## 一、背景

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API，其中，Jedis是Redis的java客户端。

spring和redis集成有利于加快开发，目前通用的方案是使用redisTemplate（类似jdbcTemplate）去做redis 的一些操作，开源的解决中间件是spring-data-redis。 鉴于spring-data-redis比较成熟，推荐使用codis来替代redis，暂不对spring-data-redis的配置进一步封装。在此，对spring-data-redis的配置使用进行说明。

## 二、配置pom文件

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring->redis</artifactId>

<version>1.4.1.RELEASE</version>

</dependency>

<dependency>

<groupId>redis.clients</groupId>

<artifactId>jedis</artifactId>

<version>2.6.1</version>

</dependency>

2.8.1版本有问题，2.8.1版本将timeout属性改成sotimeout，spring->redis调用报错

## 三、redis连接池配置说明（对应数据库的DataSource配置）

**Redis常见的配置参数**

1. **maxTotal**（老版本叫**maxActive**）：最大连接池数量

默认值：8

1. **maxIdle**：最大等待连接中的数量。超过空闲时间，数据库连接将被标记为不可用，然后被释放

默认值：8

1. **minIdle**：连接池中最小的空闲的连接数，低于这个数量会被创建新的连接（默认为0，该参数越接近maxIdle，性能越好，因为连接的创建和销毁，都是需要消耗资源的；但是不能太大，因为在机器很空闲的时候，也会创建低于minidle个数的连接，类似于jvm参数中的Xmn设置）

**默认值**：0

1. **maxWait**：获取连接时最大等待时间，单位毫秒。配置了maxWait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useUnfairLock属性为true使用非公平锁。建议配置2000ms。

**默认值**：2000ms

1. **testOnBorrow**：申请连接时执行检测连接是否有效，做了这个配置会降低性能。建议值：false

**默认值**：true

1. **testOnReturn**：归还连接时执行检测连接是否有效，做了这个配置会降低性能。

**默认值**：false

1. **testWhileIdle**：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，如果空闲时间大于timeBetweenEvictionRunsMillis，执行validationQuery检测连接是否有效。

**默认值**：false

1. **timeBetweenEvictionRunsMillis**：

a) Destroy线程定时监测的间隔， Destroy线程会检测连接的间隔时间，如果连接空闲时间大于等于minEvictableIdleTimeMillis则关闭物理连接。

b) testWhileIdle的判断依据，详细看testWhileIdle属性的说明

**默认值**：30秒

1. **minEvictableIdleTimeMillis**：连接保持空闲而不被驱逐的最长时间。

**默认值**：1分钟

其中maxTotal、minIdle、minEvictableIdleTimeMillis的配置需要根据业务的特性来配置。不根据实际情况，往大配置，会浪费Redis的资源，甚至影响其他应用。

**依据参数**：

业务所有机器的平均QPS、峰值QPS、QPS平均RT（单位s，理论上是redis的平均执行时间）、平均TPS、峰值TPS、TPS平均RT（单位s），业务机器数。

**计算公式**：

minIdle=（平均QPS\* QPS平均RT +平均TPS\* TPS平均RT）／业务机器数

maxTotal=（峰值QPS\* QPS平均RT +峰值TPS\* TPS平均RT）／业务机器数

或者maxTotal= 容器处理请求的线程池大小 （前提：业务代码中没有另起多线程）

**例：**

平均QPS=1000、峰值QPS=5000、QPS平均RT（单位s）= 0.0039、平均TPS=2000、峰值TPS=8000、TPS平均RT（单位s）=0.002、业务机器数=7

minIdle=（1000\*0.0039+2000\*0.002）／7=1

maxTotal=（5000\*0.0039+8000\*0.002）／7=6

为了稳妥起见，按两倍（或者三倍）的容量评估，则minIdle配置成2。maxTotal配置成12。

## 四、bean配置

<!-- 此处需要根据情况自己更改连接池大小/时间配置 -->

<bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig">

<property name="maxTotal" value="12"/>

<property name="maxIdle" value="12"/>

<property name="minIdle" value="2"/>

<property name="maxWaitMillis" value="2000"/>

<property name="testOnBorrow" value="true"/>

</bean>

<bean id="jedisConnectionFactory" class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory" destroy-method="destroy">

<!-- <property name="usePool" value="false"/> -->

<property name="hostName" value="{ip}"/>

<property name="port" value="{端口}"/>

<property name="password" value="{密码}"/>

<!--<property name="timeout" value="600"/>-->

<property name="database" value="0"/>

<property name="poolConfig" ref="poolConfig"/>

</bean>

<bean id="redisTemplate" class="org.springframework.data.redis.core.StringRedisTemplate">

