

参考：

前言

当今的互联网项目是越来越复杂的，分布式系统后还有了分布式数据库，而这个分布式数据库就造成了往往我们一个系统可能需要访问两个以上的数据库，这就意味着我们的系统可能得维护两个以上的连接。本篇便就是要结合Spring Boot和Mybatis实现多数据源的配置。

Demo

```
#MySQL数据源1
spring1:
  datasource:
    username: root
    jdbc-url: jdbc:mysql://127.0.0.1:3306/test?useUnicode=true&characterEncoding=UTF-8&serverTimezone=GMT%2B8
    password: 123456
    driver-class-name: com.mysql.jdbc.Driver
#MySQL数据源2
spring2:
  datasource:
    username: root
    jdbc-url: jdbc:mysql://127.0.0.1:3306/jlbblog?
useUnicode=true&characterEncoding=UTF-8&serverTimezone=GMT%2B8
    password: 123456
    driver-class-name: com.mysql.jdbc.Driver
```

先配置两个数据源配置，前缀区分好，自然也不会自动识别了，我们这时候写配置类手动注入配置。

```
@Configuration
public class DataSourceConfig {

    @Bean(name = "primaryDataSource")
    @Qualifier("primaryDataSource")
    @ConfigurationProperties(prefix = "spring1.datasource")
    public DataSource primaryDataSource() {
        return DataSourceBuilder.create().build();
    }
}
```

```

@Bean(name = "secondaryDataSource")
@Qualifier("secondaryDataSource")
@Primary
@ConfigurationProperties(prefix = "spring2.datasource")
public DataSource secondaryDataSource() {
    return DataSourceBuilder.create().build();
}
}

```

我们这边是要用Mybatis去使用数据源，自然也得写配置类重新配置，两个数据源自然要两个配置类。

```

@Configuration
@MapperScan(basePackages = {"com.xyz.dao1"}, sqlSessionFactoryRef = "sqlSessionFactory1")
public class MybatisDbAConfig {

    @Autowired
    @Qualifier("primaryDataSource")
    private DataSource ds1;

    @Bean
    public SqlSessionFactory sqlSessionFactory1() throws Exception {
        SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();
        factoryBean.setDataSource(ds1);
        return factoryBean.getObject();
    }

    @Bean
    protected SqlSessionTemplate sqlSessionTemplate1() throws Exception {
        SqlSessionTemplate template = new SqlSessionTemplate(sqlSessionFactory1()); // 使用上面配置的Factory
        return template;
    }
}

```

```

@Configuration
@MapperScan(basePackages = {"com.xyz.dao2"}, sqlSessionFactoryRef = "sqlSessionFactory2")
public class MybatisDbBConfig {

    @Autowired
    @Qualifier("secondaryDataSource")
    private DataSource ds2;

    @Bean
    public SqlSessionFactory sqlSessionFactory2() throws Exception {
        SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();
        factoryBean.setDataSource(ds2);
        return factoryBean.getObject();
    }
}

```

```
}  
  
@Bean  
public SqlSessionTemplate sqlSessionTemplate2() throws Exception {  
    SqlSessionTemplate template = new SqlSessionTemplate(sqlSessionFactory2());  
    return template;  
}  
}
```

注意到两个库对应的持久层的包是不一样的，这样不同的数据源就去不同的包下找映射。

其它的使用就和以前一样就行了。