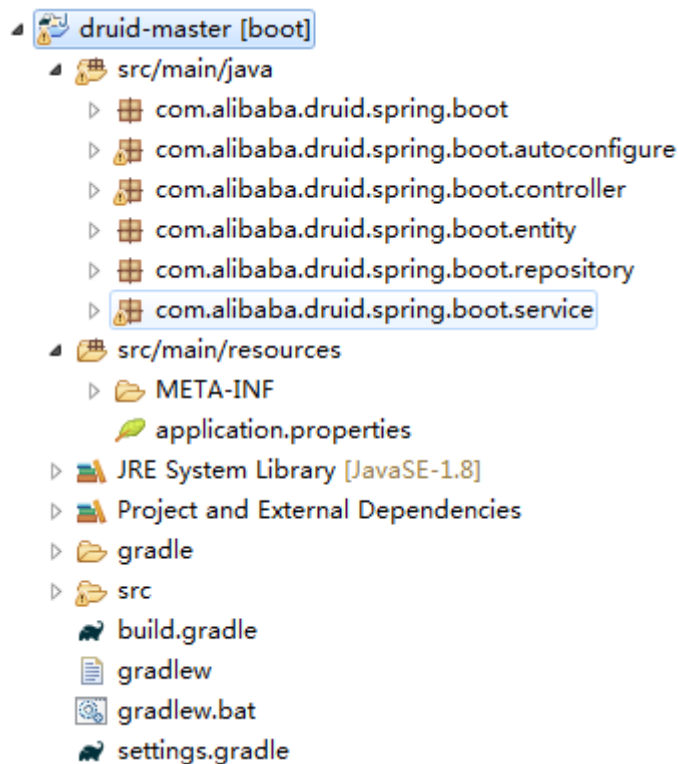


# Druid使用文档

## 下载源码

github下载地址<https://github.com/jijiana/druid.git>

目录结构



application配置连接池相关信息

```
# 只有下面三个是必填项（使用内嵌数据库的话这三个也可以不用填，会使用默认配置），其他配置不是必须的
spring.datasource.druid.url=jdbc:mysql://localhost:3306/db_test?
useUnicode=true&characterEncoding=UTF-
8&useSSL=false&autoReconnect=true&failOverReadOnly=false&allowPublicKeyRetrieval=true
spring.datasource.druid.username=root
spring.datasource.druid.password=123456
# driver-class-name 非必填可根据url推断
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
#方言
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
#hibernate sql格式化
spring.jpa.properties.hibernate.format_sql = false
#hibernate sql 是否展示
spring.jpa.properties.hibernate.show_sql = false

#Druid 初始化
spring.datasource.druid.initial-size=2
#Druid 最大
```

```
spring.datasource.druid.max-active=30
#Druid 最小
spring.datasource.druid.min-idle=2
# 配置获取连接等待超时的时间
spring.datasource.druid.max-wait=60000
# 打开PSCache，并且指定每个连接上PSCache的大小
spring.datasource.druid.pool-prepared-statements=true
spring.datasource.druid.max-pool-prepared-statement-per-connection-size=5

#检测连接是否有效的sql
spring.datasource.druid.validation-query=select 1
#检测连接是否有效的sql等待超时时间
spring.datasource.druid.validation-query-timeout=60000
#当从连接池借用连接时，是否测试该连接。
spring.datasource.druid.test-on-borrow=true
#在连接归还到连接池时是否测试该连接。
spring.datasource.druid.test-on-return=true
#当连接空闲时，是否执行连接测试。
spring.datasource.druid.test-while-idle=true
#空闲连接回收的时间间隔，与test-while-idle一起使用
spring.datasource.druid.time-between-eviction-runs-millis=10000
#连接池空闲连接的有效时间
spring.datasource.druid.min-evictable-idle-time-millis=1800000
# 配置监控统计拦截的filters，去掉后监控界面sql无法统计，'wall'用于防火墙
spring.datasource.druid.filters=stat,wall
# 通过connectProperties属性来打开mergeSql功能；慢SQL记录
spring.datasource.connectionProperties=druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000

#druid 登录用户名、密码
spring.datasource.druid.stat-view-servlet.login-username=admin
spring.datasource.druid.stat-view-servlet.login-password=admin
#IP白名单(没有配置或者为空，则允许所有访问)
#spring.datasource.druid.stat-view-servlet.allow=9.186.54.153

#IP黑名单（存在共同时，deny优先于allow）
#spring.datasource.druid.stat-view-servlet.deny=127.0.0.1
#是否启用StatFilter默认值true
spring.datasource.druid.web-stat-filter.enabled=false
#是否能够重置数据 禁用HTML页面上的“Reset All”功能
spring.datasource.druid.stat-view-servlet.reset-enable=false

#Druid 监控日志
spring.datasource.druid.filter.slf4j.enabled=true
spring.datasource.druid.filter.slf4j.connection-log-enabled=true
spring.datasource.druid.filter.slf4j.connection-close-after-log-enabled=true
spring.datasource.druid.filter.slf4j.connection-commit-after-log-enabled=true
spring.datasource.druid.filter.slf4j.connection-connect-after-log-enabled=true
spring.datasource.druid.filter.slf4j.connection-connect-before-log-enabled=true
spring.datasource.druid.filter.slf4j.connection-log-error-enabled=true
spring.datasource.druid.filter.slf4j.data-source-log-enabled=true
spring.datasource.druid.filter.slf4j.result-set-log-enabled=true
spring.datasource.druid.filter.slf4j.statement-log-enabled=true
```

