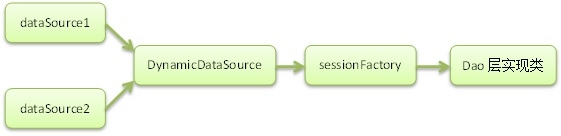
**多数据源配置**

## 1. 多数据源自动切换原理

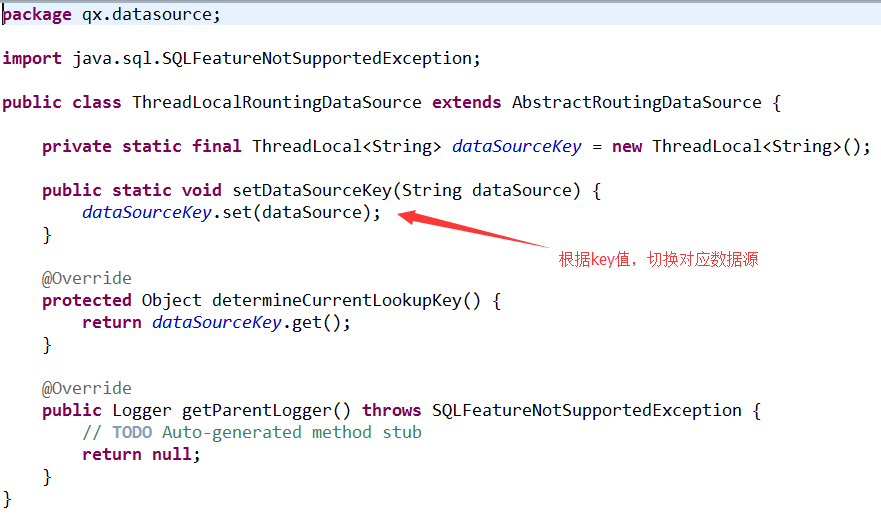
通过配置一个DynamicDataSource进行多数据源的动态切换



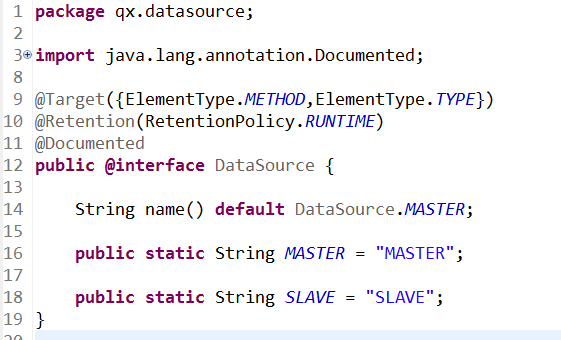
## 2.Spring+Mybatis配置多数据源

### 2.1扩展Spring的AbstractRoutingDataSource抽象类（该类充当了DataSource的路由中介, 能在运行时, 根据某种key值来动态切换到真正的DataSource上。）

ThreadLocalRountingDataSource类充当上述所说的DynamicDataSource，进行具体数据源的动态切换

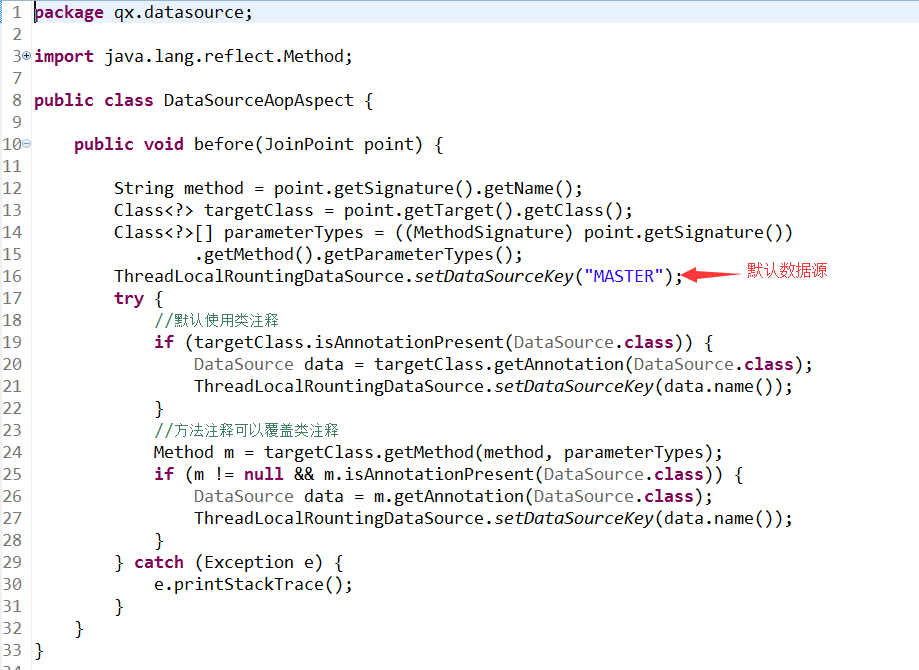


### 2.2配置切换数据源的注解标签，master为主库（可进行增删改操作），slave为从库（只读操作）



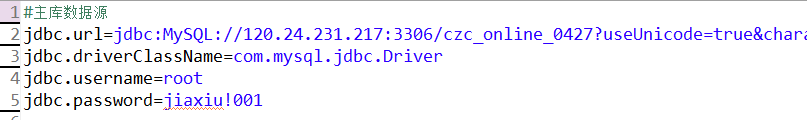
### 2.3切面实现DataSourceAopAspect

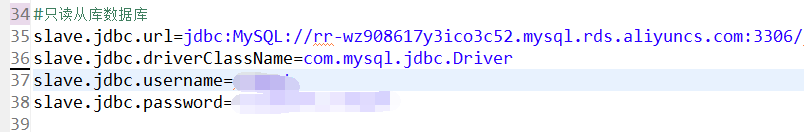
默认数据源为master，根据具体类和方法上的注解标签@DataSource来动态切换对应的数据源



### 2.4配置文件

1. Dbconfig.properties配置



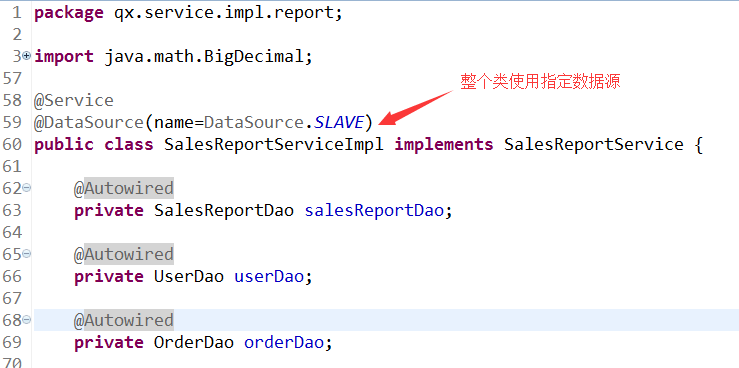


### 2.5 applicationContext.xml配置，注意标红

|  |
| --- |
| <!-- 主库数据源-->  <beanid=*"dataSource"*class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*>  <propertyname=*"driverClass"*value=*"${jdbc.driverClassName}"*/>  <propertyname=*"jdbcUrl"*value=*"${jdbc.url}"*/>  <propertyname=*"user"*value=*"${jdbc.username}"*/>  <propertyname=*"password"*value=*"${jdbc.password}"*/>  </bean>  <!-- 从库数据源-->  <beanid=*"slaveDataSource"*class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*>  <propertyname=*"driverClass"*value=*"${slave.jdbc.driverClassName}"*/>  <propertyname=*"jdbcUrl"*value=*"${slave.jdbc.url}"*/>  <propertyname=*"user"*value=*"${slave.jdbc.username}"*/>  <propertyname=*"password"*value=*"${slave.jdbc.password}"*/>  </bean>  <!