

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#)

[您还未登录！登录](#)



QING \_\_\_\_\_

浏览: 1437200 次

性别:

来自: 北京



最近访客

[更多访客>>](#)



[IXHONG](#)



[starkgeng](#)



[qjxtdxwf](#)



[overflow\\_exception](#)

博主相关

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)

文章分类

[全部博客 \(275\)](#)

[JAVA \(119\)](#)

[多线程与并发 \(7\)](#)

[NIO \(13\)](#)

[MYSQL \(10\)](#)

[BigTable \(9\)](#)

[Zookeeper \(21\)](#)

[Redis \(18\)](#)

[JMS \(6\)](#)

[其他 \(11\)](#)

[kafka \(4\)](#)

[ActiveMQ \(9\)](#)

[hadoop \(18\)](#)[架构 \(27\)](#)[nginx \(7\)](#)[mongodb \(13\)](#)[linux \(5\)](#)[leetcode \(9\)](#)

社区版块

[我的资讯 \(0\)](#)[我的论坛 \(1\)](#)[我的问答 \(0\)](#)

存档分类

[2019-02 \(3\)](#)[2019-01 \(4\)](#)[2018-12 \(11\)](#)[更多存档...](#)

最新评论

[kyo472083100](#): 写得很好, 感谢楼主解释得很清晰[Mongodb中Mapreduce特性与原理](#)[lanhz](#): 对于三个节点: 1、2、3, 如果2向1广播投给2 (即2自己), 1 ...[Zookeeper选举过程描述与状态迁移](#)[Bll](#): 厉害了[GRPC原理解析](#)[QING\\_\\_\\_\\_\\_](#): xiatiandebaofengyu 写道有个问题: 发布者在发 ...[Redis编程实践【pub/sub】](#)[xiatiandebaofengyu](#): 有个问题: 发布者在发布消息时, 是往所有的订阅者消息队列上存放该 ...[Redis编程实践【pub/sub】](#)[SpringBoot多数据源配置](#)



博客分类:

[JAVA](#)

Springboot单个数据源比较容易配置, 它自动装配特性支持下, DataSource的初始化和注入等比较便捷。但是在多数数据源情况下, 反而有些蹩脚; 其实实施起来也不算太复杂, 方式也比较多。本文展示一个比较易用的方式, 仅供参考。

我们假定有2个数据源, dataSource1和dataSource2, 基于tomcat-jdbc连接池, 粗略配置如下文所示, 部分属性可以自行按需增加。

## 1、application.properties

Java代码  

```
##### dataSource #####
datasource:
default:
url: jdbc:mysql://127.0.0.1:3306/test?autoReconnect=false
username: root
password: root
maxActive: 32
maxIdle: 4
minIdle: 1
defaultAutoCommit: true
maxAge: 60000
testOnBorrow: true
```

```
testOnReturn: true
validationQuery: SELECT 1
db0:
url: jdbc:mysql://127.0.0.1:3306/test0?autoReconnect=false
username: root
password: root
maxActive: 32
maxIdle: 4
minIdle: 1
defaultAutoCommit: true
maxAge: 60000
testOnBorrow: true
testOnReturn: true
validationQuery: SELECT 1
db1:
url: jdbc:mysql://127.0.0.1:3306/test1?autoReconnect=false
username: root
password: root
maxActive: 32
maxIdle: 4
minIdle: 1
defaultAutoCommit: true
maxAge: 60000
testOnBorrow: true
testOnReturn: true
validationQuery: SELECT 1
```

## 2、Application.java

Java代码 ✨ ☆

```
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})
@MapperScan("com.vipkid.sample.dao")
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @ConfigurationProperties(prefix = "datasource.default")
    @Bean("defaultDataSource")
    public DataSource defaultDataSource() {
        return DataSourceBuilder.create().build();
    }

    @ConfigurationProperties(prefix = "datasource.db0")
    @Bean(value = "dataSource0")
    public DataSource dataSource0() {
        return DataSourceBuilder.create().build();
    }

    @ConfigurationProperties(prefix = "datasource.db1")
    @Bean(value = "dataSource1")
    public DataSource dataSource1() {
        return DataSourceBuilder.create().build();
    }
}
```

