背景前言

因为有同学需要在项目运行过程中能够动态修改数据源（即：数据源的热更新）。所以在这里介绍一下如何动态在配置中心修改数据源。这里以com.alibaba.druid.pool.DruidDataSource数据源为例。

目标：数据源的热更新

主要步骤有三：

第一步：重写DruidAbstractDataSource类

第二步：配置动态获取nacos配置信息

第三步：测试数据源是否刷新

实现过程

第一步：重写DruidAbstractDataSource类

这里为什么要重写这个类：因为DruidDataSource数据源在初始化后，就不允许再重新设置数据库的url和userName

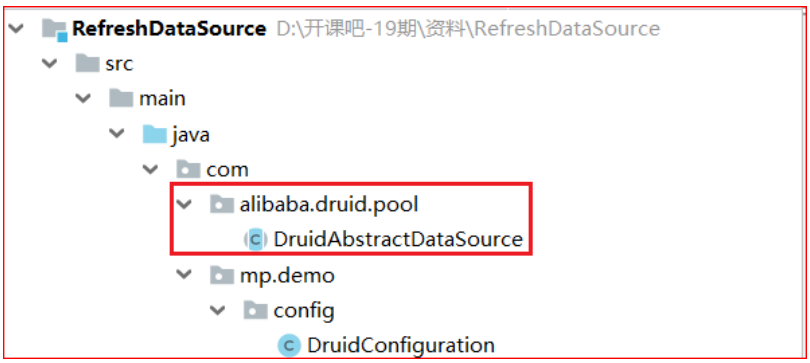
DruidAbstractDataSource.java

```
1097    public String getRawJdbcUrl() { return jdbcUrl; }
1100
1101    public void setUrl(String jdbcUrl) {
1102        if (StringUtils.equals(this.jdbcUrl, jdbcUrl)) {
1103            return;
1104        }
1105
1106        // 重写的时候，需要将这个判断注释掉，否则会报错
1107        // if (inited) {
1108        //     throw new UnsupportedOperationException();
1109        // }
1110
```


```java
public void setUrl(String jdbcUrl) {
    if (StringUtils.equals(this.jdbcUrl, jdbcUrl)) {
        return;
    }
    // 重写的时候，需要将这个判断注释掉，否则会报错
    // if (inited) {
    //     throw new UnsupportedOperationException();
    // }
    if (jdbcUrl != null) {
        jdbcUrl = jdbcUrl.trim();
    }
    this.jdbcUrl = jdbcUrl;

    // if (jdbcUrl.startsWith(ConfigFilter.URL_PREFIX))
{
    // this.filters.add(new ConfigFilter());
    // }
}
public void setUsername(String username) {
    if (StringUtils.equals(this.username, username)) {
        return;
    }
    // 重写的时候，需要将这个判断注释掉，否则会报错
    // if (inited) {
    //     throw new UnsupportedOperationException();
    // }
    this.username = username;
}
```

重写的时候包路径不能变，只有这样类加载的时候才会优先加载重写后的类



第二步：配置动态获取nacos配置信息

```
1  package com.mp.demo.config;
2  @Slf4j
3  @Configuration
4  @RefreshScope
5  @Data
6  public class DruidConfiguration
7  {
8      @Value("${spring.datasource.url}")
9      private String dbUrl;
10
11     @Value("${spring.datasource.username}")
12     private String username;
13
14     @Value("${spring.datasource.password}")
15     private String password;
16
17     @Value("${spring.datasource.driver-class-name}")
18     private String driverClassName;
19
20     @Bean
21     @RefreshScope
22     public DruidDataSource dataSource()
23     {
24         DruidDataSource datasource = new
   DruidDataSource();
25         datasource.setUrl(this.dbUrl);
26         datasource.setUsername(username);
27         datasource.setPassword(password);
28         datasource.setDriverClassName(driverClassName);
29         return datasource;
30     }
```

```
}
```

> 这里要注意增加`@RefreshScope`注解

第三步：测试数据源是否刷新

```
    @GetMapping("/refresh")
    public String refresh() throws SQLException
    {
        DruidDataSource master =
SpringUtils.getBean("dataSource");
        master.setUrl(druidConfiguration.getDbUrl());

 master.setUsername(druidConfiguration.getUsername());

 master.setPassword(druidConfiguration.getPassword());

 master.setDriverClassName(druidConfiguration.getDriver
ClassName());
        master.restart();
        return userName + "<>" + jdbcUrl+"----------
"+druidConfiguration.getDbUrl();
    }
```

> 源码在资料中