



《Spring Boot 基础教程》

第 29 节

Spring Boot 集成 Druid

## 一、Druid 的介绍

主要功能对比

	Druid	BoneCP	DBCP	C3P0	Proxool	JBoss	Tomcat-Jdbc
LRU	是	否	是	否	是	是	?
PSCache	是	是	是	是	否	否	是
PSCache-Oracle-Optimized	是	否	否	否	否	否	否
ExceptionSorter	是	否	否	否	否	是	否
更新维护	是	否	否	否	否	?	是

<https://github.com/alibaba/druid/wiki/%E5%90%84%E7%A7%8D%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%9E%E6%8E%A5%E6%B1%A0%E5%AF%B9%E6%AF%94>

中文文档：<https://github.com/alibaba/druid/wiki/%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98>

## 二、添加依赖

```
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
    <version>1.0.26</version>
</dependency>
```

## 三、配置

```
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
#spring.datasource.type=org.apache.tomcat.jdbc.pool.DataSource
spring.datasource.url=jdbc:mysql://localhost/spring_boot_demo?useUnicode=true&characterEncoding=utf-8
spring.datasource.username=root
spring.datasource.password=123456
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

注意：关于 spring.datasource.type 的说明

旧版本不支持这个属性，1.3.x 开始支持，但是 1.4.0 不支持，1.4.1 重新支持。

## 四、添加 druid 的支持类

```
package com.roncoo.example.util.configuration;

import org.springframework.boot.autoconfigure.condition.ConditionalOnClass;
```



```
import org.springframework.boot.autoconfigure.condition.ConditionalOnProperty;
import org.springframework.boot.autoconfigure.jdbc.DataSourceProperties;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.boot.jdbc.DatabaseDriver;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import com.alibaba.druid.pool.DruidDataSource;

@Configuration
public class DruidConfiguration {
    @ConditionalOnClass(DruidDataSource.class)
    @ConditionalOnProperty(name = "spring.datasource.type", havingValue =
"com.alibaba.druid.pool.DruidDataSource", matchIfMissing = true)
    static class Druid extends DruidConfiguration {
        @Bean
        @ConfigurationProperties("spring.datasource.druid")
        public DruidDataSource dataSource(DataSourceProperties properties) {
            DruidDataSource druidDataSource = (DruidDataSource)
properties.initializeDataSourceBuilder().type(DruidDataSource.class).build();
            DatabaseDriver databaseDriver =
DatabaseDriver.fromJdbcUrl(properties.determineUrl());
            String validationQuery = databaseDriver.getValidationQuery();
            if (validationQuery != null) {
                druidDataSource.setValidationQuery(validationQuery);
            }
            return druidDataSource;
        }
    }
}
```

## 五、监控

### 1. 配置 servlet

```
package com.roncoo.example.util.servlet;

import javax.servlet.annotation.WebInitParam;
import javax.servlet.annotation.WebServlet;

import com.alibaba.druid.support.http.StatViewServlet;

/**
 *
 * @author wujing
 */
```



```
@WebServlet(urlPatterns = { "/druid/*" }, initParams =
{ @WebInitParam(name = "loginUsername", value = "roncoo"),
@WebInitParam(name = "loginPassword", value = "roncoo") })
public class DruidStatViewServlet extends StatViewServlet {

    private static final long serialVersionUID = 1L;

}
```

## 2. 配置 filter

```
package com.roncoo.example.util.filter;

import javax.servlet.annotation.WebFilter;
import javax.servlet.annotation.WebInitParam;

import com.alibaba.druid.support.http.WebStatFilter;

/**
 * @author wujing
 */
@WebFilter(filterName = "druidWebStatFilter", urlPatterns =
"/*", initParams = { @WebInitParam(name = "exclusions", value
= "*.js,*.gif,*.jpg,*.bmp,*.png,*.css,*.ico,/druid/*") })
public class DruidWebStatFilter extends WebStatFilter {

}
```

### #初始化连接大小

```
spring.datasource.druid.initial-size=8
```

### #最小空闲连接数

```
spring.datasource.druid.min-idle=5
```

### #最大连接数

```
spring.datasource.druid.max-active=10
```

### #查询超时时间

```
spring.datasource.druid.query-timeout=6000
```

### #事务查询超时时间

```
spring.datasource.druid.transaction-query-timeout=6000
```

### #关闭空闲连接超时时间

```
spring.datasource.druid.remove-abandoned-timeout=1800
```

## 3. 测试: <http://localhost:8080/druid/index.html>

## 4. sql 监控配置

```
#filter类名:stat,config,encoding,logging
```



---

```
spring.datasource.druid.filters=stat
```

#### 5. spring 监控配置

```
@ImportResource(locations = { "classpath:druid-bean.xml" })
```

更多课程信息，请关注 龙果学院 官方网站 <http://www.roncoo.com/>

或关注 龙果 微信公众号 RonCoo\_com

