

java认证辅导:基于JBoss来写MBeanJava认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/607/2021_2022_java_E8_AE_A4_E8_AF_81_c104_607365.htm 用JDK自带的JMX实现来写的MBean，JMX的实现不独SUN一家，JBoss也有自己的JMX实现。如果你使用JBoss来做WEB服务器，那么基于JBoss的实现来写MBean，是一个不错的选择。象我们公司就是用JBoss的，因此所有MBean都是基于JBoss来写的。基于JBoss的MBean和基于SUN的MBean有什么不同吗？有一些不同之外，但绝大部分都一样。

一、 HelloWorld实例

1、 准备工作 JBOSS实现了JMX规范，这个实例是基于JBoss来实现的。请先去下载一个JBoss，我是JBoss-3.2.6，这个实例需要JBoss的两个JAR包的支持：jboss-system-3.2.6.jar、jboss-jmx-3.2.6.jar，如果你和我一样用Eclipse来开发（推荐），那么把这个两个包加入到项目的库引用中（加入到库引用的方法参考前面两章）。

2、 程序代码 假设我们有一个叫message的属性要需要经常进行改动配置的，那么我们就把它写成一个MBean。

1)、 HelloWorldServiceMBean接口 在写MBean之前，我们先需要写一个MBean接口，接口里的方法都是属性的set/get方法。这个接口必须继承接口ServiceMBean。

```
java 代码 import org.jboss.system.ServiceMBean. public interface HelloWorldServiceMBean extends ServiceMBean { String getMessage(). void setMessage(String message). }
```

2)、 HelloWorldService实现类 然后写出HelloWorldServiceMBean接口的实现类HelloWorldService，这个实现类还必须继承ServiceMBeanSupport类。这种类再简单不过了，就是属性和

相应的set/get方法，EJB中叫实体类、Hibernate中叫POJO。

```
java 代码 import org.jboss.system.ServiceMBeanSupport; public
class HelloWorldService extends ServiceMBeanSupport implements
HelloWorldServiceMBean { private String message; public String
getMessage() { System.out.println("getMessage()" message); return
message; } public void setMessage(String message) {
System.out.println("setMessage(" message ")"); this.message =
message; } }
```

3)、配置文件jboss-service.xml

```
xml version="1.0"
encoding="UTF-8"?lt;server?lt;mbean
code="example.mbean.HelloWorldService"
```

```
name="www.chengang.com.cn:service=HelloWorld"lt;attribute
```

```
name="Message"gt; mbean?gt; 说明： code项指向MBean的实现
```

```
类HelloWorldService name项是一个名称，格式一般是：[说
```

```
明性文字]:service=[类名] attribute是为属性设置初始值，这
```

```
样当JBOSS一加载HelloWorldService类时，message属性就有了
```

```
一个初始值 Hello World。注意Message的第一个字母必须是大
```

```
写。二、将实例部署到JBOSS
```

```
在jboss-3.2.6\server\default\deploy目录下创建一个hello.sar目录
```

```
，然后创建如下目录文件结构： hello.sar |----example |
```

```
|----mbean | |----HelloWorldService.class (注意：是*.class，不
```

```
是*.java) | |----HelloWorldServiceMBean.class |----META-INF
```

```
|----jboss-service.xml 其他说明： 也可以将hello.sar目录用zip
```

```
格式压缩成一个同名的hello.sar文件，放
```

```
到jboss-3.2.6\server\default\deploy目录下。 JBOSS支持热部署
```

```
，也就是说你在布置这个目录时并不需要重启JBOSS。三
```

```
、MBean的效果 后单击“ service=HelloWorld ”项打开详细页
```

面将“ HelloWorld ”改成“ Hello World , ChenGang ”，再单击“ Apply Changes ”应用修改

四、其他类如何使用Message属性

现在我们可以通过一个自动提供的WEB页面来设置Message属性了，接下来的问题是：“在其他类中应该如何得到Message的属性值”。MBean在JBoss是只保留一个MBean的实例（单例模式？），也就是说问题转成我们如何去取得这个唯一实例。例程如下：

- 1、创建一个使用到Message属性的类

```
import org.jboss.mx.util.MBeanProxyExt; import
org.jboss.mx.util.ObjectNameFactory; public class Client { public
void go() { HelloWorldServiceMBean mbean =
(HelloWorldServiceMBean)
MBeanProxyExt.create(HelloWorldServiceMBean.class,
ObjectNameFactory.create(www.chengang.com.cn:service=HelloW
orld)). String msg = mbean.getMessage();
System.out.println("Client.go()=" msg); } }
```

注意：go方法里是三句。第一句比较长，它是根据jboss-service.xml文件中设置的MBean名称，来取得此MBean在JBOSS中的实例。

- 2、在Mbean中加一个相应的调用Client.go的方法

在HelloWorldServiceMBean接口中加入一句：void callGo().

```
ublic void callGo() { new Client().go(); }
```

- 3、更新布署 将三个类的class文件：Client.class、 HelloWorldServiceMBean.class、 HelloWorldService.class，更新到JBOSS的hello.sar\example\mbean目录下。然后重启JBOSS。
- 4、查看效果 打开JBOSS提供的MBean设置页面，发现多了一个callGo。单击callGo项后的invoke按钮，得到DOS输出

五、其他说明

本实例仅演示了一个Message属性，你当然可以

在HelloWorldService中加入更多属性，别忘了在HelloWorldServiceMBean接口也加入相应的set/get方法。本实例的message属性是String类型的，但JMX也支持其他的类型，甚至是XML信息。对于jboss-service.xml中的XML信息，这时属性类型要求是org.w3c.dom.Element，JMX将它封装成了一个XML的DOM对象。回顾一下，JMX也是将配置信息写在了一个文件（jboss-service.xml文件）里嘛，相对于将配置文件写到*.properties文件的方式，它似乎也没什么新鲜的地方。但通过本章实例我们可以看到JMX的一些好处：我们不用写代码去读配置文件的信息，而且JMX支持的属性类型是多种多样的（如上面说的org.w3c.dom.Element）。更重要的是JMX还提供了一整套的属性之间互相访问、互相调用的功能，这个HelloWorld实例所反映的只是冰山一角而已。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com