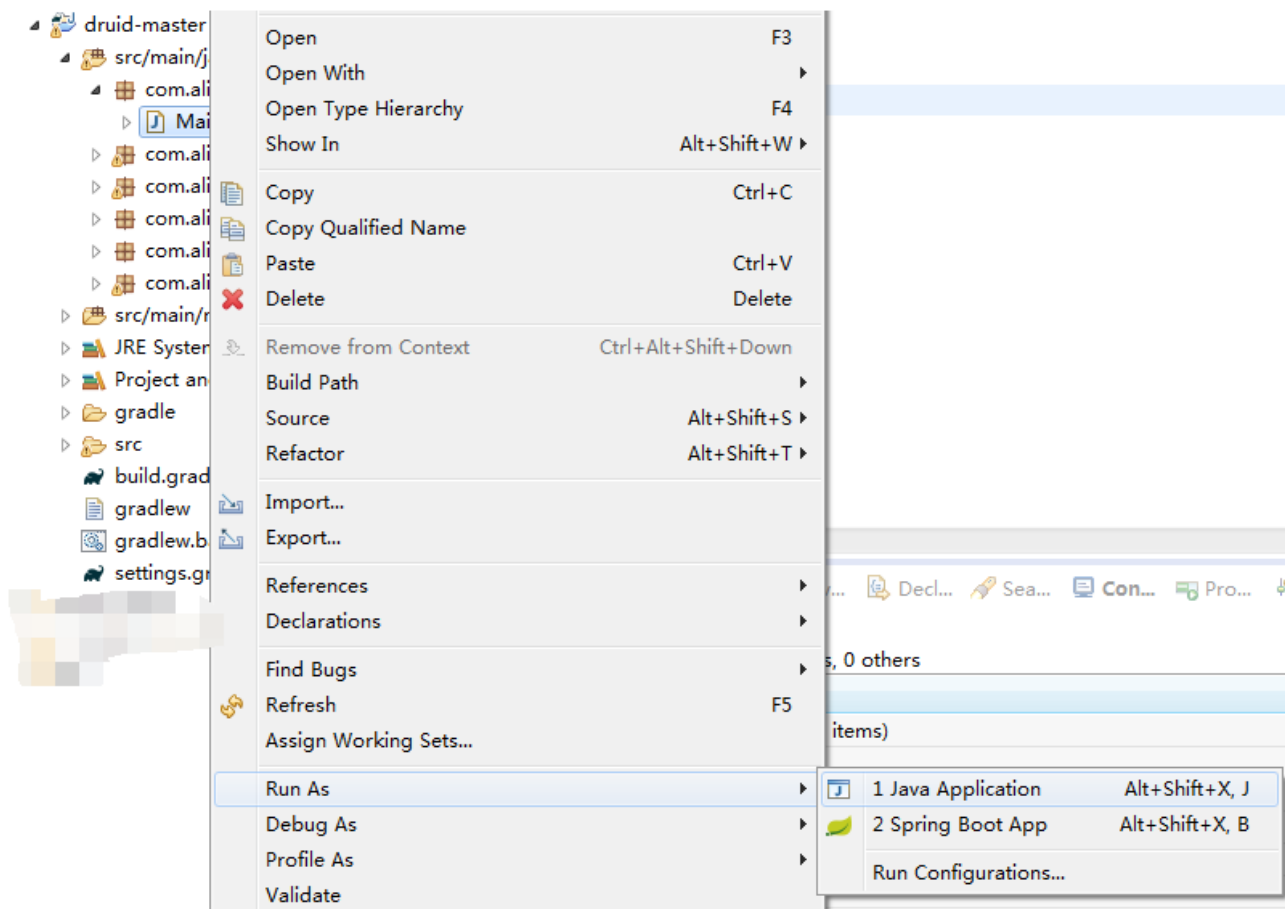
```

#cookie druid
spring.datasource.druid.web-stat-filter.principal-cookie-name=admin
#session druid
spring.datasource.druid.web-stat-filter.principal-session-name=admin
spring.datasource.druid.web-stat-filter.profile-enable=true
#session 监控是否开启
spring.datasource.druid.web-stat-filter.session-stat-enable=true
#session 监控最大数量
spring.datasource.druid.web-stat-filter.session-stat-max-count=100000