<property name="connectionFactory" ref="jedisConnectionFactory"/>

</bean>

## 五、Codis配置

#### 1. pom配置

**JDK1.6+Spring 3.x配置**

其中jedis的版本是2.1.0，spring-data-redis的版本是1.1.0.RELEASE

<dependency>

<groupId>io.codis.jodis</groupId>

<artifactId>jodis</artifactId>

<version>0.1.1-GH</version>

</dependency>

**JDK1.8+Spring 4.x配置**

其中jedis的版本是2.8.1，spring-data-redis的版本是1.7.1.RELEASE

<dependency>

<groupId>io.codis.jodis</groupId>

<artifactId>jodis</artifactId>

<version>0.3.4-GH</version>

</dependency>

#### 2. CodisAgent配置：

<!-- JedisPool配置，jodis版本：0.3.4-GH -->

<!-- 若版本为0.1.1-GH，则将"maxTotal"改为"maxActive"、"maxWaitMillis"改为"maxWait" -->

<!-- 此处需要根据情况自己更改连接池大小/时间配置 -->

<bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig">

<property name="maxTotal" value="10"/>

<property name="maxIdle" value="5"/>

<property name="maxWaitMillis" value="300"/>

<property name="testOnBorrow" value="false"/>

</bean>

<!-- Codis的工厂类 -->

<bean id="zkJedisFactory" class="io.codis.jodis.jedis.ZkJedisFactory" init-method="init">

<property name="poolConfig" ref="poolConfig"/>

<!-- 此处配置codis的zookeeper集群 -->

<property name="zkServers" value=""/>

<!-- 此处配置codis的zookeeper 地址，需要和运维同学申请 -->

<property name="zkCodisProxyDir" value="/xmlz/xmlf"/>

<!-- 此处配置zookeeper会话的超时时间-->

<property name="zkSessionTimeoutMs" value="2000"/>

<!-- 此处配置codis链接的超时时间-->

<property name="codisConnectionTimeoutMs" value="2000"/>

<!-- 此处配置codis的超时时间-->

<property name="codisTimeoutMs" value="2000"/>

<!-- 此处配置codis 密码-->

<property name="codisPassword" value="password123456"/>

</bean>

<bean name="codisAgent" class="io.codis.jodis.CodisAgent">

<property name="pool">

<bean class="io.codis.jodis.jedis.JedisPool">

<constructor-arg index="0" ref="poolConfig"/>

<constructor-arg index="1" ref="zkJedisFactory"/>

</bean>

</property>

</bean>

//CodisAgent是线程安全的，并且命令和Jedis一样

//可以使用CodisAgent注入。

@Resource

private CodisAgent codis;

//也可以使用Jedis注入

@Resource(name = "codisAgent")

private redis.clients.jedis.Jedis codis;

//使用

String hValue = codis.hget("Key","HKey");

#### 3. Spring Data Redis配置 （使用CodisAgent配置，该配置可以忽略）

<!-- JedisPool配置，jodis版本：0.3.4-GH -->

<!-- 若版本为0.1.1-GH，则将"maxTotal"改为"maxActive"、"maxWaitMillis"改为"maxWait" -->

<!-- 此处需要根据情况自己更改连接池大小/时间配置 -->

<bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig">

<property name="maxTotal" value="10"/>

<property name="maxIdle" value="5"/>

<property name="maxWaitMillis" value="300"/>

<property name="testOnBorrow" value="false"/>

</bean>

<!-- Codis的工厂类 -->

<bean id="zkJedisFactory" class="io.codis.jodis.jedis.ZkJedisFactory" init-method="init">

<property name="poolConfig" ref="poolConfig"/>

<!-- 此处配置codis的zookeeper集群 -->

<property name="zkServers" value=""/>

<!-- 此处配置codis的zookeeper 地址，需要和运维同学申请 -->

<property name="zkCodisProxyDir" value="/xmlz/xmlf"/>

<!-- 此处配置zookeeper会话的超时时间-->

<property name="zkSessionTimeoutMs" value="2000"/>

<!-- 此处配置codis链接的超时时间-->

<property name="codisConnectionTimeoutMs" value="2000"/>

<!-- 此处配置codis的超时时间-->

<property name="codisTimeoutMs" value="2000"/>

<!-- 此处配置codis 密码-->

<property name="codisPassword" value="password123456"/>

</bean>

<bean id="jedisConnectionFactory" class="io.codis.jodis.spring.data.JedisConnectionFactory">

<constructor-arg index="0" ref="poolConfig"/>

<constructor-arg index="1" ref="zkJedisFactory"/>

</bean>

<bean id="redisTemplate" class="org.springframework.data.redis.core.StringRedisTemplate">

<property name="connectionFactory" ref="jedisConnectionFactory"/>

</bean>