获取字符串键值

# 1、目前用到的的方法

## 1、获取唯一UUid

## 2、获取随机字符数字大小写混合字符串，不保证唯一 （包含字符和数字）

## 3、获取字符大小写混合字符串，不告知唯一（包含字符）

import org.apache.commons.lang3.RandomStringUtils;  
  
import java.util.Date;  
import java.util.UUID;  
  
/\*\*  
 \* Created by j.sh on 2015/6/30.  
 \*/  
public class UUIDGenerator {  
  
 private UUIDGenerator(){  
 }  
  
 /\*\*  
 \* 唯一uuid  
 \* @return  
 \*/  
 public static String generate(){  
 return UUID.randomUUID().toString().replace("-","");  
 }  
  
 /\*\*  
 \* 随机字符数字大小写混合字符串，不保证唯一  
 \* @param count  
 \* @return  
 \*/  
 public static String generate(int count){  
 return RandomStringUtils.random(count,true,true);  
 }  
  
 /\*\*  
 \* 随机字符大小写混合字符串，不保证唯一  
 \* @param count  
 \* @return  
 \*/  
 public static String generateAlpha(int count){  
 return RandomStringUtils.randomAlphabetic(count);  
 }  
  
 public static String generateYm(int count){  
 return DateHelper.convertDate2String(new Date(),DateHelper.***YYYYMM***) + UUIDGenerator.generate(count);  
 }  
  
 public static String generateDateYm(Date point,int count){  
 return DateHelper.convertDate2String(point,DateHelper.***YYYYMM***) + UUIDGenerator.generate(count);  
 }  
  
 /\*\*  
 \* 获取一个hexString  
 \* 当前毫秒数乘以1000再加上1000以内随机数的16进制。不保证唯一  
 \* 用于appId生成  
 \* @return  
 \*/  
 public static String generateHexString(){  
 return Long.toHexString(new Date().getTime() \* 1000 + new Double(Math.random() \* 1000).longValue());  
 }  
  
 public static void main(String[] args) {  
 System.***out***.println(generateAlpha(16));  
 }  
}

# 2、使用

## 1、while制作唯一id

private String getAppId(){  
 String appid = "";  
 while(true){  
 appid = UUIDGenerator.generate(16);  
 SkinAppInfo info = skinAppInfoDAO.findOne(appid);  
 if(info == null){  
 break;  
 }  
  
 }  
 return appid;  
}

# 3、订单编号

package com.duodian.youhui.admin.utils.wechat;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Random;  
  
/\*\*  
 \* @Desc: \* 订单编码码生成器，生成32位数字编码，  
 \* @生成规则 1位单号类型+17位时间戳+14位(用户id加密&随机数)  
 \* @Author HealerJean  
 \* @Date 2018/8/21 下午2:43.  
 \*/  
  
public class OrderCoderUtil {  
  
  
 /\*\* 订单类别头 \*/  
 private static final String ***ORDER\_CODE*** = "1";  
 /\*\* 退货类别头 \*/  
 private static final String ***RETURN\_ORDER*** = "2";  
 /\*\* 退款类别头 \*/  
 private static final String ***REFUND\_ORDER*** = "3";  
 /\*\* 未付款重新支付别头 \*/  
 private static final String ***AGAIN\_ORDER*** = "4";  
 /\*\* 随即编码 \*/  
 private static final int[] ***r*** = new int[]{7, 9, 6, 2, 8 , 1, 3, 0, 5, 4};  
 /\*\* 用户id和随机数总长度 \*/  
 private static final int ***maxLength*** = 14;  
  
  
  
  
 /\*\*  
 \* 生成订单单号编码  
 \* @param userId  
 \*/  
 public static String getOrderCode(Long userId){  
 return ***ORDER\_CODE*** + getCode(userId);  
 }  
  
  
 /\*\*  
 \* 生成退货单号编码  
 \* @param userId  
 \*/  
 public static String getReturnCode(Long userId){  
 return ***RETURN\_ORDER*** + getCode(userId);  
 }  
  
  
 /\*\*  
 \* 生成退款单号编码  
 \* @param userId  
 \*/  
 public static String getRefundCode(Long userId){  
 return ***REFUND\_ORDER*** + getCode(userId);  
 }  
  
 /\*\*  
 \* 未付款重新支付  
 \* @param userId  
 \*/  
 public static String getAgainCode(Long userId){  
 return ***AGAIN\_ORDER*** + getCode(userId);  
 }  
  
  
  
 /\*\*  
 \* 更具id进行加密+加随机数组成固定长度编码  
 \*/  
 private static String toCode(Long id) {  
 String idStr = id.toString();  
 StringBuilder idsbs = new StringBuilder();  
 for (int i = idStr.length() - 1 ; i >= 0; i--) {  
 idsbs.append(***r***[idStr.charAt(i)-'0']);  
 }  
 return idsbs.append(getRandom(***maxLength*** - idStr.length())).toString();  
 }  
  
 /\*\*  
 \* 生成时间戳  
 \*/  
 private static String getDateTime(){  
 DateFormat sdf = new SimpleDateFormat("yyyyMMddHHmmssSSS");  
 return sdf.format(new Date());  
 }  
  
 /\*\*  
 \* 生成固定长度随机码  
 \* @param n 长度  
 \*/  
 private static long getRandom(long n) {  
 long min = 1,max = 9;  
 for (int i = 1; i < n; i++) {  
 min \*= 10;  
 max \*= 10;  
 }  
 long rangeLong = (((long) (new Random().nextDouble() \* (max - min)))) + min ;  
 return rangeLong;  
 }  
  
 /\*\*  
 \* 生成不带类别标头的编码  
 \* @param userId  
 \*/  
 private static synchronized String getCode(Long userId){  
 userId = userId == null ? 10000 : userId;  
 return getDateTime() + toCode(userId);  
 }  
  
  
  
  
  
}