

最近在学 druid ，就干脆边学边整理一个文档

1. 配置 druid 的常用配置项如下

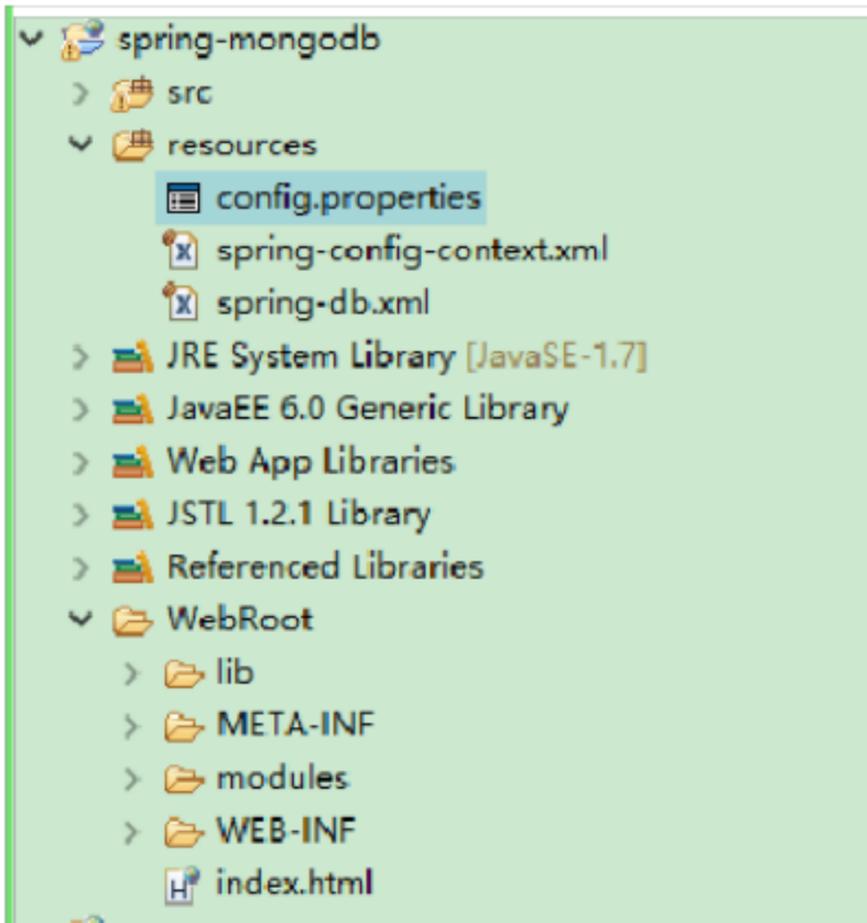
配置	缺省值	说明
name		配置这个属性的意义在于，如果存在多个数据源，监控的时候可以通过名字来区分开来。如果没有配置，将会生成一个名字，格式是： <code>"DataSource-" + System.identityHashCode(this)</code>
jdbcUrl		连接数据库的 url，不同数据库不一样。例如： mysql : <code>jdbc:mysql://10.20.153.104:3306/druid2</code> oracle : <code>jdbc:oracle:thin:@10.20.149.85:1521:ocnauto</code>
username		连接数据库的用户名
password		连接数据库的密码。如果你不希望密码直接写在配置文件中，可以使用 <code>ConfigFilter</code> 。详细看这里： https://github.com/alibaba/druid/wiki/%E4%BD%BF%E7%94%A8ConfigFilter
driverClassName	根据 url 自动识别	这一项可配可不配，如果不配置 druid 会根据 url 自动识别 dbType，然后选择相应的 driverClassName
initialSize	0	初始化时建立物理连接的个数。初始化发生在显示调用 <code>init</code> 方法，或者第一次 <code>getConnection</code> 时
maxActive	8	最大连接池数量
maxIdle	8	已经不再使用，配置了也没效果
minIdle		最小连接池数量
maxWait		获取连接时最大等待时间，单位毫秒。配置了 <code>maxWait</code> 之后，缺省启用公平锁，并发效率会有所下降，