原理比较简单,对于Spring容器而言,在Bean初始化之前(Spring Bean可以被export之前,并非Bean实例创建之前),可以执行已注册的“postProcessor”来完成一些特定操作。对于SpringBoot环境,如果@Bean被@ConfigurationProperties修饰,那么将会在Bean初始化之前通过“**ConfigurationPropertiesBindingPostProcessor**”执行一些属性设置操作:根据ConfigurationProperties.prefix通过匹配获得有效的properties列表,依次通过反射机制(内部有复杂的转换)获取同名property的setter方法并执行;最终目的就是将在@ConfigurationProperties中限定的属性列表与@Bean实例同名属性进行赋值。

以“defaultDataSource”为例:

1) **DataSource defaultDataSource()**方法内部,就是尝试加载指定的一些dataSource类型,如classpath中存在则创建相应的DataSource实例。比如我们引入了tomcat-jdbc依赖,那么将会加载“org.apache.tomcat.jdbc.pool.DataSource”并创建其实例。当然你可以强制指定:

Java代码 ★ ☆

```
return DataSourceBuilder.create()
    .type(org.apache.tomcat.jdbc.pool.DataSource.class)
    .build();
```

此方法返回dataSource实例,但是此实例并没有任何我们所需要的属性。即此时@ConfigurationProperties注释还没有发挥实际作用,只是@Bean部分执行完毕。

2) 根据Spring Bean原理,@Bean在export之前,需要执行一些postProcessor;那么在SpringBoot环境下,被@ConfigurationProperties修饰的Bean,需要执行ConfigurationPropertiesBindingPostProcessor,那么在yml中申明的属性才会被bind到DataSource实例中;比如“url”、“maxActive”等等。内部主要是PropertiesConfigurationFactory在发挥作用,将prefix传递给此Factory,此Factory将会从全局的properties列表中匹配符合prefix的属性列表(会遵守springboot properties规范,比较将大写与“-”转化等),并对属性依次构建“DataSource”实例的setter方法(反射机制Method),然后执行。

(源码参见, *ConfigurationPropertiesBindingPostProcessor.postProcessBeforeInitialization(Object bean,String beanName)*,其中bean为dataSource实例,beanName为“defaultDataSource”)



3) 此外,为了避免与SpringBoot默认的dataSource装配冲突,我们需要禁用内置的DataSourceAutoConfiguration:

Java代码 ★ ☆

```
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})
```

## 平均薪酬38K! 前端工程师凭什么这么值钱?

2019年,这类程序员需求增速达174%? 原因竟是...

分享到:  

[Spring环境读取Properties文件与编码问题](#) | [JMX最佳实践与详解](#)

2018-03-31 17:58

浏览 4976

[评论\(0\)](#)

分类:[编程语言](#)

[查看更多](#)

评论

发表评论



[您还没有登录,请您登录后再发表评论](#)

相关资源推荐

[springboot多数据源配置](#)

[springboot多数据源配置](#)

## [Springboot mybatis 多数据源配置项目实例](#)

Springboot mybatis 多数据源配置 项目实例.....

## [springboot 多数据源配置与使用](#)

多数据源配置 application.properties 配置两个数据库 #数据库配置#

```
spring.datasource.primary.driverClassName = com.mysql.jdbc.Driver spring.datasource.primary.url = jdbc:mysql://x.x.x.x:3306/hll spring.datasource.p...
```

## [SpringBoot配置多数据源](#)

SpringBoot下配置多数据源，有的时候我们的项目可能需要从不同的数据源获取或者操作数据，这个时候就需要配置多数据源；环境 Spring-Boot 1.5.3.RELEASE、JDK1.8、HikariDataSource 搭建SpringBoot项目可以参看我之前写的博客；配置HikariDataSource数据库连接池可以参看[SpringBoot配置 HikarDataSource](

