



链滴

关于 hibernate 中 @Transient 注解使用的一点心得

作者: [liuyishi](#)

原文链接: <https://ld246.com/article/1517627283507>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

@Transient 表示该属性并非一个到数据库表的字段的映射,ORM 框架将忽略该属性.

如果一个属性并非数据库表的字段映射,就务必将其标示为 **@Transient**,否则, ORM 框架默认其注解为 **Basic**

// 根据birth计算出age属性

```
@Table(name="person")
```

```
public class Person{
```

```
    private Date birth;
```

```
    @Column(name = "birth")
```

```
    public Date getBirth() {
```

```
        return birth;
```

```
    }
```

```
    public void setBirth(Date birth) {
```

```
        this.birth = birth;
```

```
    }
```

```
    @Transient
```

```
    public int getAge(){
```

```
        return getYear(new Date())-getYear(getBirth()); //getYear为类中的自定义方法, 返回传入
```

```
        的年份
```

```
    }
```

```
}
```

比如上面例子在一个持久化类 person 中定义了 birth 的属性并且用了 @Column 注解在数据表中进行了映射, 而 getAge() 方法注解为 @Transient, 表明在数据库表中并没有 age 这个映射字段。

当你通过字段映射从数据库获取数据时, 因为getAge()加了@Transient, 并且在数据库中没有字段所以getAge()没有从数据库中获取数据, 而是通过getBirth()方法获取数据库中返回的birth, 并通过算返回getAge()的值。

```
return getYear(new Date())-getYear(getBirth());
```

当你在程序中获取到持久化类person的对象时, person.getAge()和person.getBirth()都有数据返回供给你使用。

这样做的话, 能减少数据库的字段,比如这个例子中就不需要新建一个关于 age 的数据库字段了, 通过 birth 字段和计算就能获取到 person 的 age 属性。@Transient 还可以在持久化类中直接获取关联中的字段值, 许多业务逻辑中都有很大的作用。