		如果需要可以通过配置 <code>useUnfairLock</code> 属性为 <code>true</code> 使用非公平锁。
<code>poolPreparedStatements</code>	<code>false</code>	是否缓存 <code>preparedStatement</code> ，也就是 <code>PSCache</code> 。 <code>PSCache</code> 对支持游标的数据库性能提升巨大，比如说 <code>oracle</code> 。 在 <code>mysql5.5</code> 以下的版本中没有 <code>PSCache</code> 功能，建议关闭掉。 5.5 及以上版本有 <code>PSCache</code> ，建议开启。
<code>maxOpenPreparedStatements</code>	<code>-1</code>	要启用 <code>PSCache</code> ，必须配置大于 <code>0</code> ，当大于 <code>0</code> 时， <code>poolPreparedStatements</code> 自动触发修改为 <code>true</code> 。 在 <code>Druid</code> 中，不会存在 <code>Oracle</code> 下 <code>PSCache</code> 占用内存过多的问题， 可以把这个数值配置大一些，比如说 <code>100</code>
<code>validationQuery</code>		用来检测连接是否有效的 <code>sql</code> ，要求是一个查询语句。 如果 <code>validationQuery</code> 为 <code>null</code> ， <code>testOnBorrow</code> 、 <code>testOnReturn</code> 、 <code>testWhileIdle</code> 都不会起作用。在 <code>mysql</code> 中通常为 <code>select 'x'</code> ，在 <code>oracle</code> 中通常为 <code>select 1 from dual</code>
<code>testOnBorrow</code>	<code>true</code>	申请连接时执行 <code>validationQuery</code> 检测连接是否有效， 做了这个配置会降低性能。
<code>testOnReturn</code>	<code>false</code>	归还连接时执行 <code>validationQuery</code> 检测连接是否有效， 做了这个配置会降低性能
<code>testWhileIdle</code>	<code>false</code>	建议配置为 <code>true</code> ，不影响性能，并且保证安全性。 申请连接的时候检测，如果空闲时间大于 <code>timeBetweenEvictionRunsMillis</code> ， 执行 <code>validationQuery</code> 检测连接是否有效。
<code>timeBetweenEvictionRunsMillis</code>		有两个含义： 1) <code>Destroy</code> 线程会检测连接的间隔时间 2) <code>testWhileIdle</code> 的判断依据，详细看 <code>testWhileIdle</code> 属性的说明
<code>numTestsPerEvictionRun</code>		不再使用，一个 <code>DruidDataSource</code> 只支持一个

		EvictionRun
minEvictableIdleTimeMillis		Destory 线程中如果检测到当前连接的最后活跃时间和当前时间的差值大于 minEvictableIdleTimeMillis , 则关闭当前连接。
connectionInitSqls		物理连接初始化的时候执行的 sql
exceptionSorter	根据 dbType 自动识别	当数据库抛出一些不可恢复的异常时, 抛弃连接
filters		属性类型是字符串, 通过别名的方式配置扩展插件, 常用的插件有: 监控统计用的 filter:stat 日志用的 filter:log4j 防御 sql 注入的 filter:wall
proxyFilters		类型是 List<com.alibaba.druid.filter.Filter> , 如果同时配置了 filters 和 proxyFilters , 是组合关系, 并非替换关系
removeAbandoned		对于建立时间超过 removeAbandonedTimeout 的连接强制关闭
removeAbandonedTimeout		指定连接建立多长时间就需要被强制关闭
logAbandoned		指定发生 removeabandoned 的时候, 是否记录当前线程的堆栈信息到日志中

表 1.1 配置属性

整个工程的目录：



config.properties

```
1. ds_erp.driver=oracle.jdbc.driver.OracleDriver
2. ds_erp.url=jdbc\:oracle\:thin\:@10.2.0.52\:1521\:orcl
3. ds_erp.username=nwsys
4. ds_erp.password=neiwai
5. ds_erp.initialSize=1
6. ds_erp.minIdle=1
7. ds_erp.maxActive=100
8. ds_erp.maxWait=60000
9. ds_erp.timeBetweenEvictionRunsMillis=60000
10. ds_erp.minEvictableIdleTimeMillis=300000
11. ds_erp.validationQuery=SELECT SYSDATE FROM DUAL
12. ds_erp.testWhileIdle=true
13. ds_erp.testOnBorrow=false
14. ds_erp.testOnReturn=false
15. ds_erp.poolPreparedStatements=true
16. ds_erp.maxOpenPreparedStatements=100
17. ds_erp.filters=stat
18.
19. #禁用对于长时间不使用的连接强制关闭的功能
20. removeAbandoned:false
21. #超过 30 分钟开始关闭空闲连接, 由于 removeAbandoned 为 false , 这个设置项不再起作用
22. removeAbandonedTimeout:1800
23. #将当前关闭动作记录到日志, 由于 removeAbandoned 为 false , 这个设置项不再起作用
24. logAbandoned:true
```

