

KodoEJB：实现类和类之间的关联关系 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_271647.htm

对象和对象之间除了继承关系之外，还存在着关联关系：包括分作一对一、一对多、多对一和多对多，由于这几种关系在Kodo EJB中的实现原理基本类似，因此本文中主要就一对一类关联关系进行深入的讲述，同时通过简单例子的分析和实践详细的说明如何使用Kodo EJB中提供的注释来定义类和类之间的关联关系，剩下的一对多、多对一和多对多三种关系将只在文章最后进行说明，请读者参考一对一关系的实现过程。面向对象的世界里，类A和类B之间的一对一关系必须满足如下条件：对象A1引用了对象B1 类A其他对象An不可能应用同样的对象B1 在关系数据库中，我们通常使用唯一外键的方式来实现一对一关系，下面这个图说明了这种情况。下面开始介绍一下Kodo EJB中和一对一关系实现相关的知识，为了说明的需要，我们首先定义一个虚拟的场景。模拟场景 我们假定要完成一个图书馆管理系统，该系统中需要管理很多书，我们需要记录书的基本信息如编号、书名、出版日期等基本信息，还需要记录书的前言，序等信息。假设我们根据上面的需求，将书设计成一个类（Book），包括了书的编号和名称两个属性，同时将书的前言信息设计成另外一个类（BookExtend），它包括了书的编号和前言信息两个属性。由于一本书有前言而且也不可能其他的书前言部分会和他一样，所以类Book和BookExtend之间很自然的形成了一对一的关系。这两个类的属性以及类之间的关系如下图所示。 [注] 1、为了说明的简

单，例子设计时每个对象仅仅选择了必要的属性。2、上面的设计仅仅为了演示的要求而特意采用，不代表设计合理。

Kodo EJB中和一对一关系实现相关的内容在Kodo EJB中，我们可以使用简单的OneToOne注释来声明类和类之间的一对一关系，另外可选的，我们可以使用JoinColumn注释来声明两个类对应的表之间使用什么字段来进行关联。OneToOne

OneToOne注释提供了5个属性供开发者定义类和类之间一对一关系的细节内容。

- targetEntityClass类型的属性。定义关系类的类型，默认是该成员属性对应的类类型，所以通常不需要提供定义。
- mappedByString类型的属性。定义类之间的双向关系。如果类之间是单向关系，不需要提供定义，如果类和类之间形成双向关系，我们就需要使用这个属性进行定义，否则可能引起数据一致性的问题。比如上面的演示场景中，我们只是定义Book类有BookExtend属性，而BookExtend并没有Book属性，那么他们是单向关系，如何BookExtend中也定义了Book属性，那么Book和BookExtend之间就构成了双向关系。
- cascadeCascadeType[]类型。该属性定义类和类之间的级联关系。定义的级联关系将被容器视为对当前类对象及其关联类对象采取相同的操作，而且这种关系是递归调用的。举个例子：Book和BookExtend有级联关系，那么删除Book时将同时删除它所对应的BookExtend对象。而如果BookExtend还和其他的对象之间有级联关系，那么这样的操作会一直递归执行下去。

cascade的值只能从CascadeType.PERSIST（级联新建）、CascadeType.REMOVE（级联删除）、CascadeType.REFRESH（级联刷新）、CascadeType.MERGE（级联更新）中选择一个或多个。还有一个选择是使

用CascadeType.ALL，表示选择全部四项。fetchFetchType类型的属性。可选择项包括：FetchType.EAGER和FetchType.LAZY。前者表示关系类在主类加载的时候同时加载，后者表示关系类在被访问时才加载。默认值是FetchType.EAGER。

optionalboolean类型的属性。定义该关联类对是否必须存在。如果设置为false，那么该属性就不能设置为null。默认值是true。

OneToOne用法举例

```
public class Book{ // 其他内容...
@OneToOne(optional=true,cascade=CascadeType.ALL) public
BookExtend bookExtend. }
```

JoinColumn JoinColumn注释用于定义主类在数据库中对应的表通过什么字段和关系类的主键进行关联，这个注释是可选的，如果不提供该注释，Kodo在使用“对象名_ID”和关联表进行关联（简单情况下），比如演示场景中类Book的bookExtend没有使用JoinColumn注释进行声明，我们使用Kodo EJB提供的Mapping Tool工具生成表格的时候，Book类对应的表Book中将自动加入列bookExtend_ID，它的类型将和BookExtend对应表的主键字段id类型保持一致。

JoinColumn注释中有两个属性：name和referencedColumnName属性。nameString类型。它用于指定主类对应的表中与关系类的主键进行关联的字的名称，比如上例中，我们不希望使用默认的bookExtend_ID字段名进行关联，我们可以在Book类中使用JoinColumn注释bookExtend属性，设置JoinColumn注释为自己想要的名字比如extendID或者其他。referencedColumnNameString类型。指定关联表中与主表形成关联关系的字段名。主要用于设置区别于主键字段的情况。比如BookExtend表中默认使用Id进行关联，但现在需要使用其他字段进行关联，我们就可以提供该属性。

JoinColumn用法举例 public class Book{ // 其他内容...
@OneToOne(optional=true,cascade=CascadeType.ALL)
@JoinColumn(name="extendID",referencedColumnName="ID")
public BookExtend bookExtend. } 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