Conf = sysConf.getString("log4j.configuration");
  }
```

4) Tested successfully using jconsole through local jetty-java process (see attached picture)

Next steps

- Remote access to JMX Mbeans hasn't been tested. Maybe jetty additional configuration is needed. (in progress)

- Regression test. Unknown collateral effects when some properties are changed (e.g. db connection?)

- To change a property doesn't imply to system be aware of it. If some object has already read the property (e.g. on its constructor) won't get new value anyway. Maybe JMX event feature should be enabled for these classes.

- It only affects to server side of RPC server. How are changes populated to GWT client?

#7 - 03/10/2013 01:43 PM - Vicente J. Ruiz Jurado

Looks good to me ;) I'll just use LOG.error("Something", e) instead of printStackTrace. If you want to commit...

Can you use http://reviews.comunes.org in the future with these kind of patchs?

Clients gets the properties from a SiteRPC call, on each reload. So, if SiteRPC reloads is properties, is enough.

Don't worry about mysql connection properties, etc. For now it's not a necessity to change this kind of things.

#8 - 03/22/2013 04:15 PM - Pablo Ojanguren

- Status changed from New to Resolved

- % Done changed from 0 to 80

#9 - 03/22/2013 04:17 PM - Pablo Ojanguren

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

Files

jmx-in-action-1.png	187 KB	03/08/2013	Pablo Ojanguren