# **[spring事务什么时候会自动回滚](https://www.cnblogs.com/panchanggui/p/11038282.html)**

在java中异常的基类为Throwable，他有两个子类xception与Errors。同时RuntimeException就是Exception的子类，只有RuntimeException才会进行回滚；

1，Spring事务回滚机制是这样的：当所拦截的方法有指定异常抛出，事务才会自动进行回滚！

      我们需要注意的地方有四点： 如果你在开发当中引入Spring进行事务管理，但是事务没能正常的自动回滚，可以对照下面四点，缺一不可！

         ①被拦截方法-—— 注解式：方法或者方法所在类被@Transactional注解；

         ②异常—— 该方法的执行过程必须出现异常，这样事务管理器才能被触发，并对此做出处理；

        ③指定异常——  默认配置下，事务只会对Error与RuntimeException及其子类这些UNChecked异常，做出回滚。 一般的Exception这些Checked异常不会发生回滚（如果一般Exception想回滚要做出配置）；

                    举例：实现一般异常的回滚：

                     注解式：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | @Transactional(rollbackFor=Exception.class)    @RequestMapping("consump")    public void insertConsumption(Customer c,Consumption s,Mygift m) throws Exception{         } |

         ④异常抛出—— 即方法中出现的指定异常，只有在被事务管理器捕捉到以后，事务才会据此进行事务回滚；

                       1，不捕捉，会回滚：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public void insertConsumption(Customer c,Consumption s,Mygift m) throws Exception{          int a=consumpDao.insert(s);         int b=customerDao.update(customer);    } |

                       2，如果异常被try{}捕捉到，那么事务管理器就无法再捕捉异常，所以就无法做出反应，事务不回滚；

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public void insertConsumption(Customer c,Consumption s,Mygift m) throws Exception{       try{              }catch(RuntimeException){          }    } |

                        3，如果异常被try{}捕捉了，我们还可以在Catch(){}中throw   new  RuntimeException()，手动抛出运行时异常供事务管理器捕捉；

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | public void insertConsumption(Customer c,Consumption s,Mygift m) throws Exception{         try{               }catch(){             throw new RuntimeException();            }    } |

2，在实际开发中，有时并没有异常发生，但是由于事务结果未满足具体业务需求，所以我们不得不手动回滚事务！

        有如下两种方法：

                    ①手动抛出异常（如果你没有配置一般异常事务回滚，请抛出运行时异常）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | if(){    }else{        throw new RuntimeException();    } |

|  |  |
| --- | --- |
| 1 | ②编程式实现手动回滚 |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | if(){    }else{       TransactionAspectSupport.currentTransactionStatus().setRollbackOnly();     } |

转自：https://my.oschina.net/gaomq/blog/1580290?email=Xie.Chaoyue@outlook.com