

配置多数据源 required a single bean, but 2 were found:

作者: [workJun](#)

原文链接: <https://ld246.com/article/1587627180715>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

两个数据源的时候可以，加了一个UserDb就不行了

Description:

```
Parameter 2 of method dataSource in com.gomefinance.gold.user.common.config.DruidConfig required a single bean but 2 were found:  
- masterDataSource: defined by method 'masterDataSource' in class path resource [com.gomefinance.gold.user.common.config.DruidConfig.class]  
- userDbDataSource: defined by method 'userDbDataSource' in class path resource [com.gomefinance.gold.user.common.config.DruidConfig.class]
```

Action:

Consider marking one of the beans as @Primary, updating the consumer to accept multiple beans, or using @Qualifier to identify the bean that should be consumed

@Configuration

```
public class DruidConfig
```

```
{  
    @Bean  
    @ConfigurationProperties("spring.datasource.druid.master")  
    public DataSource masterDataSource()  
    {  
        return DruidDataSourceBuilder.create().build();  
    }  
  
    @Bean  
    @Qualifier("userDbDataSource")  
    @ConfigurationProperties("spring.datasource.druid.userDb")  
    @ConditionalOnProperty(prefix = "spring.datasource.druid.userDb", name = "enabled", havingValue = "true")  
    public DataSource userDbDataSource() {  
        return DruidDataSourceBuilder.create().build();  
    }  
  
    @Bean  
    @Qualifier("slaveDataSource")  
    @ConfigurationProperties("spring.datasource.druid.slave")  
    @ConditionalOnProperty(prefix = "spring.datasource.druid.slave", name = "enabled", havingValue = "true")  
    public DataSource slaveDataSource()  
    {  
        return DruidDataSourceBuilder.create().build();  
    }  
  
    @Bean(name = "dynamicDataSource")  
    @Primary  
    public DynamicDataSource dataSource(DataSource masterDataSource, DataSource userDbDataSource, DataSource slaveDataSource)  
    {  
        Map<Object, Object> targetDataSources = new HashMap<>();  
        targetDataSources.put(DataSourceType.MASTER.name(), masterDataSource);  
        targetDataSources.put(DataSourceType.USERDB.name(), userDbDataSource);  
        targetDataSources.put(DataSourceType.SLAVE.name(), slaveDataSource);  
        return new DynamicDataSource(masterDataSource, targetDataSources);  
    }  
}
```