### 走一遍

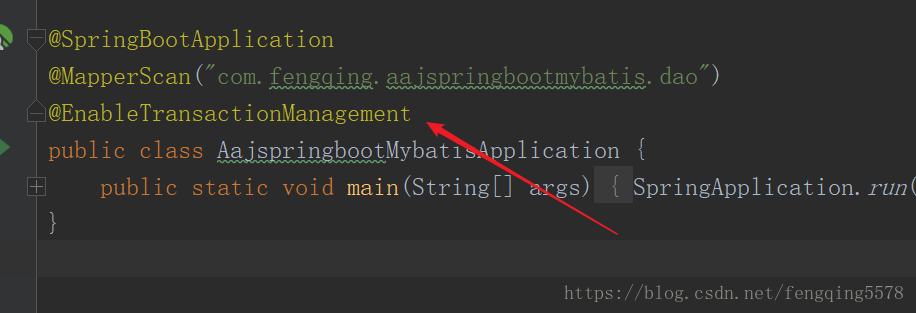
systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

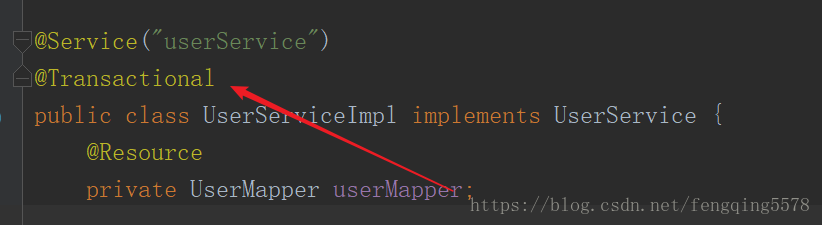
firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

# 在springboot中添加事务

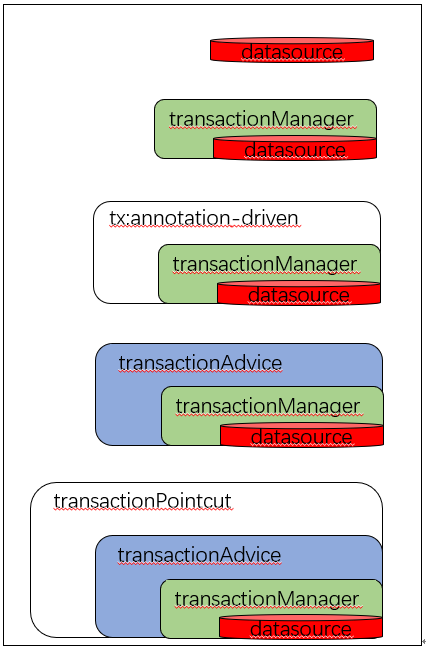
在启动类上添加注解@EnableTransactionManagement

s

在业务逻辑类上添加注解@Transactional



# 在ssm项目中添加事务



在网上找了spring事务的配置

<!-- 配置事务管理器 -->  
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  
 <property name="dataSource" ref="dataSource" />  
</bean>

<!-- 注解方式配置事物 -->  
<tx:annotation-driven transaction-manager="transactionManager" />

<!-- 拦截器方式配置事物 -->

<tx:advice id="transactionAdvice" transaction-manager="transactionManager">  
<tx:attributes>  
 <tx:method name="add\*" propagation="REQUIRED" />  
 <tx:method name="append\*" propagation="REQUIRED" />  
 <tx:method name="insert\*" propagation="REQUIRED" />  
 <tx:method name="save\*" propagation="REQUIRED" />  
 <tx:method name="update\*" propagation="REQUIRED" />  
 <tx:method name="modify\*" propagation="REQUIRED" />  
 <tx:method name="edit\*" propagation="REQUIRED" />  
 <tx:method name="delete\*" propagation="REQUIRED" />  
 <tx:method name="remove\*" propagation="REQUIRED" />  
 <tx:method name="repair" propagation="REQUIRED" />  
 <tx:method name="delAndRepair" propagation="REQUIRED" />

<tx:method name="get\*" propagation="SUPPORTS" />  
 <tx:method name="find\*" propagation="SUPPORTS" />  
 <tx:method name="load\*" propagation="SUPPORTS" />  
 <tx:method name="search\*" propagation="SUPPORTS" />  
 <tx:method name="datagrid\*" propagation="SUPPORTS" />

<tx:method name="\*" propagation="SUPPORTS" />  
 </tx:attributes>  
</tx:advice>

<!-- 这个配置很重要，切面的切入位置-->  
<aop:pointcut id="transactionPointcut" expression="execution(\* com.heb.user.service..\*Impl.\*(..))" />  
 <aop:advisor pointcut-ref="transactionPointcut" advice-ref="transactionAdvice" />

</aop:config>

service层也是运用的@Transactional注解，直接加在方法上，只对此方法生效

需要事物的具体操作需要try｛｝catch（）｛｝

具体例子

try{  
i = numeralLoginDao.insertInToResellAll(resellCar);//操作1  
j = numeralLoginDao.deleteTemporaryResell(resellCar.getPhonenumber());//操作2  
if(i==0||j==0){  
 throw new Exception();  
}

}catch(Exception e){  
System.out.println("数据操作错误");

}

这样只要发生定义的和未知的错误都会回滚！