

初学者如何开发出一个高质量的J2EE系统 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_\\_E5\\_88\\_9D\\_E5\\_AD\\_A6\\_E8\\_80\\_85\\_E5\\_c104\\_252359.htm](https://www.100test.com/kao_ti2020/252/2021_2022__E5_88_9D_E5_AD_A6_E8_80_85_E5_c104_252359.htm) J2EE学习者越来越多，J2EE本身技术不断在发展，涌现出各种概念，本文章试图从一种容易理解的角度对这些概念向初学者进行解释，以便掌握学习J2EE学习方向。首先我们需要知道Java和J2EE是两个不同概念，Java不只是指一种语言，已经代表与微软不同的另外一个巨大阵营，所以Java有时是指一种软件系统的流派，当然目前主要是.NET和Java两大主流体系。J2EE可以说指Java在数据库信息系统上实现，数据库信息系统从早期的dBase、到Delphi/VB等C/S结构，发展到B/S（Browser浏览器/Server服务器）结构，而J2EE主要是指B/S结构的实现。J2EE又是一种框架和标准，框架类似API、库的概念，但是要超出它们。如果需要详细了解框架，可先从设计模式开始学习。J2EE是一个虚的大的概念，J2EE标准主要有三种子技术标准：WEB技术、EJB技术和JMS，谈到J2EE应该说最终要落实到这三个子概念上。这三种技术的每个技术在应用时都涉及两个部分：容器部分和应用部分，Web容器也是指Jsp/Servlet容器，你如果要开发一个Web应用，无论是编译或运行，都必须要有JSP/Servlet库或API支持（除了JDK/J2SE以外）。Web技术中除了Jsp/Servlet技术外，还需要JavaBeans或Java Class实现一些功能或者包装携带数据，所以Web技术最初裸体简称为JSP/Servlet JavaBeans系统。谈到JavaBeans技术，就涉及到组件构件技术（component），这是Java的核心基础部分，很多软件设计概念（设计模式）都是通过JavaBeans实现的。

JavaBeans不属于J2EE概念范畴中，如果一个JavaBeans对象被Web技术（也就是JSP/Servlet）调用，那么JavaBeans就运行在J2EE的Web容器中；如果它被EJB调用，它就运行在EJB容器中。EJB（企业JavaBeans）是普通JavaBeans的一种提升和规范，因为企业信息系统开发中需要一个可伸缩的性能和事务、安全机制，这样能保证企业系统平滑发展，而不是发展到一种规模重新更换一套软件系统。至此，JavaBeans组件发展到EJB后，并不是说以前的那种JavaBeans形式就消失了，这就自然形成了两种JavaBeans技术：EJB和POJO，POJO完全不同于EJB概念，指的是普通JavaBeans，而且这个JavaBeans不依附某种框架，或者干脆可以说：这个JavaBeans是你为这个应用程序单独开发创建的。J2EE应用系统开发工具有很多：如JBuilder、Eclipse等，这些IDE首先是Java开发工具，也就是说，它们首要基本功能是可以开发出JavaBeans或Java class，但是如果开发出J2EE系统，就要落实到要么是Web技术或EJB技术，那么就有可能要一些专门模块功能(如eclipse需要lomboz插件)，最重要的是，因为J2EE系统区分为容器和应用两个部分，所以，在任何开发工具中开发J2EE都需要指定J2EE容器。J2EE容器分为WEB容器和EJB容器，Tomcat/Resin是Web容器；JBoss是EJB容器 Web容器等，其中Web容器直接使用Tomcat实现的。所以你开发的Web应用程序可以在上面两种容器运行，而你开发的Web EJB应用则只可以在JBoss服务器上运行，商业产品Websphere/Weblogic等和JBoss属于同一种性质。J2EE容器也称为J2EE服务器，大部分时它们概念是一致的。如果你的J2EE应用系统的数据库连接是通过JNDI获得，也就是说是从容器中获得，那么你

