我不是高手，我只是一个普通的程序员，我特别愿意大家留言讨论，批评指正，您给我指正了，我就去查资料，去做实验，我技术就能得到提高，我认为这是一个程序员的基本素养。

接上一篇

摘要算法常用的里除了MD5，就是S
HA了。S
HA又分SHA0,1,2其中SHA0，SHA1都是因为强度和算法不够有被破解的风险，最常用的就是SHA2了，名称来自于安全散列算法2（英语：Secure Hash Algorithm 2）的缩写，一种密码散列函数算法标准，由美国国家安全局研发，由
美国国家标准与技术研究院
（NIST）在2001年发布。属于SHA算法
之一，是SHA-1的后继者。其下又可再分为六个不同的算法标准，包括了：SHA-224、SHA-256、SHA-384、SHA-512、SHA-512/224、SHA-512/256。这些变体除了生成摘要的长度、循环运行的次数等一些细微差异之外，基本结构是一致的。
算法的原理很复杂，可以负责任地告诉你们，我也不懂，但是具体的应用和使用场景和具体的使
用的代码，我日常工作中用到的我就分享给大家。其中最常用的就是SHA256。

具体的算法我不懂，但是我表面理解的是MD5会生成128位散列值生成32长度的字符串（128bit，一个byte是8个bit，所以一共就是16个byte，16个byte用hex以后就是32个字符串），百度显示
2004年，证实MD5算法无法防止碰撞（collision），就是会不一样的数据得到的MD5值会一样，当然概率极极极极低，一般的系统是遇不到的，对于需要高度安全性的数据，专家一般建议改用其他算法
，如SHA-2系列
。因为SHA256会生成256位散列值64位长度的字符串，安全性就更高了，当然除了强度大了，散列函数跟MD5的也不一样了，据说是没有漏洞，又据说在量子计算机面前不堪一击，那不是我们这种程序员该考虑的事了。

下面干货来了，贴代码

```
public static String getSHA256(String str) {        MessageD
igest messageDigest;        String encodeStr = "";        tr
```

```
y {            messageDigest = MessageDigest.getInstance("SH
A-256");            messageDigest.update(str.getBytes("UTF-8
"));            byte[] bytes = messageDigest.digest();
      StringBuilder sb = new StringBuilder();            Str
ing temp = null;            for (byte aByte : bytes) {
          // temp = Integer.toHexString(aByte & 0xFF);
          // if (temp.length() == 1) {                //  //
 1?????????0??                //  sb.append("0");
    // }                temp = Integer.toHexString((0x000000
FF & aByte) | 0xFFFFFF00).substring(6);
           sb.append(temp);            }            encodeSt
r = sb.toString();        } catch (NoSuchAlgorithmException
| UnsupportedEncodingException e) {            e.printStackT
race();        }        return encodeStr;    }
```