## [SpringBoot配置多数据源（结合Druid）](#)

在单数据源的情况下，Spring Boot的配置非常简单，只需要在application.properties文件中配置连接参数即可。但是往往随着业务量发展，我们通常会进行数据库拆分或是引入其他数据库，从而我们需要配置多个数据源，下面介绍多数据源的配置方式。主要目录结构：主要关注红色方框中与本节相关的类。

pom.xml <?xml version="1.0" encoding="UTF-

## [spring boot 注解方式配置多数据源与使用](#)

spring boot 注解方式配置多数据源与使用

## [springboot-多数据源配置-mybatis-jpa](#)

springboot 多数据源配置代码示例，不错的学习资料。

## [SpringBoot2.0之八 多数据源配置](#)

在开发的过程中我们可能都会遇到对接公司其他系统等需求，对于外部的系统可以采用接口对接的方式，对于一个公司开发的两个系统，并且知道相关数据库结构的情况下，就可以考虑使用多数据源来解决这个问题。SpringBoot为我们提供了相对简单的实现。一、建立如下结构的maven项目 二、添加相关数据库配置信息 server: port: 8080 spring: datasour...

## [SpringBoot之MyBatis多数据源配置实现](#)

1.新建SpringBootDemo\_mutilDatasource项目 配置pom.xml <project

```
xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM
```

## [springboot中多数据源的配置](#)

1.场景还原 在实际项目中，一个工程配置多个数据源很常见，工程可能会根据业务或者模块访问不同的数据库或表；今天笔者就springboot中配置多数据源作个详细的讲解 2.实现方案 注意：一个应用工程中有且只有一个启动类，其依赖的模块不能是带有启动类的模块 ①application.yml配置 spring: datasource: druid: maste...

## [springboot+mybatis 多数据源配置，AOP注解动态切换数据源](#)

应用场景：项目使用多数据源，并实现读写分离。springboot默认加载application.properties或application.yml配置，配置规则已经定好且为单数据源，想要配置多数据源必须禁用默认加载，然后手动去配置多数据源，完整代码如下：数据源配置：application.properties#springboot单数据源配置 spring.datasource.url=jdbc:...

## [SpringBoot+mybatis之配置多数据源](#)

1.在配置文件中添加多数据源

## [手把手教你用springboot配置多数据源](#)

## [SpringBoot配置多数据源\(druid\)](#)

分析 spring本身是支持多数据源动态切换的，AbstractRoutingDataSource这个抽象类就是spring提供的一个数据源路由的一个入口，该抽象类暴露了一个determineCurrentLookupKey()的方法，该方法返回值是Object，该返回值作为key去取Map中的DataSource。AbstractRoutingDataSource getConnect...

## [SpringBoot的多数据源配置](#)

在项目中需要在不同的IP里的数据库获取数据，所以要求要可以灵活的指定具体要操作的数据库。主要使用的框架是spring-boot+mybatis等。一：先将maven项目配置好（略）；二：*springboot*的启动类：  
import org.apache.log4j.Logger; import org.springframework.boot.SpringApplication; import

## [springBoot多数据源配置以及事务控制](#)

一：多数据源配置 1、主数据源配置文件 package com.bert.datasource; import  
org.apache.ibatis.session.SqlSessionFactory; import org.mybatis.spring.SqlSessionFactoryBean; import  
org.mybatis.spring.SqlSessionTemplate; i...

## [SpringBoot多数据源及事务配置](#)

多数据源配置 其实多数据源的配置大体上跟Spring是相同的，不同之处仅在于*SpringBoot*推崇省略xml配置，取而代之使用JavaConfig来配置多数据源 针对目前所使用的Spring+MyBatis框架而言，数据源的配置最重要的两点是配置DataSource以及SqlSessionFactory，然而为了实现动态数据源我们还需要拓展AbstractRoutingDataSou

