Atitit tkmybatis 多数据源

创建SqlSessionFactory

通过重新上面的配置，初始化多个SqlSessionFactory，如果使用spring-boot，可以扩展ImportBeanDefinitionRegistrar来动态创建。

扫描Mapper

同样实现ImportBeanDefinitionRegistrar，通过ClassPathMapperScanner来扫描不同的包，不同的包使用不同的SqlSessionFactory

# **通用Mapper多数据源配置**

当在 SpringBoot 配置文件中创建多个数据源的配置时，若使用 Mybatis，mapper 是不能自动识别哪个 mapper 该基于哪个数据源连接的。这里我们的关键是需要****为 mapper 指定使用的数据源****。

根据我的调研，有几种指定的思路，例如使用 AOP，或者本文介绍的，****显式指定不同 mapper 使用不同的数据源****。

新建 Db1MybatisConfig.java，使用 @Configuration 指定为 Spring 的配置文件。接着，我们使用 @MapperScan 来指定，该份配置文件需要扫描哪些包（basePackages 参数），以及所扫描的包，使用哪份模板文件来与数据库操作（sqlSessionTemplateRef 参数）。

|  |  |
| --- | --- |
| 1 2 3 4 5 | **@Configuration** **@MapperScan**(basePackages = "com.example.demo.mapper.db1", sqlSessionTemplateRef = "db1SqlSessionTemplate") **public** **class** **Db1MybatisConfig** {  ... |

1. 对象；
2. 与 1 类似，使用 @ConfigurationProperties 获取 yml 中特定数据源的 Mybatis 配置信息，自动注入到 Configuration 对象；
3. 准备好上述数据后，进一步手动封装 SqlSessionFactory，再由其生成对应的 SqlSessionTemplate，并使用该数据源来进行事务管理 DataSourceTransactionManager。

至此我们便绑定了特定数据源到特定的 SqlSessionTemplate，进而实现了特定包目录下的 mapper 文件，通过对应绑定的 SqlSessionTemplate 来操作特定的数据源。

### **二、原由**

由于使用了tk.mybatis的包，默认自动全包扫描@Mapper注解。tk.mybatis扫描生成的Mapper比其他自定义配置的bean生成对应的Mapper对象快，而自定义的mybatis@MapperScan后扫描，Dao Bean已经生成了，无法再指定数据源进行注入了。