Jetty and Spring EJB3 (Pitchfork)

Jetty and Spring EJB3 (Pitchfork)

Pitchfork is an add-on for the Spring Framework performing JSR-250 dependency injection, annotation processing and EJB 3.0-style interception, developed jointly by Interface21 and BEA Systems.

It is simple to use in conjunction with Jetty.

At the time of writing, the integration with Pitchfork is checked in to jetty svn trunk, but will be released with 6.1.0 final. Until then, it will be first necessary to checkout and build jetty.

Once jetty is built, make a new directory: `$jetty.home/lib/ext/spring`. Copy the jar from `$jetty.home/extras/spring/target` into it. Also copy all of the jars from the Pitchfork download into it.

Developing a webapp using EJB3

The JEE tutorial from Sun will take you through how to create EJB3 classes.

Here's an extremely simple example of a stateless session bean showing some of the JEE-style annotations:

```
package com.acme;

import java.util.Date;
import javax.annotation.PostConstruct;
import javax.ejb.Remote;
import javax.ejb.Stateless;
import javax.interceptor.Interceptors;

@Stateless
@Remote(Echo.class)
public class EchoBean implements Echo
{
    public String echo()
    {
        return "Hello "+new Date();
    }

    @PostConstruct
    public void init()
    {
        System.err.println("EchoBean init
called");
    }
}
```

We would declare this class in a `META-INF/spring-ejb-jar.xml` file. That file can reside in a jar if you're packaging your ejbs into jars, or you can make a `WEB-INF/classes/META-INF/` directory if your ejbs are just part of your webapp.

The spring-ejb-jar.xml file for the above bean looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
xmlns="http://www.springframework.org/schema
/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:aop="http://www.springframework.org/sc
hema/aop"

xsi:schemaLocation="http://www.springframewo
rk.org/schema/beans
http://www.springframework.org/schema/beans/
spring-beans.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/sp
ring-aop.xsd">

  <bean id="EchoBean"
class="com.acme.EchoBean"></bean>

</beans>
```

Next, you can make some classes that use @EJB notation to inject references to your beans. Here's another stupidly simple class that uses the `com.acme.EchoBean` ejb:

```
package com.acme;

import javax.ejb.EJB;

public class EchoTest
{
  @EJB private static EchoBean echoBean;

  public static String echo ()
  {
      return echoBean.echo();
  }
};
```

In order to have Pitchfork notice that the `com.acme.EchoTest` class wants a reference to the `com.acme.EchoBe` an ejb, we declare it in the Spring `WEB-INF/applicationContext.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="no"

xmlns="http://www.springframework.org/schema
/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xmlns:aop="http://www.springframework.org/sc
hema/aop"

xmlns:tx="http://www.springframework.org/sch
```

```
ema/tx"

xsi:schemaLocation="http://www.springframewo
rk.org/schema/beans
http://www.springframework.org/schema/beans/
spring-beans-2.0.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/sp
ring-aop-2.0.xsd

http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spr
ing-tx-2.0.xsd">

  <bean id="echoTest"
class="com.acme.EchoTest"/>

  <bean id="jeeEjbPostProcessor"
```

```
class="org.springframework.jee.ejb.config.Je
eEjbBeanFactoryPostProcessor"/>
</beans>
```

Notice the very important second line with id **jeeEjbPostProcessor**. This line tells Pitchfork to look at all other <bean> declarations and perform any necessary resource injections on them.

To complete the webapp, here's a trivial jsp:

```
<html><head>
<%@ page import="com.acme.*" %>
</head><body>
<h1>Echo EJB</h1>

<%
   String str = EchoTest.echo();
%>
 <%=str%>
</body></html>
```

When you deploy it and surf to it, you'll see something like this:

```
Echo EJB

Hello Thu Dec 07 12:14:16 CET 2006
```

A maven project that builds a deployble webapp for the example we've been looking at is attached to this page.