ApacheOpenJPA开发EJB3.0应用 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/270/2021_2022_ApacheOpen_c67_270960.htm 对象和对象之间除了继承关系之外，还存在着关联关系：包括一对一、一对多、多对一和多对多关系，在 OpenJPA 框架下，开发者只需要使用 javax.persistence.OneToOne 这样的注释，并提供相应的配置内容，就可以轻松的实现实体之间的关联关系，并且能够实现实体的级联创建、更新和删除。 本文中我们将以实体之间的一对一关联关系为例，深入地讲述如何使用 OpenJPA 框架提供的注释，实现企业应用中实体之间的关联关系。文中将提供一个简单的例子，详细的说明如何定义类和类之间的一对一关联关系的步骤，同时会重点讲述这些注释所支持的属性。一对多、多对一和多对多这三种关联关系在 OpenJPA 中的实现过程和一对一关联关系的实现过程是一致的，只是需要选择使用不同的注释，在本文的最后，会对实现这三种关联关系进行简单说明，读者可以参考一对一关系的实现过程来实现一对多、多对一和多对多的关联关系。 一对一关系 在面向对象的世界里，类 A 和类 B 之间形成一对一关系必须满足如下条件： 1. 对象 A1 引用了对象 B1； 2. 类 A 的其它对象 An 不能引用同样的对象 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的情况。 图 1. 关系数据库中的一对一关系 下面开始介绍 OpenJPA 中实现实体之间一对一关联关系的相关知识，为了说明的需要，我们首先定义一个简单的应用场景。 模拟场景 假定开发者要完成一个图书馆管理系统，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。 为了说明实体之间的一对一关系，我们将书设计成一个类（Book），包括书的编号和名称两个属性，同时将书的前言设计成另外一个类（BookExtend），它包括书的编号和前言两个属性。由于一本书有前言而且也不可能有其它书的前言部分会和它一样，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。 图 2. 类之间的一对一关系 [注]：为了说明的简单，本例子设计时每个对象只选择了必要的属性。 描述一对一关系 在 OpenJPA 中，开发者用来描述实体之间一对一关系时可选择的注释包括 javax.persistence.OneToOne 和 javax.persistence.JoinColumn.其中 javax.persistence.OneToOne 注释是必须使用的，它被用来声明类和类之间存在着一对一关系，javax.persistence.JoinColumn 注释是可选的，开发者使用 JoinColumn 注释来声明两个类在数据库中对应的表之间关联时的细节，包括主表中关联字段的名称、从表中使用什么字段来进行关联等。 javax.persistence.OneToOne javax.persistence.OneToOne 注释支持如下 5 个属性，它们可以被开发者用来定义实体和实体之间一对一关联关系的细节内容。 target Entity targetEntity 属性是 Class 类型的属性。定义实体一对一关系中处于从属地位的实体类的类型。如果没有为该属性设置值，OpenJPA 容器默认 targetEntity 属性的值是该成员属性对应的类类型，所以实体关系定义时通常不需要为 targetEntity 属性设置值。 mappedBy mappedBy 属性是 String 类型的属性。mappedBy 属性的值是当前实体在关联实体中的属性名称，使用 mappedBy

可以定义实体类之间的双向关系。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。 以演示场景中 Book 和 BookExtend 实体为例，假设我们只定义 Book 类有 BookExtend 类型的属性，而 BookExtend 并没有 Book 类型的属性，那么说明 Book 和 BookExtend 实体之间是单向关系；如果 BookExtend 中也定义了 Book 属性，那么 Book 和 BookExtend 实体之间就构成了双向关系。 