的J2EE应用系统基本与数据库无关，如果你在您的J2EE应用系统耦合了数据库JDBC驱动的配置，那么您的J2EE应用系统就有数据库概念色彩，作为一个成熟需要推广的J2EE应用系统，不推荐和具体数据库耦合，当然这其中如何保证J2EE应用系统运行性能又是体现你的设计水平了。衡量J2EE应用系统设计开发水平高低的标准就是：解耦性；你的应用系统各个功能是否能够彻底脱离？是否不相互依赖，也只有这样，才能体现可维护性、可拓展性的软件设计目标。为了达到这个目的，诞生各种框架概念，J2EE框架标准将一个系统划分为WEB和EJB主要部分，当然我们有时不是以这个具体技术区分，而是从设计上抽象为表现层、服务层和持久层，这三个层次从一个高度将J2EE分离开来，实现解耦目的。因此，我们实际编程中，也要将自己的功能向这三个层次上靠，做到大方向清楚，泾渭分明，但是没有技术上约束限制要做到这点是很不容易的，因此我们还是必须借助J2EE具体技术来实现，这时，你可以使用EJB规范实现服务层和持久层，Web技术实现表现层；EJB为什么能将服务层从JSP/Servlet手中分离出来，因为它对JavaBeans编码有强制的约束，现在有一种对JavaBeans弱约束，使用Ioc模式实现的（当然EJB 3.0也采取这种方式），在Ioc模式诞生前，一般都是通过工厂模式来对JavaBeans约束，形成一个服务层，这也是是Jive这样开源论坛设计原理之一。由此，将服务层从表现层中分离出来目前有两种可选架构选择：管理普通JavaBeans（POJO）框架（如Spring、JdonFramework）以及管理EJB的EJB框架，因为EJB不只是框架，还是标准，而标准可以扩展发展，所以，这两种区别将来是可能模糊，被纳入同一个标准了。但是，个人认

为：标准制定是为某个目的服务的，总要牺牲一些换取另外一些，所以，这两种架构会长时间并存。这两种架构分歧也曾经诞生一个新名词：完全POJO的系统也称为轻量级系统(lightweight)，其实这个名词本身就没有一个严格定义，更多是一个吸引人的招牌，轻量是指容易学习容易使用吗？按照这个定义，其实轻量Spring等系统并不容易学习；而且EJB 3.0（依然叫EJB）以后的系统是否可称为轻量级了呢？前面谈了服务层框架，使用服务层框架可以将JavaBeans从Jsp/Servlet中分离出来，而使用表现层框架则可以将JSP中剩余的JavaBeans完全分离，这部分JavaBeans主要负责显示相关，一般是通过标签库（taglib）实现，不同框架有不同自己的标签库，Struts是应用比较广泛的一种表现层框架。这样，表现层和服务层的分离是通过两种框架达到目的，剩余的就是持久层框架了，通过持久层的框架将数据库存储从服务层中分离出来是其目的，持久层框架有两种方向：直接自己编写JDBC等SQL语句（如iBatis）；使用O/R Mapping技术实现的Hibernate和JDO技术；当然还有EJB中的实体Bean技术。持久层框架目前呈现百花齐放，各有优缺点的现状，所以正如表现层框架一样，目前没有一个框架被指定为标准框架，当然，表现层框架现在又出来了一个JSF，它代表的页面组件概念是一个新的发展方向，但是复杂的实现让人有些忘而却步。在所有这些J2EE技术中，虽然SUN公司发挥了很大的作用，不过总体来说：网络上有这样一个评价：SUN的理论天下无敌；SUN的产品用起来撞墙；对于初学者，特别是那些试图通过或已经通过SUN认证的初学者，赶快摆脱SUN的阴影，立即开溜，使用开源领域的产品来实现自己的应用系统。

最后，你的J2EE应用系统如果采取上面提到的表现层、服务层和持久层的框架实现，基本你也可以在无需深刻掌握设计模式的情况下开发出一个高质量的应用系统了。还要注意的  
是：开发出一个高质量的J2EE系统还需要正确的业务需求理解，那么域建模提供了一种比较切实可行的正确理解业务需求的方法，相关详细知识可从UML角度结合理解。当然，如果你想设计自己的行业框架，那么第一步从设计模式开始吧，因为设计模式提供你一个实现JavaBeans或类之间解耦参考实现方法，当你学会了系统基本单元JavaBean或类之间解耦时，那么系统模块之间的解耦你就可能掌握，进而你就可以实现行业框架的提炼了，这又是另外一个发展方向了。以上理念可以总结为一句话：J2EE开发三件宝：Domain Model（域建模）、patterns（模式）和framework（框架）。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](http://www.100test.com)