Atitit 信息安全 线程安全 内存安全 连接安全

Spring把JDBC 的 Connection或者Hibernate的Session等访问数据库的链接（会话）都统一称为资源，显然我们知道Connection这种是线程不安全的，同一时刻是****不能****被多个线程共享的。

简单的说：同一时刻我们每个线程持有的Connection应该是独立的，且都是互不干扰和互不相同的

****但是Spring管理的Service、Dao等他们都是无状态的单例Bean，怎么破？****，如何保证单例Bean里面使用的Connection都能够独立呢？ Spring引入了一个类：事务同步管理类org.springframework.transaction.support.TransactionSynchronizationManager来解决这个问题。它的做法是内部使用了很多的ThreadLocal为不同的事务线程提供了独立的资源副本，****并同时维护这些事务的配置属性和运行状态信息 （比如强大的事务嵌套、传播属性和这个强相关）。****

这个同步管理器TransactionSynchronizationManager是掌管这一切的大脑，它管理的TransactionSynchronization是开放给调用者一个非常重要的扩展点，下面会有详细介绍~

TransactionSynchronizationManager 将 Dao、Service 类中影响线程安全的所有 “ 状态 ” 都统一抽取到该类中，并用 ThreadLocal 进行封装，这样一来， Dao （基于模板类或资源获取工具类创建的 Dao ）和 Service （采用 Spring 事务管理机制）就不用自己来保存一些事务状态了，****从而就变成了线程安全的单例对象了，优秀~****

###### **DataSourceUtils**