1 Spring中 纯XML 配置事务

```
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="pooledDataSource"/>
</bean>
<aop:config>
    <aop:pointcut expression="execution(* cn.yuanyu.crud.service..*(..))"
id="txPoint"/>
    <aop:advisor advice-ref="txAdvice" pointcut-ref="txPoint"/>
</aop:config>
<tx:advice id="txAdvice" transaction-manager="transactionManager">
    <tx:attributes>
        <tx:method name="*"/>
        <tx:method name="get*" read-only="true"/>
    </tx:attributes>
</tx:advice>
```

2 Spring中 XML + 注解 配置事务

一般xml里面配置粗粒度的控制，然后使用注解

```
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="pooledDataSource"/>
</bean>
<!-- 定义切入点，expression为切人点表达式，如下是指定service包下的所有方法，具体
以自身实际要求自定义  -->
<aop:config>
    <aop:pointcut expression="execution(* cn.yuanyu.crud.service..*(..))"
id="txPoint"/>
    <aop:advisor advice-ref="txAdvice" pointcut-ref="txPoint"/>
</aop:config>
<!-- <tx:advice>定义事务通知，用于指定事务属性，其中“transaction-manager”属性
指定事务管理器，并通过<tx:attributes>指定具体需要拦截的方法-->
<tx:advice id="txAdvice" transaction-manager="transactionManager">
    <tx:attributes>
        <!--<tx:method>拦截方法，其中参数有:
            name:方法名称，将匹配的方法注入事务管理，可用通配符
            propagation：事务传播行为，
            isolation：事务隔离级别定义；默认为“DEFAULT”
            timeout：事务超时时间设置，单位为秒，默认-1，表示事务超时将依赖于底层事
务系统;
            read-only：事务只读设置，默认为false，表示不是只读;
            rollback-for：需要触发回滚的异常定义，可定义多个，以“，”分割，默认
RuntimeException都将导致事务回滚;
            no-rollback-for：不被触发进行回滚的 Exception(s)；可定义多个，以“，”分
割;
```

```
        -->
        <tx:method name="*"/>
        <tx:method name="get*" read-only="true"/>
    </tx:attributes>
</tx:advice>

<tx:annotation-driven transaction-manager="transactionManager"/>//开启事务注解
```

3 Spring中 纯注解 配置事务

```
@Configuration //声明配置类
@MapperScan("cn.yuanyu.tx.mapper")
@EnableTransactionManagement // 开启事务注解，等同于配置文件<tx:annotation-driven/>
public class MybatisPlusConfiguration {

@Transactional()
public void registerUser(User user) throws Exception {
    ...
}
```