加密与解密，我也想不出其他更好的名字。

***MD5***

Message Digest Algorithm 5，消息摘要算法第五版。

**MD5**用于确保信息传输完整一致。为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。该算法的文件号为RFC 1321（R.Rivest,MIT Laboratory for Computer Science and RSA Data Security Inc. April 1992）。是计算机广泛使用的杂凑算法之一（又译[摘要算法](https://baike.baidu.com/item/%E6%91%98%E8%A6%81%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/MD5/_blank)、[哈希算法](https://baike.baidu.com/item/%E5%93%88%E5%B8%8C%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/MD5/_blank)），主流编程语言普遍已有MD5实现。将数据（如汉字）运算为另一固定长度值，是杂凑算法的基础原理，MD5的前身有MD2、[MD3](https://baike.baidu.com/item/MD3" \t "https://baike.baidu.com/item/MD5/_blank)和[MD4](https://baike.baidu.com/item/MD4" \t "https://baike.baidu.com/item/MD5/_blank)。

MD5算法具有以下特点：

1、压缩性：任意长度的数据，算出的MD5值长度都是固定的。

2、容易计算：从原数据计算出MD5值很容易。

3、抗修改性：对原数据进行任何改动，哪怕只修改1个字节，所得到的MD5值都有很大区别。

4、强抗碰撞：已知原数据和其MD5值，想找到一个具有相同MD5值的数据（即伪造数据）是非常困难的。

MD5的作用是让大容量信息在用[数字签名](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D" \t "https://baike.baidu.com/item/MD5/_blank)软件签署私人[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5" \t "https://baike.baidu.com/item/MD5/_blank)前被"[压缩](https://baike.baidu.com/item/%E5%8E%8B%E7%BC%A9/13032501" \t "https://baike.baidu.com/item/MD5/_blank)"成一种保密的格式（就是把一个任意长度的字节串变换成一定长的[十六进制](https://baike.baidu.com/item/%E5%8D%81%E5%85%AD%E8%BF%9B%E5%88%B6" \t "https://baike.baidu.com/item/MD5/_blank)数字串）。除了MD5以外，其中比较有名的还有[sha-1](https://baike.baidu.com/item/sha-1" \t "https://baike.baidu.com/item/MD5/_blank)、[RIPEMD](https://baike.baidu.com/item/RIPEMD" \t "https://baike.baidu.com/item/MD5/_blank)以及Haval等。

摘自百度百科。

***Base64***

Base64是网络上最常见的用于传输8Bit[字节代码](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82%E4%BB%A3%E7%A0%81" \t "https://baike.baidu.com/item/base64/_blank)的编码方式之一，大家可以查看RFC2045～RFC2049，上面有MIME的详细规范。Base64编码可用于在[HTTP](https://baike.baidu.com/item/HTTP" \t "https://baike.baidu.com/item/base64/_blank)环境下传递较长的标识信息。例如，在Java Persistence系统Hibernate中，就采用了Base64来将一个较长的唯一[标识符](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%AC%A6" \t "https://baike.baidu.com/item/base64/_blank)（一般为128-bit的UUID）编码为一个字符串，用作HTTP[表单](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95" \t "https://baike.baidu.com/item/base64/_blank)和HTTP GET URL中的参数。在其他应用程序中，也常常需要把二进制[数据编码](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BC%96%E7%A0%81" \t "https://baike.baidu.com/item/base64/_blank)为适合放在URL（包括隐藏[表单域](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95%E5%9F%9F" \t "https://baike.baidu.com/item/base64/_blank)）中的形式。此时，采用Base64编码具有不可读性，即所编码的数据不会被人用肉眼所直接看到。

以上摘自百度百科

########################### URL ##############################

代码1：

String placesSearchStr="https://maps.googleapis.com/maps/api/place/nearbysearch/"

+"json?"

+"radius=5000&sensor=true"

+"&types=food|bar|church|museum|art\_gallery"

+"&key=AIzaSyDdMnQpqT9pr-k6VhwesT1OBAg\_qkvflxU";

代码2：

String latVal=String.valueOf(lat);String lngVal=String.valueOf(lng);String url;try {

url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location="

+URLEncoder.encode(latVal, "UTF-8")

+"&radius="

+URLEncoder.encode("5000", "UTF-8")

来自StackOverflow，这种问题我也遇到过，代码1那种写法会报这种异常。

java.lang.IllegalArgumentException: Illegal character in query at index 135:

因为我url后会跟token，token里可能包含“=”而“=”在url中是有特殊含义的，代表参数值。所以要对参数值进行编码。我起初是对整个url进行编码，但这样一来会把所有的“=”编码成“%3D”。这就扯淡了，键值对都不存在，url没法解析，所以还是会报错。