

Example Application Exposing Internals Using JMX

Here's a simple application that illustrates exposing internals using JMX.

Description

Application contains:

- Web deployment descriptor (e.g. WEB-INF/web.xml) that describes the application (optional)
- JSP views: home.jsp and WEB-INF/views/view.jsp
- A counter (e.g. example.jmx.web.MyCounter) that can increment, get current count, reset counter and reset to a specific value.
- A servlet (e.g. example.jmx.web.CounterServlet) that calls counter and forwards to a view (e.g. WEB-INF/views/view.jsp)
- A servlet listener (e.g. example.jmx.web.RegisterMBeansListener), that registers the MBean to JMX MBeanServer
- An MBean interface (e.g. example.jmx.mbean.CounterMBean) that defines JMX attributes and operations, e.g. name, resetCounter, getCurrentCount, getCurrentTime
- An MBean implementation (e.g. example.jmx.mbean.Counter) that implements MBean interface

Code Examples

example.jmx.web.MyCounter

```
package example.jmx.web;

import java.util.concurrent.atomic.AtomicInteger;

public class MyCounter {

    private static AtomicInteger counter = new AtomicInteger(0);

    public static int incrementAndGet() {
        return counter.incrementAndGet();
    }

    public static int getCurrentCount() {
        return counter.get();
    }

    public static void resetCounter() {
        resetCounter(0);
    }

    public static void resetCounter(int start) {
        counter = new AtomicInteger(start);
    }

}
```

example.jmx.web.CounterServlet servlet

```
package example.jmx.web;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/count")
public class CounterServlet extends HttpServlet {

    private static final long serialVersionUID = 886395215542306826L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {

        request.setAttribute("COUNT", MyCounter.incrementAndGet());
        request.getRequestDispatcher("/WEB-INF/views/view.jsp").forward(request, response);
    }
}
```

example.jmx.web.RegisterMBeansListener web listener

```

package example.jmx.web;

import java.lang.management.ManagementFactory;

import javax.management.InstanceAlreadyExistsException;
import javax.management.InstanceNotFoundException;
import javax.management.MBeanRegistrationException;
import javax.management.MBeanServer;
import javax.management.MalformedObjectNameException;
import javax.management.NotCompliantMBeanException;
import javax.management.ObjectName;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import javax.servlet.annotation.WebListener;

import example.jmx.mbean.CounterMBean;
import example.jmx.mbean.Counter;

@WebListener
public class RegisterMBeansListener implements ServletContextListener {

    private ObjectName objectName;

    public RegisterMBeansListener() {
    }

    @Override
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("Registering MBean...");
        final MBeanServer server = ManagementFactory.getPlatformMBeanServer();
        try {
            objectName = new ObjectName("JmxExampleApp:type=Counter");
            final CounterMBean mbean = new Counter();
            server.registerMBean(mbean, objectName);
            System.out.println("MBean registered: " + objectName);
        } catch (MalformedObjectNameException mone) {
            mone.printStackTrace();
        } catch (InstanceAlreadyExistsException iaee) {
            iaee.printStackTrace();
        } catch (MBeanRegistrationException mbre) {
            mbre.printStackTrace();
        } catch (NotCompliantMBeanException ncmb) {
            ncmb.printStackTrace();
        }
    }

    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("Unregistering MBean...");
        final MBeanServer server = ManagementFactory.getPlatformMBeanServer();
        try {
            objectName = new ObjectName("JmxExampleApp:type=Counter");
            server.unregisterMBean(objectName);
            System.out.println("MBean unregistered: " + objectName);
        } catch (MalformedObjectNameException mone) {
            mone.printStackTrace();
        } catch (MBeanRegistrationException mbre) {
            mbre.printStackTrace();
        } catch (InstanceNotFoundException infe) {
            infe.printStackTrace();
        }
    }
}

```

example.jmx.mbean.CounterMBean interface

```

package example.jmx.mbean;

import java.util.Date;

public interface CounterMBean {
    void resetCounter();
    void resetCounter(int start);
    int getCurrentCount();
    Date getCurrentTime();
    long getCurrentTimeMillis();
}

```

example.jmx.mbean.Counter implementation

```

package example.jmx.mbean;

import java.util.Date;

import example.jmx.web.MyCounter;

public class Counter implements CounterMBean {

    private String name;

    public Counter() {
        this.name = "InitialName";
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public void resetCounter() {
        MyCounter.resetCounter();
    }

    @Override
    public void resetCounter(int start) {
        MyCounter.resetCounter(start);
    }

    @Override
    public int getCurrentCount() {
        return MyCounter.getCurrentCount();
    }

    @Override
    public Date getCurrentTime() {
        return new Date();
    }

    @Override
    public long getCurrentTimeMillis() {
        return System.currentTimeMillis();
    }

}

```

Download

Download [JmxExample.war](#) (with source code).