## [SpringBoot 多数据源配置](#)

本文主要描述在 *SpringBoot* 中多数据源配置及在配置过程中遇到的一些坑

## [Spring Boot系列之八 多数据源配置整体介绍](#)

简介 前面两章在介绍Spring Boot整合MyBatis的时候简单的看了一下Spring Boot如何去配置单一的数据源.并没有详细的单独去看数据源的配置. 本文就简要介绍下在Spring Boot中对内置内存数据库H2,多数据源等的配置. 由于本文档的代码基本都是基于MyBatis搭建，所以关于Spring-Data-JPA的内容并不能展示出来，Spring Boot中

## [SpringBoot多数据源配置和使用](#)

*SpringBoot*中使用JPA如果是单数据源，很好处理，只需要在配置文件中配置连接参数和性能参数即可。如下：username、password、url、driver-class-name等配置省略。。。spring.datasource.type=com.alibaba.druid.pool.DruidDataSource spring.datasource.driver-class-n...

springboot多数据源配置 2017-07-28 29

springboot多数据源配置

Springboot mybatis多数据源配置项目实例 2017-11-19 53

Springboot mybatis 多数据源配置 项目实例.....

海外点卡 便宜



springboot 多数据源配置与使用 2018-07-20

1569

多数据源配置 application.properties 配置两个数据...

SpringBoot配置多数据源 2017-10-24 10700

SpringBoot下配置多数据源，有的时候我们的项目...

SpringBoot配置多数据源（结合Druid） 2017-09-13 4489

在单数据源的情况下，Spring Boot的配置非常简单...

手把手教你用springboot配置多数据源 2018-09-28 244

1.文件结构： 2.pom： &lt;project xmlns=&quot;...

单反相机排行榜



SpringBoot2.0之八 多数据源配置 2018-04-27 11493

在开发的过程中我们可能都会遇到对接公司其...

spring boot 注解方式配置多数据源与使用 2018-01-06 5495

spring boot 注解方式配置多数据源与使用

SpringBoot之MyBatis多数据源配置实现 2017-12-18 5603

1.新建SpringBootDemo\_mutilDatasource项目 配置...

springboot-多数据源配置 -mybatis-jpa 2018-05-22 3

springboot 多数据源配置代码示例，不错的学习资...

springboot中多数据源的配置 2018-07-27 631

1.场景还原 在实际项目中，一个工程配置多个数...

第三方支付公司

序号	名称	开发语言	第三方支付	第三方支付平台	第三方支付接口
1	银联	银联	银联	银联	银联
2	支付宝	支付宝	支付宝	支付宝	支付宝
3	微信支付	微信支付	微信支付	微信支付	微信支付
4	京东支付	京东支付	京东支付	京东支付	京东支付
5	百度支付	百度支付	百度支付	百度支付	百度支付
6	美团支付	美团支付	美团支付	美团支付	美团支付
7	饿了么支付	饿了么支付	饿了么支付	饿了么支付	饿了么支付
8	携程支付	携程支付	携程支付	携程支付	携程支付
9	去哪儿支付	去哪儿支付	去哪儿支付	去哪儿支付	去哪儿支付
10	途牛支付	途牛支付	途牛支付	途牛支付	途牛支付

springboot+mybatis多数据源配置，AOP注解 2018-05-06 8615

应用场景：项目使用多数据源，并实现读写分离。s...

SpringBoot+mybatis之配置多数据源 2018-06-21 583

1.在配置文件中添加多数据源

springBoot多数据源配置以及事务控制 2018-09-17 237

一：多数据源配置 1、主数据源配置文件 package ...

SpringBoot配置多数据源(druid) 2018-09-13 577

分析 spring本身是支持多数据源动态切换的，Abstr...





CSDN学院



CSDN企业招聘

---

👤 QQ客服

✉ kefu@csdn.net

🗣 客服论坛

☎ 400-660-0108

工作时间 8:30-22:00

---

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

🐼 百度提供站内搜索 [京ICP证19004658号](#)

©1999-2019 北京创新乐知网络技术有限公司

---

[网络110报警服务](#) [经营性网站备案信息](#)

[北京互联网违法和不良信息举报中心](#)

[中国互联网举报中心](#)

---

Global site tag (gtag.js) - Google Analytics