链滴

记录一下花了一下午时间研究出的SpringMVC + Redis(缓存)的配置

作者：zjhch123

原文链接：https://ld246.com/article/1468830131732

来源网站：链滴

1. 使用缓存的目的是为了减少数据库的压力。

2. 必须设置缓存过期时间，防止野数据出现。

3. 缓存适用于读多写少的场合，查询时缓存命中率很低、写操作很频繁等场景不适宜用缓存。

4. 引入依赖

```
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-redis</artifactId>
  <version>1.6.0.RELEASE</version>
</dependency>
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>2.8.1</version>
</dependency>
```

spring-data-redis使用的版本不是最新版，原因是最新版的redisTemplate我一直无法成功注入。

2. 配置bean.xml

```
<cache:annotation-driven key-generator="myKeyGenerator"/>
<bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig">   <!-- Redis连接池参数 -
>
  <property name="maxIdle" value="50" />
  <property name="maxWaitMillis" value="1500" />
  <property name="testOnBorrow" value="true" />
</bean>

<bean id="jedisConnFactory"
  class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory">
  <property name="hostName" value="localhost"/>
  <property name="port" value="6379"/>
  <property name="poolConfig" ref="poolConfig"/>
</bean>

<!-- redis template definition -->
<bean id="redisTemplate" class="org.springframework.data.redis.core.RedisTemplate">
  <property name="connectionFactory" ref="jedisConnFactory"/>
</bean>

<bean id="cacheManager" class="org.springframework.data.redis.cache.RedisCacheMana
er">
  <constructor-arg type="org.springframework.data.redis.core.RedisOperations" ref="redi
Template"/>
  <property name="defaultExpiration" value="10"/> <!-- 过期时间 -->
</bean>

<bean id="myKeyGenerator" class="com.zjh.cache.CacheKeyGenerator"/> <!-- 缓存键的生
策略 -->
```

这里注意,cacheManager使用构造方法注入，设置的是type,而不是name。如果是name可能会导致出错

com.zjh.cache.CacheKeyGenerator文件

```
package com.zjh.cache;

import java.lang.reflect.Method;

import org.springframework.cache.interceptor.KeyGenerator;

public class CacheKeyGenerator implements KeyGenerator {

    @Override
    public Object generate(Object target, Method method, Object... args) {
        StringBuilder sb = new StringBuilder();
        sb.append(target.getClass().getName());
        sb.append(method.getName());
        for(Object arg : args) {
            sb.append(arg.toString());
        }
        return sb.toString();
    }

}
```

接下来就可以愉快的使用啦！