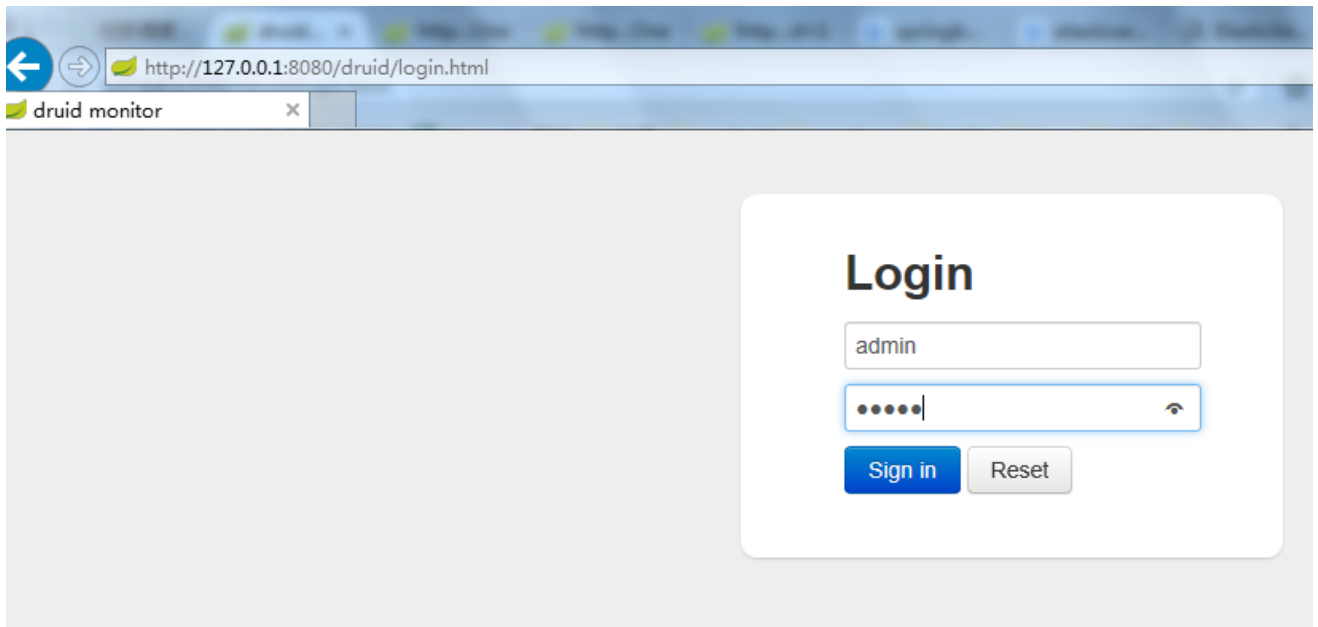
#spring监控是否开启
spring.aop.proxy-target-class=true
#spring监控, hello.controller是我的控制层包名, 也可以是服务层, 用逗号分隔多个监控内容
spring.datasource.druid.aop-patterns=com.alibaba.druid.spring.boot.controller.*
# 是否开启sql的log, 默认为: false
spring.jpa.show-sql= true
#指定DDL mode (none, validate, update, create, create-drop). 当使用内嵌数据库时, 默认是
create-drop, 否则为none.
spring.jpa.hibernate.ddl-auto=create-drop