使用 spring 配置文件

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.   xmlns:tx="http://www.springframework.org/schema/tx"
5.   xmlns:aop="http://www.springframework.org/schema/aop"
6.   xsi:schemaLocation="http://www.springframework.org/schema/beans
7.     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
8.     http://www.springframework.org/schema/tx
9.     http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
10.    http://www.springframework.org/schema/aop
11.    http://www.springframework.org/schema/aop/spring-aop-3.0.xsd">
12.   <!-- 数据源 ERP -->
13.   <bean id="ds_erp" class="com.alibaba.druid.pool.DruidDataSource"
14.     init-method="init" destroy-method="close">
15.     <!-- 基本属性 url 、 user 、 password -->
16.     <property name="url" value="${ds_erp.url}" />
17.     <property name="username" value="${ds_erp.username}" />
18.     <property name="password" value="${ds_erp.password}" />
19.     <property name="connectionProperties" value="${ds_erp.driver}"></property>
20.
21.     <!-- 配置初始化大小、最小、最大 -->
22.     <property name="initialSize" value="${ds_erp.initialSize}" />
23.     <property name="minIdle" value="${ds_erp.minIdle}" />
24.     <property name="maxActive" value="${ds_erp.maxActive}" />
25.
26.     <!-- 配置获取连接等待超时的时间 -->
27.     <property name="maxWait" value="${ds_erp.maxWait}" />
28.
29.     <!-- 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->
30.     <property name="timeBetweenEvictionRunsMillis"
31.       value="${ds_erp.timeBetweenEvictionRunsMillis}" />
32.
33.     <!-- 配置一个连接在池中最小生存的时间，单位是毫秒 -->
34.     <property name="minEvictableIdleTimeMillis"
35.       value="${ds_erp.minEvictableIdleTimeMillis}" />
```

```

36.         如果 validationQuery 为 null ,testOnBorrow 、testOnReturn 、testWhileIdle
           都不会起作用。
37.         在 mysql 中通常为 select 'x' , 在 oracle 中通常为 select 1 from dual -->
38. <property name="validationQuery" value="${ds_erp.validationQuery}" />
39.
40. <!-- 建议配置为 true , 不影响性能, 并且保证安全性。申请连接的时候检测,
41.     如果空闲时间大于 timeBetweenEvictionRunsMillis , 执行 validationQuery 检测连接是
           否有效。 -->
42. <property name="testWhileIdle" value="${ds_erp.testWhileIdle}" />
43.
44. <!-- 申请连接时执行 validationQuery 检测连接是否有效, 做了这个配置会降低性能。 -->
45. <property name="testOnBorrow" value="${ds_erp.testOnBorrow}" />
46.
47. <!-- 归还连接时执行 validationQuery 检测连接是否有效, 做了这个配置会降低性能 -->
48. <property name="testOnReturn" value="${ds_erp.testOnReturn}" />
49.
50. <!-- 打开 PSCache , 并且指定每个连接上 PSCache 的大小 ( Oracle 使用 ) -->
51. <property name="poolPreparedStatements"
           value="${ds_erp.poolPreparedStatements}" />
52. <property name="maxOpenPreparedStatements"
           value="${ds_erp.maxOpenPreparedStatements}" />
53.
54. <!-- 配置监控统计拦截的 filters 是否开启监控的统计功能 -->
55. <property name="filters" value="${ds_erp.filters}" />
56. </bean>
57.
58. </beans>
59.

```

3 监控

3.1 web 监控

druid 提供了 sql 语句查询时间等信息的监控功能。为了让数据库查询一直运行, 下面特地写了一个 ajax 进行轮询。同时, 还要保证在 web.xml 中配置如下信息

```

1. <servlet>
2.     <servlet-name> DruidStatView </servlet-name>
3.     <servlet-class> com.alibaba.druid.support.http.StatViewServlet </servl
           et-class>
4. </servlet>
5. <servlet-mapping>

```

```

6.         <servlet-name>   DruidStatView   </servlet-name>
7.         <url-pattern>    /druid/*      </url-pattern>
8.         </servlet-mapping>

```

老版本的 druid 的 jar 包中不支持通过 web 界面进行远程监控，从 0.2.14 开始可以通过配置 jmx 地址来获取远程运行 druid 的服务器的监控信息。具体配置方法如下：

```

1. <servlet>
2.     <servlet-name>   DruidStatView   </servlet-name>
3.
4.     <servlet-class>   com.alibaba.druid.support.http.StatViewServlet   </servlet-class>
5. </servlet>
6. <servlet-mapping>
7.     <servlet-name>   DruidStatView   </servlet-name>
8.     <url-pattern>    /druid/*      </url-pattern>
9. </servlet-mapping>
10. <session-config>
11.     <session-timeout> 120 </session-timeout>
12. </session-config>

```

启动 tomcat ：

输入：http://localhost:8080/spring-mongodb/druid/datasource.html

