# Hibernate常用的注解

注意：JPA的注解可以写在getter方法上和属性上

## 1、@Entity

标记当前的类是映射的表

## 2、@Table

标记当前类对应的表信息

常用属性：

name:表名

## 3、@Id

标记当前属性为主键

## 4、@GeneratedValue

主键的生成策略

常用属性：

1、strategy：整型的生成策略

取值：

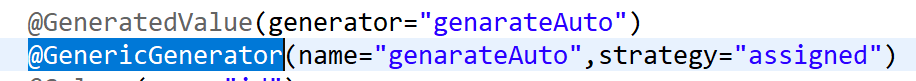
GenerationType.IDENTITY：MySQL的数据库的自增策略

GenerationType.SEQUENCE：Oracle的序列

GenerationType.AUTO:根据数据库自动选择，会创建表来维护数据的增长

GenerationType.TABLE:创建一个表，自己来维护

2、@GenericGenerator 是hibernate对jpa主键生成策略注解的扩展



## 5、@Column

设置当前属性对应的字段信息

常用属性：

name:设置属性对应的字段名称

length:设置字段的长度

nullable：是否允许为null

unique:是否为唯一约束

columnDefinition:手动标记数据库中字段类型，默认也会转换

|  |
| --- |
| columnDefinition:表示该字段在数据库中的实际类型 通常ORM框架可以根据属性类型自动判断数据库中字段的类型, 但是对于Date类型仍无法确定数据库中字段类型究竟是DATE,TIME,TIMESTAMP还是DATETIME, 此外,String的默认映射类型为VARCHAR,如果要将String类型映射到特定数据库的BLOB或TEXT字段类型,该属性非常有用 如: @Column(name="BIRTH",nullable="false",columnDefinition="DATE")  insertable:默认情况下,JPA持续性提供程序假设所有列始终包含在 SQL INSERT 语句中。 如果该列不应包含在这些语句中，请将 insertable 设置为 false  updatable：列始终包含在 SQL UPDATE 语句中。如果该列不应包含在这些语句中，请将 updatable 设置为 false |

注：如果属性没有写@Column该注解，那么默认拥有@Basic注解

@Basic  
@javax.persistence.Basic(fetch=FetchType.LAZY,optional=true)  
fetch:抓取策略,延时加载与立即加载  
optional:指定在生成数据库结构时字段是否允许为 null

## 6、@Transient

标记该属性不是数据库表中字段

## 7、@JoinColumn

多表关系中的外键字段

常用属性：

name:外键名称

多表关系中的主键的通用属性：

1、fetch：抓取策略，可以标记勤加载或懒加载

取值：

FetchType.EAGER:勤加载

FetchType.LAZY:懒加载，如果是集合那么就是懒加载

2、cascade：级联设置，在有外键的一方设置，新增、修改、删除

取值说明：

CascadeType.ALL:全部

CascadeType.DETACH:挂载

CascadeType.MERGE:修改 merge() /update()

CascadeType.PERSIST:保存 persist()/ save()

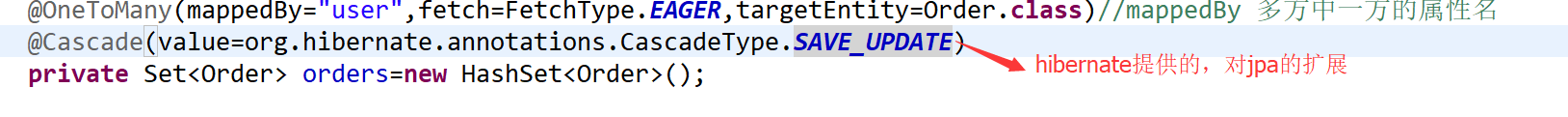
CascadeType.REFRESH:刷新

CascadeType.REMOVE:删除

|  |
| --- |
|  CascadeType.REMOVE Cascade remove operation，级联删除操作。 删除当前实体时，与它有映射关系的实体也会跟着被删除。   CascadeType.MERGE Cascade merge operation，级联更新（合并）操作   CascadeType.DETACH Cascade detach operation，级联脱管/游离操作。 如果你要删除一个实体，但是它有外键无法删除，你就需要这个级联权限了。它会撤销所有相关的外键关联。   CascadeType.REFRESH Cascade refresh operation，级联刷新操作 不用。   CascadeType.ALL Cascade all operations，清晰明确，拥有以上所有级联操作权限。 |

3、targetEntity:目标类的Class类型

注意：



|  |
| --- |
| **public** **enum** CascadeType {  /\*\*  \* Includes all types listed here.  \*/  ***ALL***,  /\*\*  \* Corresponds to {@link javax.persistence.CascadeType#PERSIST}.  \*/  ***PERSIST***,  /\*\*  \* Corresponds to {@link javax.persistence.CascadeType#MERGE}.  \*/  ***MERGE***,  /\*\*  \* Corresponds to {@link javax.persistence.CascadeType#REMOVE}.  \*/  ***REMOVE***,  /\*\*  \* Corresponds to {@link javax.persistence.CascadeType#REFRESH}.  \*/  ***REFRESH***,  /\*\*  \* Corresponds to the Hibernate native DELETE action.  \*/  ***DELETE***,  /\*\*  \* Corresponds to the Hibernate native SAVE\_UPDATE (direct reattachment) action.  \*/  ***SAVE\_UPDATE***,  /\*\*  \* Corresponds to the Hibernate native REPLICATE action.  \*/  ***REPLICATE***,  /\*\*  \* Hibernate originally handled orphan removal as a specialized cascade.  \*  \* **@deprecated** use **@OneToOne**(orphanRemoval=true) or **@OneToMany**(orphanRemoval=true)  \*/  @Deprecated  ~~DELETE\_ORPHAN~~,  /\*\*  \* Corresponds to the Hibernate native LOCK action.  \*/  ***LOCK***,  /\*\*  \* JPA originally planned on calling DETACH EVICT.  \*  \* **@deprecated** use javax.persistence.CascadeType.DETACH  \*/  @Deprecated  ~~EVICT~~,  /\*\*  \* Corresponds to {@link javax.persistence.CascadeType#REFRESH}.  \*/  ***DETACH***  } |

## 8、@OneToOne

体现一对一关系映射

一般要结合@JoinColumn标记外键名称

## 9、@ManyToOne

体现的是多对一

一般要结合@JoinColumn标记外键名称

## 10、@OneToMany

体现一对多的映射，不会出现外键

常用属性：

mappedBy：关联的属性名称，就是存在多的一方中的本类的属性名称

targetEntity:集合中泛型类的Class类对象，其实就是多的类的Class

## 11、@ManyToMany

体现的多对多，隐式实现

需要配合的@JoinTable:关系表

## 12、@JoinTable

多对多的时候的关系表

常用属性：

name:关系表名

joinColumns:外键字段名称，当前类在关系表中的外键名称，得使用注解@JoinColumns

inverseJoinColumns:存在多对多关系的另一方在关系表中的外键名称