```

通过MainApplication启动服务



在浏览器输入<http://127.0.0.1:8080/druid>



点击Sign in进入Druid监控首页，包含了如下几个模块：数据源、SQL监控、SQL防火墙、Web应用、URI监控、Session监控、JSONAPI

版本	1.1.9
驱动	com.alibaba.druid.mock.MockDriver com.mysql.jdbc.Driver com.mysql.fabric.jdbc.FabricMySQLDriver com.alibaba.druid.proxy.DruidDriver
是否允许重置	false
重置次数	0
java版本	1.8.0_151
jvm名称	Java HotSpot(TM) 64-Bit Server VM
classpath 路径	E \\projectweb_project\\springBoot\\druid-spring-boot-gradle\\bin E \\web\\gradle-4.1-bin\\gradle-4.1\\caches\\modules-2\\files-2.1\\com.alibaba\\druid-spring-boot-starter\\1.1.9\\7c133b4dbf3c94e1dd81c5bb5e4a4724ffa15d5e\\druid-spring-boot-

## 数据源

项目中管理的所有数据源配置的详细情况

Druid Monitor

首页

数据源

SQL 监控

SQL 防火墙

Web 应用

URI 监控

Session 监控

spring 监控

JSON API

重置

记录日志并重置

English | 中文

DataSourceStat List [View JSON API](#)

DataSource-657998502

Basic Info For DataSource-657998502[View JSON API](#)

* 用户名	root	指定建立连接时使用的用户名
* 连接地址	jdbc:mysql://localhost:3306/db_test?useUnicode=true&characterEncoding=UTF-8&useSSL=false&autoReconnect=true&failOverReadOnly=false&allowPublicKeyRetrieval=true	jdbc连接字符串
* 数据库类型	mysql	数据库类型
* 驱动类名	com.mysql.jdbc.Driver	jdbc驱动的分类名
* filter类名	com.alibaba.druid.filter.logging.Slf4jLogFilter,com.alibaba.druid.filter.stat.StatFilter,com.alibaba.druid.wall.WallFilter	filter的分类名
* 获取连接时检测	true	是否在获得连接后检测其可用性
* 空闲时检测	true	是否在连接空闲一段时间后检测其可用性

SQL监控

查看所有执行的sql语句

```
# 配置监控统计拦截的filters，去掉后监控界面sql无法统计，'wall'用于防火墙
spring.datasource.druid.filters=stat,wall
```

Druid Monitor

首页

数据源

SQL 监控

SQL 防火墙

Web 应用

URI 监控

Session 监控

spring 监控

JSON API

重置

记录日志并重置

English | 中文

SQL Stat [View JSON API](#)

刷新时间 5s [暂停刷新](#)

N	SQL ▼	执行数	执行时间	最慢	事务执行	错误数	更新行数	读取行数	执行中	最大并发	执行时间分布 [-----]	执行+RS时分布 [-----]	读取行分布 [-----]	更新行分布 [-----]
1	drop table if exists t_ap...	1	284	284						1	[0,0,0,1,0,0,0,0]	[0,0,0,1,0,0,0,0]	[0,0,0,0,0,0,0]	[1,0,0,0,0,0,0]
2	create table t_app_user (...	1	125	125						1	[0,0,0,1,0,0,0,0]	[0,0,0,1,0,0,0,0]	[0,0,0,0,0,0,0]	[1,0,0,0,0,0,0]

SQL防火墙

druid提供了黑白名单的访问，sql防护情况。

### 防御统计

检查次数	2	
硬检查次数	2	
非法次数		
黑名单命中次数		
黑名单长度		
白名单命中次数		
白名单长度	2	
语法错误次数		

### 表访问统计

序号	表名	Select数	SelectInto数	Insert数	Update数	Delete数	Truncate数	Create数	Alter数	Drop数	Replace数	删除数 [-- ---	更新 行分布 [-- ---	更新 数 [-- ---	读取 行分布 [-- ---
1	t_app_user							1		1					

### 函数调用统计

Function Name	InvokeCount
---------------	-------------

### SQL防御统计 - 白名单

### Web应用

目前运行的web程序的详细信息。

### WebAppStat List [View JSON API](#)

WebApp-/

ContextPath	/
执行中	0
最大并发	0
请求次数	0
SessionCount	0
事务提交数	0
事务回滚数	0
Jdbc执行数	0
Jdbc时间	0
读取行数	0

### URI监控

监控到所有的请求路径的请求次数、请求时间等其他参数。

Druid Monitor													
首页		数据源	SQL 监控	SQL 防火墙	Web 应用	URI 监控	Session 监控	spring 监控	JSON API	重置	记录日志并重置		

