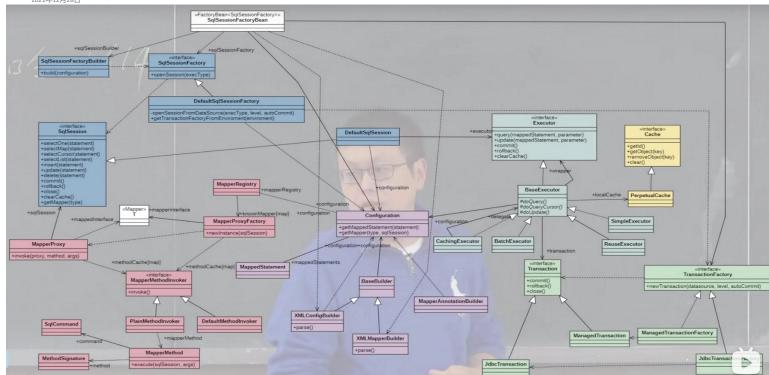
## Configuration **™** Mapper

2021年12月28日 12:45



紫色的这一部分是配置信息,配置信息是在MyBatis起来的时候就要读的

MyBatis中的配置信息有两部分:

 MyBatis本身的配置信息 使用Spring-boot时,这部分配置信息直接写在application.yaml里

 接口的信息 用独立的接口或者接口对应的xml文件去描述接口的配置信息

这两部分配置信息在Spring框架去加载MyBatis时,会全部读取,其中有出错就会有报错信息

我们可以看到在上面SqlSessionFactoryBean里有一个属性叫做configuration(一个箭头指下来,所有指向Configuration的属性名字都叫做configuration)

所以在SqlSessionFactoryBean中是有一个Configuration的对象的,这样在Spring框架运行起来时,会去读所有MyBatis的配置信息,然后把它变成Configuration对象

因为配置信息有两种不用的形式(XML和注解),所以可以看到下面的这几个Builder:

- XMLConfigBuilder 这个是早期的做法,用xml来写MyBatis的配置信息。如果我们想用独立的xml文件去配置MyBatis,那么就会用到这个 builder
- XMLMapperBuilder和MapperAnnotationBuilder 这个是Mapper接口的配置信息,前者是用xml文件配置,后者是用注解配置,所以就会有两个Builder去读这两种配置信息

不管是什么样的Builder,最后读进来的都是Configuration对象

配置信息读进来后,存储在Configuration对象中。Configuration对象中是有很多属性的,这里主要说Mapper的配置信息 我们看红色的一部分类,这部分就是Mapper的配置

Configuration中Mapper的配置信息是写在MapperRegistry的对象中的,所以MapperRegistry和 Configuration之间有一条双向的线。MapperRegistry中存的是所有Mapper的配置信息,以HashMap的方式来存。它里面有一个属性叫做konwMapper,是把它知道的Mapper都读进去,以Map的格式存起来,Map的key是Mapper接口类(不是对象),value是MapperProxyFactory

不把Mapper对象放进来的原因来在于,在配置里写的信息,并不代表某个对象真正会用到,所以没用到的就不必去实例化出一个对象。它是在注入时才把需要的Mapper对象注入进去的,如果没有用到,就不需要注入,也不需要实例化。所以Mapper的配置信息以Map的方式来存储,key是Mapper的类,value是MapperProxyFactory,即工厂对象,用这个工厂对象去制造我们要注入进去的Mapper对象

其余的红色类中,MapperProxy是在注入时真正要注入的东西。毕竟Mapper接口不是MyBatis包中的东西,是在外面定义的(白色类),所以不可能说直接做出一个Mapper对象出来。它会使用Java的动态代理机制,当我们调用Mapper的某个方法时,实际上调用的是MapperProxy的invoke方法,去模拟实现Mapper接口中的方法。剩下的红色的类是关于MapperProxy如何执行的类

所以,在MyBatis中,主要的面向对象特征就是表现在Mapper和Configuration这两部分中的,因为这两个部分包含了 Mapper的接口、传进去的参数是如何映射到Mapper定义中的Sql语句参数中的、返回值是如何变成我们所需要的对象的