

# 一、JPA

## A.基本概念

JPA: java persistence api

JPA1.0 整合查询语言（Query）和对象关系映射（ORM）元数据定义

JPA2.0 在1.0的基础上，增加Criteria查询、元数据API以及校验支持

09年JPA2.0最终规范

06年JPA1.0最终规范 EJB 3.0的子规范

### 实体（Entities）：轻量级持久化域（Domain）对象

实体类（Entity Class）:实体类可以利用辅助类或者用于表示状态

约束：

- (1) 实体类必须使用@Entity标注或者XML描述
- (2) 实体类至少包含一个默认构造器，并且构造器必须是public或者protected
- (3) 实体类必须是顶级类，不能是枚举或者接口
- (4) 实体类禁止是final类
- (5) 实体支持继承、多态关联以及多态查询

实体持久字段和属性

实体持久状态由字段（Field）或者属性（Prooerties），字段即实例的属性或变量，属性则是JavaBeans实例的setter和getter方法。

实例字段的访问性必须是private、protected或者包可见，属性的可见性必须是public或者protected。

字段和属性访问类型（Access Type）

默认访问类型

非transient或者@Transient字段

非@Transient属性

显示访问类型

## 注解类型

- 实体类

- 映射超类

- 嵌套类

## 注解

- `@Access(AccessType.FIELD)` 字段

- `@Access(AccessType.PROPERTY)` 属性

## 实体主键

每个实体必须存在主键，主键必须定义在实体类

### 简单主键

- `@Id`

### 复合主键

- `@EmbeddedId`

- `@IdClass`

## 实体关系

实体关系可能是一对一、一对多、多对一或多对多，这些关系是多态性的，可以是单向或者双向。

### 注解表达方式

- `@OneToOne`

- `@OneToMany`

- `@ManyToMany`

- `@ManyToMany`

## 实体双向关系

实体双向关系是指两实体之间不仅存在拥有方（owning），也存在倒转方（inverse）。主方决定了更新级联关系到数据库。

### 规则

倒转必须通过`@OneToOne`、`@OneToMany`或者`@ManyToMany`中的`mappedBy`属性方法关联到拥有方的字段或者属性。

一对多、多对一双向关系中的多方必须是主方，因此`@ManyToOne`注解不能指定`mappedBy`属性方法。

双向一对一关系中，主方相当于包含外键的一方。

双向多对多关系中，任何一方可能是拥有方。

一对一的关系中在编码应该注意的事情：

(1) 拥有方的选择 怎么去选择主从的关系 mappedby的值是在对方实体中定义的字段名称 拥有方不用写mappedby

(2) 映射关系是有级联关系的

多对一的关系在编码中应该注意的事情：

(1) @ManyToOne的注解 和 @OneToMany的注解

多对多的关系应该注意的事情：

(1) 需要中间表进行关联 拥有方\_被关联方 中间表名称。

实体继承在jpa中的应用

@Inheritance 表示继承注解 多用在继承抽象实体类

继承已映射父类型：

@MappedSuperclass

@AssociationOverride

继承非实体类型

## B.JPA的服务化调用

### 1.jpa的服务调用

调用JPA的服务，需要去使用：

@PersistenceContext

private EntityManager entityManager;

持久化上下文使用期限（Persistence Context Lifetime）

这两种都得显示的设置出来事务的类型，到底是默认的，还是拓展的容器去实现的。这里要去使用的话必须要配合容器的@Transactional注解来提供给

JPA事务的类型和控制。这里启动类也要记得去加上

@EnableTransactionManagement注解开启事务注解

类型

事务类型（默认）

拓展类型

阶段

事务提交阶段

事务类型：实体状态 -> 脱管

拓展类型：实体状态 -> 继续维持

事务回滚阶段

实体状态 -> 脱管

当有对应关系的时候，如果传入参数有级联对象，要设置CascadeType级联的关系。否则会报错。

## 2.实体持久化Event监听机制

Event这种模式的一个理解：

(1) EventListener是一个标记型接口，所有的Listener接口都要继承这个接口。比如ServletContextListener就继承了这个接口，在其中定义了init和destory的方法。

(2) EventObject是一个所有的Event接口都要去继承的类，比如ServletContextEvent这个类，就是继承了EventObject，这里会把ServletContext作为source字段传入到其中，提供了servletContext对象的访问和设置。

这样其实自定义的一些Listener就是实现了ServletContextListener接口，可以实现了在容器初始化和销毁时要做的事情。

JPA中的@EntityListeners(value="") 这其中的value就是传入你做处理的Listener。可以在其中定义持久化之前要做的，持久化之后要做的。或者是remove时要做的，update时要做的。这个注解是要在实体类上进行使用的。

方法注解：

@Prepersist

@PostPersist

## 二、Spring Data JPA

spring data jpa其实是帮我们去实现了上面一些复杂操作，提供的是repository的方式

```
interface Repository<T, ID extends Serializable>
```

这里ID必须要实现序列化接口，必须是序列化的

还提供了 PageAndSort Query等一系列的Repository。

客户端去使用的时候，只要去实现吗SimpleJpaRepository类即可。

这里可以去使用Repository相关的接口去实现数据库的操作。