English | 中文

### Web URI Stat View JSON API

刷新时间 5s 暂停刷新

N	URI	请求次数	请求时间 (和)	请求最慢 (单次)	执行中	最大并发	Jdbc 执行数	Jdbc 出错数	Jdbc 时间	事务提交数	事务回滚数	读取行数	更新行数	区间分布 [-----]
1	/user/insertOne	1	191	191		1	1		43			1	1	[0,0,0,1,0,0,0,0]

## 配置Web Url 过滤规则

druid-master [boot]

src/main/java

com.alibaba.druid.spring.boot

com.alibaba.druid.spring.boot.autoconfigure

DruidDataSourceAutoConfigure.java

com.alibaba.druid.spring.boot.controller

com.alibaba.druid.spring.boot.entity

com.alibaba.druid.spring.boot.repository

com.alibaba.druid.spring.boot.service

src/main/resources

JRE System Library [JavaSE-1.8]

Project and External Dependencies

gradle

src

build.gradle

gradlew

gradlew.bat

settings.gradle

```

57 public DataSource dataSource() {
58     LOGGER.info("Init DruidDataSource");
59     return new DruidDataSourceWrapper();
60 }
61
62 @Bean
63 public FilterRegistrationBean druidStatFilter2(){
64     FilterRegistrationBean filterRegistrationBean = new FilterRegistrationBean();
65     //添加过滤规则.
66     filterRegistrationBean.addUrlPatterns("/user/insertOne");
67     filterRegistrationBean.addUrlPatterns("/user/checktransaction");
68     //添加不需要忽略的格式信息.
69     filterRegistrationBean.addInitParameter("exclusions","*.js,*.gif,*.jpg,*
70     return filterRegistrationBean;
71 }
72 }
73

```

## Session监控

当前的session状况，创建时间、最后活跃时间、请求次数、请求时间等详细参数。

```

spring.datasource.druid.web-stat-filter.principal-session-name=admin
spring.datasource.druid.web-stat-filter.profile-enable=true
#session 监控是否开启
spring.datasource.druid.web-stat-filter.session-stat-enable=true
#session 监控最大数量
spring.datasource.druid.web-stat-filter.session-stat-max-count=100000

```

Druid Monitor													
首页		数据源	SQL 监控	SQL 防火墙	Web 应用	URI 监控	Session 监控	spring 监控	JSON API	重置	记录日志并重置		

English | 中文

### Web Session Stat View JSON API

☐ Principal 过滤 | 刷新时间 5s 暂停刷新

N	SESSIONID	Principal	创建时间	最后访问时间	访问ip地址	请求次数	总共请求时间	执行中	最大并发	Jdbc 执行数	Jdbc 时间	事务提交数	事务回滚数	读取行数	更新行数
1	1D633CDDCA28801687F47E23C921726D		2,018-10-12 13:47:06	2,018-10-12 13:47:11	9.181.91.250	1	42		1	1	35			1	1

## spring监控

controller每个方法的访问情况

```

#spring监控,hello.controller是我的控制层包名,也可以是服务层,用逗号分隔多个监控内容
spring.datasource.druid.aop-patterns=com.alibaba.druid.spring.boot.controller.*

```

Spring Stat **View JSON API**

N	Class	Method	执行数	执行时间	执行中	最大并发	Jdbc执行数	Jdbc时间	事务提交数	事务回滚数	读取行数	更新行数	区间分布 [-----]
1	com.alibaba.druid.spring.boot.controller.UserController	<a href="#">insertOne()</a>	4	389		2	2	79			2	2	[0,0,2,2,0,0,0]

## JSONAPI

通过api的形式访问Druid的监控接口，api接口返回json形式数据。

JSON API

Name	Description
<a href="#">basic.json</a>	basic info.
<a href="#">datasource.json</a>	DatsSourceStat List
<a href="#">datasource-{id}.json</a>	DatasourceStat for id={id},id is DatsSourceStat.ID Example: druid/datasource-1.json
<a href="#">activeConnectionStackTrace-{datasourceId}.json</a>	StrackTrace for activeConnection in datasource which datasource.id={datasourceId},id is DatsSourceStat.ID Required: set removeAbandoned=true Example: druid/activeConnectionStackTrace-1.json
<a href="#">sql.json</a>	SqlStat for id={id} ,id is SqlStat.ID Example: druid/sql-1.json
<a href="#">WallStat.json</a>	WallStat List
<a href="#">wall-{id}.json</a>	WallStat for id={id},id is DatsSourceStat.ID Example: druid/wall-1.json
<a href="#">basic.json</a>	
<a href="#">weburi.json</a>	
<a href="#">websession.json</a>	
<a href="#">reset-all.json</a>	Reset all stat