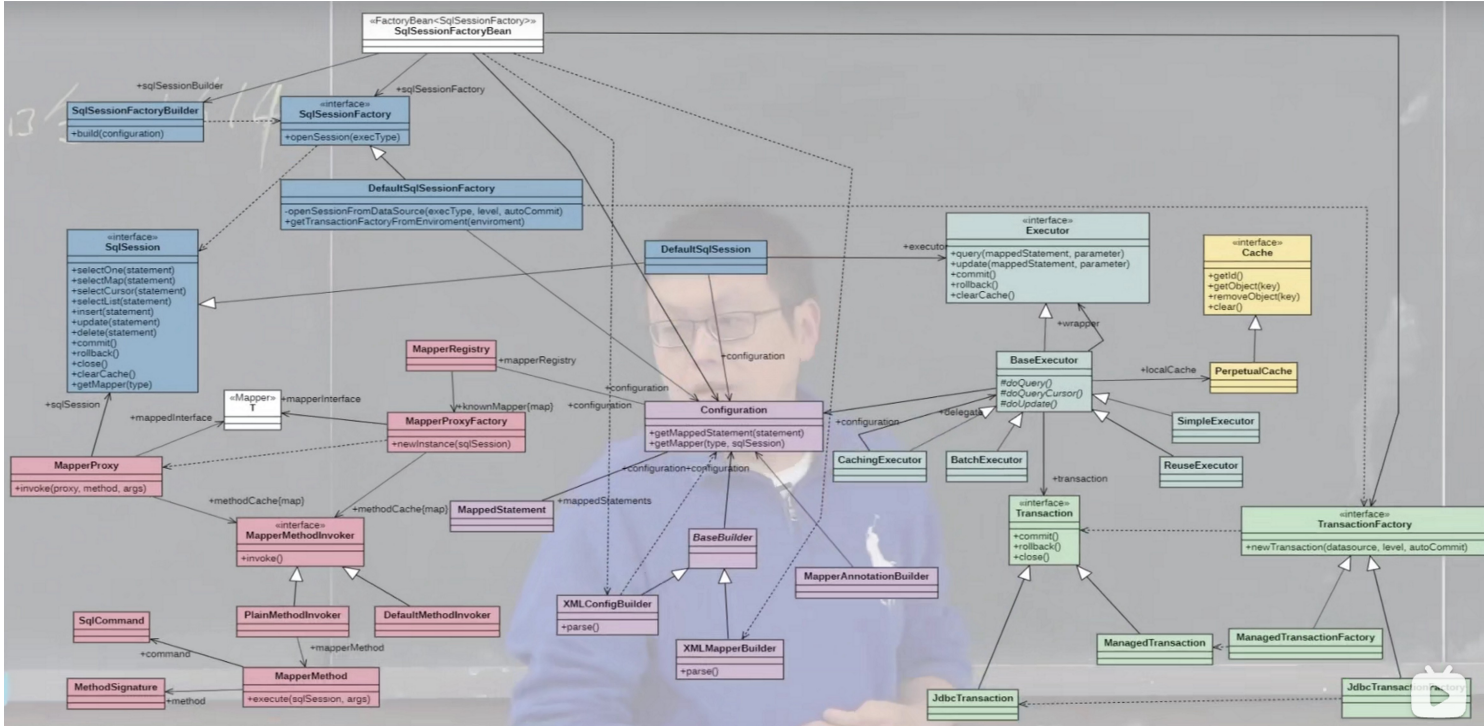


Transaction接口

2022年1月2日 0:23



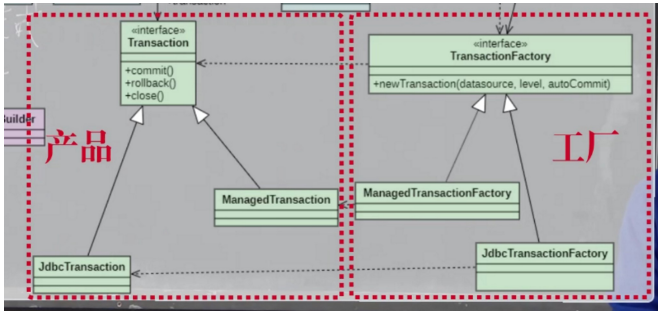
绿色的这一部分是和事务有关的

在使用MyBatis时都是使用的事务，即@Transaction，放在某个方法的前面，这样就会在这个方法开始执行时开始一个事务，方法结束后关闭这个事务。这种用注解的方式是Spring容器自动帮我们管理事务，叫做容器管理的事务。如果我们不想让容器帮我们管理，即想自定义事务的开始或结束，可以使用SqlSession的对应的方法

所以说这里的事务主要分为两种，一种是Bean容器管理的事务，一种是自己写代码开启的事务，对应的是Transaction下的两个实现——JdbcTransaction和ManagedTransaction

这两种事务，不管是哪一种，我们都希望其他代码不会受到影响，所以我们在这里采用实现Transaction接口的方式，因此不管使用哪种事务是无关的

但构造这两种事务所需要的信息时不同的，比如构建容器级别的Transaction需要DataSource、容器上下文等。所以分别由ManagedTransactionFactory、JdbcTransactionFactory去构造这两种Transaction。这两个Factory分别实现了TransactionFactory。这样的设计模式称之为抽象工厂设计模式



在工厂设计模式中，可以分为产品和工厂两部分，由对应的工厂来生产对应的产品。产品一边存在继承关系，工厂一边存在继承关系，产品和工厂之间是一一对应的关系，这就是抽象工厂的定义。

使用抽象工厂是因为Transaction会在很多地方用到，但Transaction有两种，而且各自的构造方法也不一样，所以要把各自的构造方法分开。所以可以看到DefaultSessionFactory中关联的是一个TransactionFactory对象。所以往SqlSession中放什么样的Factory，就会构造出来什么样的Transaction。这是抽象工厂的一个用途，也体现了保护变化的原则