

JBoss4.0简化了中间件的开发 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_JBoss40_E7_AE_c104_167308.htm 概要 JBoss 应用程序服务器 (AS) 4.0不仅是一个通过J2EE认证的应用程序服务器，而且也是多种领先优势开源技术的一个融结点。这些技术简化了基于POJO的中间件应用的开发模型，而且将成为下一代J2EE标准。在这篇文章中，Dr. Michael 除了探究了JBoss AS 4.0中的新特性之外，还给你预览了明日将出现什么。（2005/02/21）在2004年的九月，JBoss应用服务器 (AS) 4.0 通过了J2EE 1.4的认证。对于JBoss的核心开发人员和JBoss早期的采用者，JBoss AS 4.0最振奋的并不是J2EE的认证，而是目前J2EE无法涵盖的新技术和极大地简化Java中间件开发的目标。就是使用更加简单的，更加易于管理的POJO来替代已存在的EJB的理念。简化程序将提高开发人员的效率，更好的程序性能和更少的Bug.简单化（a.k.a 轻量级开发）将是服务器端Java社区下一个重大的事件，JBoss AS 4.0 将是第一个在该方面迈出坚实步伐的J2EE主流服务器。这篇文章中，我将用三个示例程序来展示JBoss AS 4.0中POJO中间件框架的简单性，以及他们是如何与当前和明日的J2EE规范关联起来的。如果你是一个JBoss的用户或者一个普通的J2EE开发人员，这篇文章将教你一些不仅在目前的JBoss AS 4.0和将来的JBoss 5.0 或者 J2EE 1.5 服务器上可以应用的轻便技巧。让我们从目前EJB 2.1中间件框架中固有的问题开始，来展示对一个更加简单的基于POJO框架的需要。（开源和J2EE规范——对于Java社区和开源社区来说，JBoss的官方J2EE认证是一个具有里程碑意义的事件。因为不久之前

，由于高成本和所要求的大量的保证质量的工作，人们认为，如果Sun不发慈悲的话，任何开源的J2EE项目都不可能通过认证的。JBoss 仅仅依靠自己就获得了J2EE的认证，证明了开源开发模型在交付迅速的企业Java解决方案的正确性）（一）EJB 2.1出了什么问题了？开始的时候，J2EE 在开发具有伸缩性和分布性的服务器端应用市场获得了巨大的成功。然而，EJB，在J2EE里的一个核心的开发中间件的构件，却获得了一个太复杂和难用的名声，特别是对于中小型业务应用的开发。额外的EJB基础代码和部署描述符不仅使服务器资源承担不起，而且，更加重要的是降低了开发人员的效率。导致开发人员最终写更多了和需要维护更多的基础代码而不是业务逻辑。为了证明以上观点，和提供一个JBoss AS 4.0 支持的更加简单的解决方案的选择性的比较，让我们来看一个基于EJB 2.1的示例程序。抵押金计算器Web应用程序先计算每个月每笔贷款的抵押金，将结果保存到一个关系数据库里。每次计算之后，该程序在数据库中之前的结果中搜索所需数额较低的抵押金。那些结果将在页面的底部显示。图1 展示了该程序如何工作的。当你初次使用该程序的时候，你将被要求填入一个用户名和密码。使用“ user1 ” / “ pass1 ” 登录，如果你想看到显示给未授权用户的错误信息，请尝试“ user3/pass3 ”。图1 实战抵押金计算器web应用，从示例源码包ejb2文件夹可以获取该程序的源代码，在ejb2目录（ Windows ）只要执行build.bat命令或者build.sh命令（ Linux ， Unix ， 或者Mac OS X系统 ）来重建该应用。将生成的ejb2/build/jar/MortgageCalculatorEJB2.ear文件拷贝到JBoss 服务器的server/default/deploy/部署。访问该应用的URL将是

http://localhost : 8080/MortgageCalculatorEJB2/servlet/Calculator.

