《046\_案例实战：社交网站的网址点击追踪机制》

开胃菜，初步讲解了一下redis里的一些字符串的用法，最最简单的key-value对，加上一些数字的操作方法，增加/减少；hash，list，set，sorted set，把核心的redis的四大数据结构的各种用法结合案例讲解清楚；下周，hyperloglog、流、事务、lua、流水线、过期时间，高阶的功能，结合案例

hash数据结构，社交网站的网址点击追踪机制，类似微博

hset hash field value

hsetnx

hget hash field

社交网站（微博）一般会把你发表的一些微博里的长连接转换为短连接，这样可以利用短连接进行点击数量追踪，然后再让你进入短连接对应的长连接地址里去，所以可以利用hash数据结构去实现网址点击追踪机制

<http://t.cn/XsGGA9d> -> <http://redis.com/index.html>?dfd=sf& dfd=sf& dfd=sf& dfd=sf& dfd=sf& dfd=sf&

利用redis的incr自增长，然后10进制转36进制，接着hset存放在hash数据结构里，再提供一个映射转换的hget获取方法，hash数据结构，说白了就是我们的Java里的HashMap，redis里的hash就是一个map，一个map里可以放一些key-value对

short\_url\_access\_count: {

<http://t.cn/XsGGA9d>: 152,

<http://t.cn/I93yUUaF>: 269

}

public class Ten2Thirty {

private static final String X36 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";

private static final String[] X36\_ARRAY = "0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z".split(",");

public static String tenTo36(int num) {

StringBuffer sBuffer = new StringBuffer();

if(num == 0) {

sBuffer.append("0");

}

while(num > 0) {

sBuffer.append(X36\_ARRAY[num % 36]);

num = num / 36;

}

return sBuffer.reverse().toString();

}

}