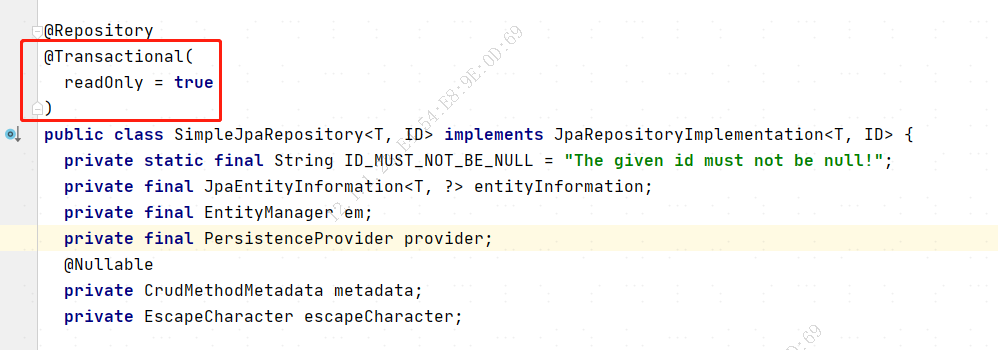
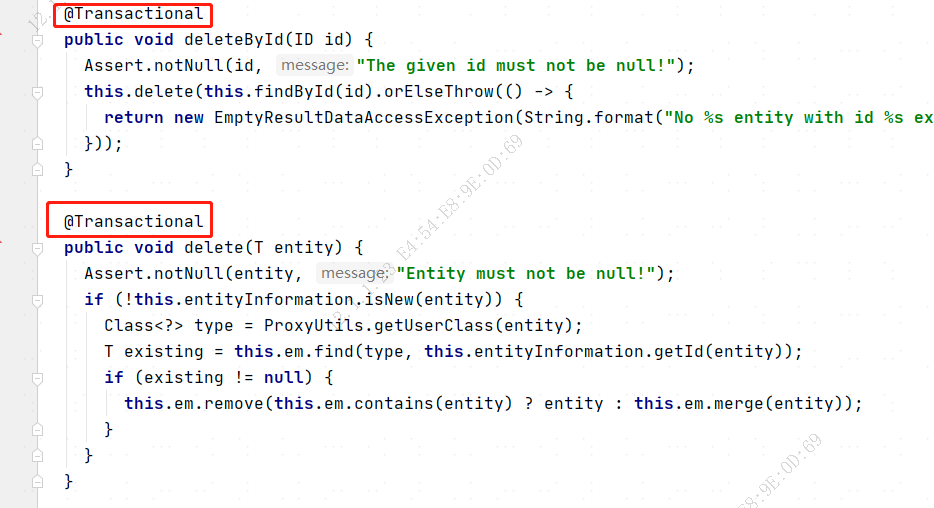
## JPA事务

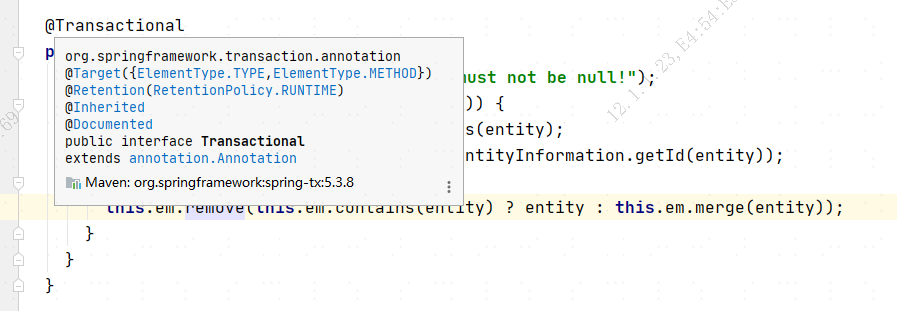
1.jpa自带的save等等方法其实是自带@transactional的，如下：



默认是readOnly=true，是因为在findOne/findById时未添加任何的事务注解，让这两个方法使用类的默认注解。若其他方法需要使用save/update/delete，则在方法上单独再加注解,如下：



2.其实jpa的事务使用的就是spring框架提供的@transactional事务注解，所以spring事务注解的机制是适用于JPA的



3.Spring事务：默认是rollBack runtimeException的，如果不是runtimeException需要在事务注解中声明，否则不会rollback，且异常抛出的地方需要在事务之内

@transactional中的常用属性：

Propagation：规定事务的传播行为

rollbackFor/noRollBackFor：规定回滚/不回滚的异常类型

Isolation：事务隔离级别

Value：规定事务处理器（一般多事务的时候）

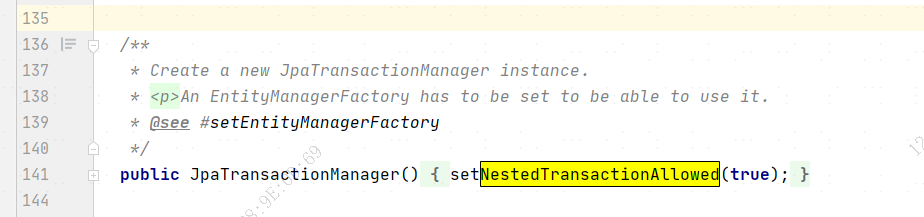
readOnly：规定事务只读

4.事务的传播行为Propagation：

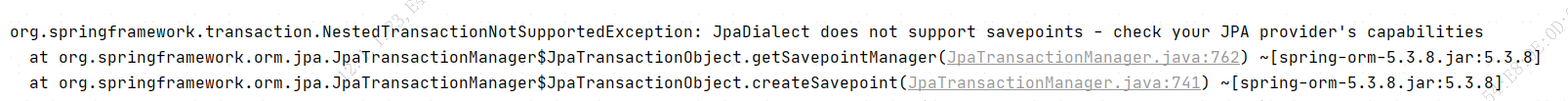
Propagation.REQUIRED：如果有事务, 那么加入事务, 没有的话新建一个(默认情况下)  
Propagation.NOT\_SUPPORTED：容器不为这个方法开启事务  
Propagation.REQUIRES\_NEW：不管是否存在事务,都创建一个新的事务,原来的挂起,新的执行完毕,继续执行老的事务  
Propagation.MANDATORY：必须在一个已有的事务中执行,否则抛出异常  
Propagation.NEVER：必须在一个没有的事务中执行,否则抛出异常(与Propagation.MANDATORY相反)  
Propagation.SUPPORTS：如果其他bean调用这个方法,在其他bean中声明事务,那就用事务.如果其他bean没有声明事务,那就不用事务.

Propagation.NESTED:如果一个活动的事务存在，则运行在一个嵌套的事务中。如果没有活动事务，则按REQUIRED属性执行。Spring首先检查transaction是否存在，如果存在则创建一个savepoint，如果我们的程序抛出异常的时候，transaction将会回滚到该savepoint。如果没有transaction，NESTED的表现和REQUIRED一样。

但注意，虽然jpaEntityManager中有定义nestedTransactionAllowed为true，



但事实上jpaEntityManager是不支持nested transaction的，hibernate也不支持。

要测试的话只能使用原生的jdbc测试。此处默认true也只是让原生jdbc支持nested事务，但若本身并没有用到原生jdbc则还是没用

事物超时设置:

　　@Transactional(timeout=30) //默认是永不超时