-- 多数据源配置 -->  <beanid=*"****dataSourceOption****"*class=*"qx.datasource.ThreadLocalRountingDataSource"*>  <propertyname=*"defaultTargetDataSource"*ref=*"dataSource"*/>  <propertyname=*"targetDataSources"*>  <mapkey-type=*"java.lang.String"*>  <entrykey=*"MASTER"*value-ref=*"dataSource"*/>  <entrykey=*"SLAVE"*value-ref=*"slaveDataSource"*/>  <!-- 这里还可以加多个dataSource -->  </map>  </property>  </bean>  <beanid=*"sqlSessionFactory"*class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <propertyname=*"configLocation"*value=*"classpath:/conf/configuration.xml"*/>  <propertyname=*"dataSource"*ref=*"****dataSourceOption****"*/>  <propertyname=*"typeAliasesPackage"*value=*"qx.dao"*/>  </bean>  <beanid=*"transactionManager"*class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <propertyname=*"dataSource"*ref=*"****dataSourceOption****"*/>  </bean>  <beanclass=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <propertyname=*"basePackage"*value=*"qx.dao"*/>  <propertyname=*"sqlSessionFactoryBeanName"*value=*"sqlSessionFactory"*/>  </bean>  <beanid=*"jdbcTemplate"*  class=*"org.springframework.jdbc.core.JdbcTemplate"*>  <propertyname=*"dataSource"*ref=*"****dataSourceOption****"*/>  </bean>  <beanid=*"sqlSessionTemplate"*class=*"org.mybatis.spring.SqlSessionTemplate"*scope=*"prototype"*>  <constructor-argindex=*"0"*ref=*"sqlSessionFactory"*/>  </bean>  <!-- 多数据源切面配置 -->  <aop:aspectorder=*"0"*ref=*"dataSourceAopAspect"*>  <aop:pointcutexpression=*"(execution(\* qx.service..\*.\*(..)) or execution(\* qx.dao..\*.\*(..))) or execution(\* com.kiso.ec..\*.\*(..))"*id=*"dataSourceAopPointcut"*/>  <aop:beforepointcut-ref=*"dataSourceAopPointcut"*method=*"before"*/>  </aop:aspect> |

## 3.使用教程

### 3.1类使用指定实例



### 3.2方法使用指定实例

