



链滴

通过订阅 redis 事件实现“时间截止后删除”功能

作者: [cando](#)

原文链接: <https://ld246.com/article/1611818297197>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

思考

情景

有一个这样的场景，某系统里为用户开辟了一个空间，这个空间在有效期里可以随意使用。但是到期要回收。我们可以通过定时任务对数据库表中存在的空间信息进行检查，如果截止时间到了，就进行应的操作。也可以把这个定时的工程扔给系统以外。

另一种思路

我们可以尝试另一种方案，例如创建空间的时候，将空间id作为key的一部分存放在redis中，而exi设为有效时间。在redis将这个超时的key删除的时候，通知我们系统，从而完成清除空间的操作。

实施

具体实现如下：

1,打开事件

1.1,修改配置文件

默认的redis并没有开启这个功能，需要修改配置文件中的notify-keyspace-events配置

```
##### Event notification #####
##

# Redis can notify Pub/Sub clients about events happening in the key space.
# This feature is documented at http://redis.io/topics/notifications
#
# For instance if keyspace events notification is enabled, and a client
# performs a DEL operation on key "foo" stored in the Database 0, two
# messages will be published via Pub/Sub:
#
# PUBLISH __keyspace@0__:foo del
# PUBLISH __keyevent@0__:del foo
#
# It is possible to select the events that Redis will notify among a set
# of classes. Every class is identified by a single character:
#
# K   Keyspace events, published with __keyspace@<db>__ prefix.
# E   Keyevent events, published with __keyevent@<db>__ prefix.
# g   Generic commands (non-type specific) like DEL, EXPIRE, RENAME, ...
# $   String commands
# l   List commands
# s   Set commands
# h   Hash commands
# z   Sorted set commands
# x   Expired events (events generated every time a key expires)
# e   Evicted events (events generated when a key is evicted for maxmemory)
# A   Alias for g$lshzxe, so that the "AKE" string means all the events.
#
# The "notify-keyspace-events" takes as argument a string that is composed
```

```
# by zero or multiple characters. The empty string means that notifications
# are disabled at all.
#
# Example: to enable list and generic events, from the point of view of the
#         event name, use:
#
# notify-keyspace-events Elg
#
# Example 2: to get the stream of the expired keys subscribing to channel
#         name __keyevent@0__:expired use:
#
# notify-keyspace-events Ex
#
# By default all notifications are disabled because most users don't need
# this feature and the feature has some overhead. Note that if you don't
# specify at least one of K or E, no events will be delivered.
notify-keyspace-events Ex
```

是的，重点就是：

```
notify-keyspace-events Ex
```

Ex代表：

```
# 以 "__keyevent@<db>__" 为前缀发布Keyevent事件，<db>为所使用的redis库编号
1,E: Keyevent events, published with __keyevent@<db>__ prefix.
# 过期事件（每次键过期时生成的事件）
2,x: Expired events (events generated every time a key expires)
```

1.2,运行时通过参数设置

```
#设置:
redis-cli: CONFIG SET notify-keyspace-events "Ex"
```

```
#查询
redis-cli:CONFIG GET notify-keyspace-event
```

#redis: 指的是进入redis-cli后，并且执行成功的情况

但是这种重启redis服务就没了（恢复为配置文件中的设置）。

2,程序方面

2.1,增加依赖包

pom.xml:

```
<!-- https://mvnrepository.com/artifact/org.springframework.data/spring-data-redis -->
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-redis</artifactId>
  <version>1.8.22.RELEASE</version>
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/redis.clients/jedis -->
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>2.10.2</version>
</dependency>
```

spring用的是4.3.7.RELEASE。需要注意匹配。

代码:

```
import org.springframework.data.redis.connection.Message;
import org.springframework.data.redis.connection.MessageListener;

public class RedisKeyExpiredMessageDelegate implements MessageListener {
    private final String EVENT_CHANNEL = "__keyevent@0__:expired";

    @Override
    public void onMessage(Message message, byte[] bytes) {
        if (EVENT_CHANNEL.equalsIgnoreCase(new String(message.getChannel()))) {
            String msg = new String(message.getBody());

            System.out.println(msg + "过期");
        }
    }
}
```

spring配置文件:

```
<!-- redis 连接工厂 -->
<bean id="redisConnectionFactory" class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory"
    p:host-name="localhost"
    p:port="6379"/>

<!-- redis消息监听 -->
<bean id="redisMessageListener"
    class="org.springframework.data.redis.listener.adapter.MessageListenerAdapter">
  <constructor-arg>
    <bean class="com.imadiaos.redis.RedisKeyExpiredMessageDelegate" />
  </constructor-arg>
</bean>
<bean id="redisContainer"
    class="org.springframework.data.redis.listener.RedisMessageListenerContainer">
  <property name="connectionFactory" ref="redisConnectionFactory" />
  <property name="messageListeners">
    <map>
      <entry key-ref="redisMessageListener">
        <list>
          <!-- <bean class="org.springframework.data.redis.listener.ChannelTopic">
            <constructor-arg value="__keyevent@1__:expired" /> </bean> -->
```

```
<!-- <bean class="org.springframework.data.redis.listener.PatternTopic">
  <constructor-arg value="*" /> </bean> -->
<bean class="org.springframework.data.redis.listener.PatternTopic">
  <constructor-arg value="__key*__:expired" />
</bean>
</list>
</entry>
</map>
</property>
</bean>
```

代码示例: github.com/imadiaos