

KodoEJB：符合EJB3规范的持久层框架 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/271/2021\\_2022\\_KodoEJB\\_EF\\_BC\\_9A\\_c104\\_271649.htm](https://www.100test.com/kao_ti2020/271/2021_2022_KodoEJB_EF_BC_9A_c104_271649.htm) Kodo是BEA公司收购SolarMetric公司后获得的持久层框架项目，以前只是支持JDO标准，2006年2月13日，BEA公司宣布发布Kodo项目的最新版本--Kodo 4.0.0 Early Access 4，Kodo 4.0.0 EA4支持EJB3和JDO2两个标准，本文中，我们将首先学习和了解KODO EJB，了解如何使用Kodo EJB完成开发工作。我们将按照两种不同的情况讲述如何使用Kodo EJB进行开发，一种是通过命令行工具，另外一种是在Eclipse中使用Ant任务。关于EJB3和JDO2的更多内容请大家查看文章最后参考资源中的相关内容。为什么使用Kodo EJB 在Kodo EJB框架中，对象和关系数据库之间的映射（对象-表，对象属性-字段等）都是使用JDK5.0中的最新特性注释（Annotation）来提供，不再需要提供额外的配置文件。根据EJB3规范的要求，Kodo EJB除了支持在EJB容器中使用满足重量级企业应用的需求之外，也支持在普通Java应用中采用，提供轻量级的持久层框架。只不过当我们在EJB容器中使用Kodo EJB时我们需要更多的工作以便满足EJB容器的要求。

下载、安装Kodo 准备工作 由于Kodo是基于注释机制的框架，我们必须使用JDK5.0完成开发工作。所以下载、安装Kodo之前，请确保您已经下载和安装了JDK5.0。为了演示的需要，我们选择MySQL数据库作为持久化的目标数据库，请大家自己到[www.mysql.com](http://www.mysql.com)下载最新的MySQL数据库后安装。下载、安装Kodo Kodo的最新版本是Kodo 4.0.0 Early Access 4，目前大家可以到<http://www.solarmetric.com/>去下载试用版本，

下载时需要注册，你会得到30天的License。将下载的压缩文件解压到c:/kodo4目录下(后面将使用%KODO\_HOME%来引用这个目录)，打开%KODO\_HOME%/bin/kodocmd.cmd文件，将其中的KODODIR设置为您的Kodo的安装目录，将JDKHOME设置为Java的安装目录。Kodo EJB实例上面的工作完成以后，我们就可以开发Kodo EJB应用了，下面的例子将主要讲述如何开发一个轻量级的Kodo EJB例子，支持在Java应用中的调用，而不依赖于EJB容器，关于如何在EJB容器中配置使用Kodo EJB是一个大的课题，作者将另外撰文详细论述。下面我们将用一个简单的例子来说明创建Kodo EJB应用的典型步骤，实例中我们将创建名为Book的持久化对象，该对象将被持久化到本地的MySQL数据库中。请大家注意的是，下面的说明都基于Windows 2000操作系统，如果您使用其它操作系统，可能需要做出相应的变更。建立工程目录在C:盘根目下创建名为KodoExamples的目录，我们所有的类文件和配置都放在这个目录下。创建持久化类 新创建持久化类Book，为了说明的简单，该类只有两个属性:id和name，其中id是书的编号（编号由MySQL数据库自动生成），name属性表示书的名称。持久化类的全部代码、注释和说明如下：

```
package org.vivianj.kodo.examples.beans; import
javax.persistence.Basic; import javax.persistence.Column; import
javax.persistence.Entity; import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType; import
javax.persistence.Id; import javax.persistence.Inheritance; import
javax.persistence.InheritanceType; import javax.persistence.Table;
/** * Book 用于表征系统中的书籍对象，他有两个属性id - 书
```

籍编号，书籍编号将由MySQL数据库自动生成name - 书名 \*/

```
/* Entity注释表示该类是持久化类，的name属性是该实体在查询中对应的唯一名称，默认是类名 */ (name = "Book") /* Table注释的name属性指定该持久化类对应的数据表的名称，默认数据表名和类名保持一致，为了增强代码的可移植性，建议大家name属性中使用大写英文字母 */ (name = "BOOKS") /* Inheritance注释的strategy确定了持久化对象和数据表之间的关系，可选择项包括SINGLE_TABLE、JOINED和TABLE_PER_CLASS，我们这里采用TABLE_PER_CLASS */ (strategy = InheritanceType.TABLE_PER_CLASS) public class Book { /* Id注释表示该字段是标识字段 */ /* GeneratedValue注释定义了该标识字段的产生方式，我们的演示系统中id由MySQL数据库字段自动生成，因此选择 GenerationType.IDENTITY */ (strategy = GenerationType.IDENTITY) /* Column注释的name属性定义了该类属性对应的数据字段的名称，为了最大限度保持系统和数据库之前的独立性，建议使用大写字符 */ (name = "ID") public int id. /* Basic注释表示该属性是基本属性 */ /* Column注释的name属性定义了该类属性对应的数据字段的名称，为了最大限度保持系统和数据库之前的独立性，建议使用大写字符 */ (name = "NAME") public String name = null.
```

准备数据库在MySQL数据库中新建名为Kodo的数据库。准备配置文件在C:\KodoExamples新建META-INF目录，随后在该目录下新建kodo.XML和persistence.xml文件。 a) Kodo.xml文件中提供访问数据库所需要的细节信息、使用Kodo所需要的授权(License)内容、Kodo运行时的日志管理等。 < ?xml

```
version="1.0"? > < persistence > < persistence-unit name="" >
< properties > <!-- Kodo的序列号，请输入您下载或者购
买Kodo时得到的License -- > < property
name="kodo.LicenseKey" value="093D-BF3F-C10E-0F8F-0F00" /
> 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
```