为何要使用EJB？ 对一个如此简单的应用为什么使用EJB？ EJB容器提供了一些有用的服务，无须我们写其他代码，这些服务可以立即增加一些企业性的特性给我们的web应用。例如，EJB会检查用户的信任度，对于所有的EJB方法的调用，容器将根据配置文件监控其相关的数据库事务。另外容器也管理者数据库的表，和数据库连接，所有这些都无须我们写任何的SQL或者JDBC代码。 在该分支下嘛，Web应用有一个servlet接受用户的输入，同时产生HTML页面。Servlet将押金计算和数据库相关的工作分派给一个EJB模块来完成。 该应用有两个EJB构件，Calculator bean是一个无状态的会话bean，它包含了计算押金的，保存结果到数据库的，搜索数据库的事务性的方法。这些方法都曝露给了servlet. Calculator bean使用了History实体bean访问数据库。在EJB的配置文件里，我们定义了History实体bean的数据域是如何映射到数据库的列的，对于一系列的History对象如何搜索数据库。由于History bean是一个EJB，它无法在EJB容器外使用，所以我们创建了HistoryList 值对象来保持任何的搜索操作，结果将返回给servlet. 图2，阐明了EJB模块的结构。它展示了所有需要的EJB组成接口和部署描述符的要素。 图2 EJB模块的重要构件。如你所看到的，在图2中展示的结构是复杂的，包括了几个紧耦合的Java接口，类，和史前的XML.如果一个框架允许开发人员专注于他们所擅长的也就是写Java代码而不是用一堆构件接口和XML转移他们的注意力，那该多好啊！好消息是，JBoss AS 4.0 中的POJO中间件框架使开发人员很好的利用EJB容器的服务而没有EJB2.1的包袱。（二）简单化的创新

从使用EJB危险中，我们认识到一个成功的可选框架应该具有以下重要特性：。该框架不应该给开发人员增加任意的构件模型，因为这些模型会打破面向对象的设计结构。换句话说，该框架应该支持一些开发人员可以对其扩展和在应用容器内部或者外部重用的POJO。该框架应该摒弃需要手工编写大量冗余的EJB部署描述符。一个POJO应该可以以简单的声明它需要什么样的容器服务。该框架应该支持通过引用对POJO进行本地访问。Java对象序列化是缓慢的，在允许的时候应该避免使用，特别是对于小型到中型的应用更加需要避免。在过去的几年，Java开源社区并没有等待Java社区过程（JCP）去发明或者标准化一个基于POJO的轻量级的中间件框架。而是已经实验了无数的方法。这些开源的项目的例子包括了XDoclet，Hibernate，Spring和一些面向方面的编程（AOP）项目。JBoss 4.0 中的新的中间件框架补充了过去的研究和开发努力。两个JBoss赞助的开源项目在它的POJO中间件框架中扮演了重要的角色。JBoss AOP项目支持了通过Java注解把服务传输给POJO。Hibernate项目是对象关系映射和POJO持久化实际上的标准框架。为了支持企业级的应用，两个项目都集成和强化了经过检验的，和通过J2EE认证的JBoss AS容器服务。现在让我们重构抵押金计算器EJB构件成为POJO，看看这个强大的新开发模型是如何工作的。企业POJO的强大之处 POJO抵押金计算器应用也包含在源码包里。就如我将要说明的，该应用的源代码和创建脚本对于J2SE 1.4和J2SE 1.5环境有稍微的不同。我准备了两个版本，分别存放在pojo-jdk14 pojo-jdk15目录里。创建和部署的命令跟ejb2样例应用都是一样的。一旦部署了，访问POJO样例

的URL将是 `http://localhost`

`: 8080/MortgageCalculatorPOJO/servlet/Calculator`. 就如我讲述的，我们的目标是使用POJO替代会话和实体bean.期间，该框架应该能够保持EJB服务器的一个关键的好处声明应用容器服务给应用程序。这样的话，当我们今后需要改变容器服务的时候，我们仅仅需要修改声明而且无须修改大量的Java代码。但是，无须部署描述符，一个POJO如何告知JBoss容器它需要什么服务呢？这就是AOP和Java注解表演的地方了。重用注解 注解是从J2SE 5.0开始引入为官方Java语言语义的一部分。JBoss AS 4.0定义了一套注解标签作为POJO访问J2EE容器服务的API，并且声明了容器服务是如何应用的。在该分支下，JBoss AS使用了JBoss AOP框架来动态改变被注解的对象和方法的行为。从JBoss AOP的用户指导和参考文档，你可以找到更多关于JBoss AOP的信息和它是如何与注解一道运作的。以下列出了替代了Calculator会话Bean的Calculator POJO类的基本骨架。

```
@SecurityDomain ("other")public class Calculator {  
    @Unchecked public Calculator () {}  
    @Permissions ({ "AuthorizedUser" }) @Tx (TxType.REQUIRED) public double  
    getPayment (int principal, double rate, int term) throws Exception {  
        // Calculate and save to database. // Code omitted for clarity. }  
    @Permissions ({ "AuthorizedUser" }) @Tx (TxType.REQUIRED)  
    public List getHistory (double payment) throws Exception { //  
        Search the database. // Code omitted for clarity. }}
```

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