**Spring的多租户数据源管理AbstractRoutingDataSource**

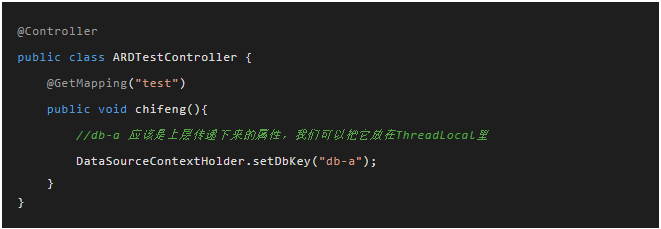
<https://mp.weixin.qq.com/s/KS8GJEqFIRTiSiWsmRTKNg>

## 1.基本原理

多数据源能进行动态切换的核心就是spring底层提供了AbstractRoutingDataSource类进行数据源路由。AbstractRoutingDataSource实现了DataSource接口，所以我们可以将其直接注入到DataSource的属性上。

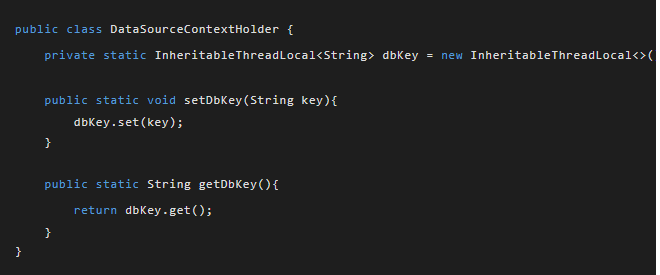
我们主要继承这个类，实现里面的方法determineCurrentLookupKey()，而此方法只需要返回一个数据库的名称即可。

比如，Controller通过拿到前端业务传递的数值，进行业务逻辑分发。它就可以手动设置当前请求的数据库标识，然后路由到正确的库表里面。



那么当sql语句执行的时候，它如何知道自己需要切换到哪个数据源呢？是不是需要把db-a这个属性一直透传下去呢？

在Java中，可以使用ThreadLocal绑定这个透传的属性。像Spring的嵌套事务等实现的原理，也是基于ThreadLocal去运行的。所以，DataSourceContextHolder.本质上是一个操作ThreadLocal的类。

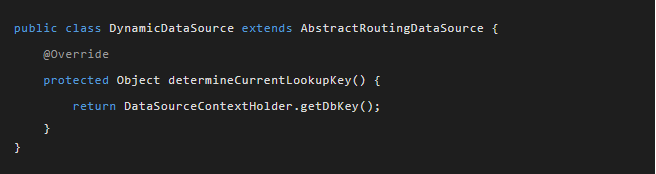


## 2.配置代码

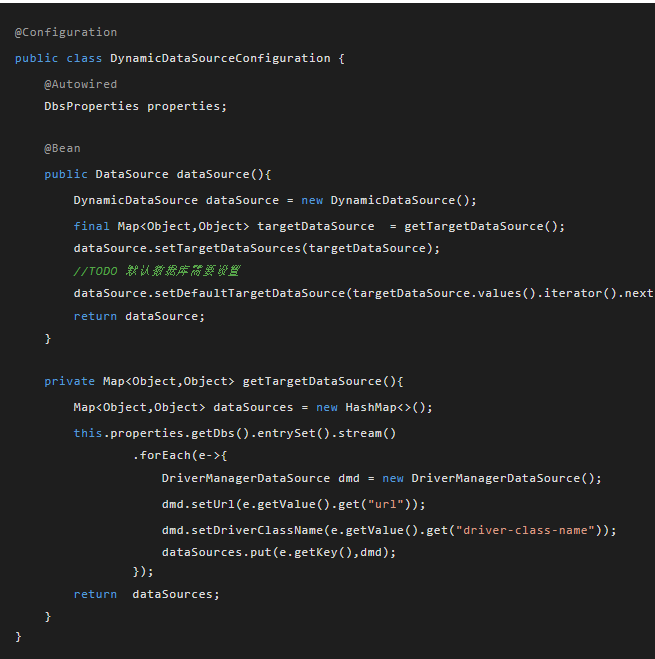
首先，我们自定义了配置文件的格式。如下面的代码，就配置了db-a和db-b两个数据库。



接下来一步，需要配置整个应用所默认的数据源。如你所见，它的主要逻辑，就是在运行的时候，从ThreadLocal里取出提前设置的这个值。



最后一步，设置整个项目中默认的DataSource。注意，我们生成DynamicDataSource之后，还需要提供targetDataSource和defaultTargetDataSource两个属性的值，才能够正常运行。



## 3.问题

通过以上简单的代码，就可以实现Spring简单的多数据源管理。但明显的，它还存在很多问题。

1. 需要产品设计选择模式，进行业务切换。
2. 前端可以采用放在localStroage的方式，保存属性，可使用拦截器方式将变量每次都传递。
3. 后端每次请求，都需要带上目标db，可以采用放在ThreadLocal里的方式。但ThreadLocal有线程透传的问题，如果任务里开启了子线程，则变量不能共享。
4. 由于表是动态选择的，所以JPA自动创建和update等模式，将不可用。不方便测试和单元测试，在测试接口的时候，也需要每次强制指定指向的库。
5. 由于是修改数据源的模式，每次增加库，都需要重新启动上线才可以。如果要做到动态性，数据源销毁是个问题。

## End

对于一个微服务来说，有很多默认的限制策略，比如，不同域之间的服务是不能共享一个数据库的。这些基本原则，把微服务整的清清爽爽，是一些基本的原则。

同理的，如果我们在设计开始，就给每一张表加上租户的字段ID，那么写代码的时候就顺畅的多。但是世界上没有这么多如果。