# IP utils

## com.sf.vishnu.common.utility. IpUtils

（在文末补充了一些简单的掩码计算）

几个常用的方法：

**1.以下两个方法用来判断两个IP是不是出于同一个网段**。

public static boolean isBelongSegment(String ip, String targetIp, String subnetMask)

public static boolean isBelongSegment(String ip, String targetIp, int subnetMask)

这两个方法只有第三个参数不同，即子网掩码不用，一个为IP类型的子网掩码，一个为数字类型的子网掩码，如果将一个数字类型的字符串传入方法1则返回false,所以在使用是请根据不同的子网掩码选择。

使用场景：自动代理的选择，IP分配时的网段选择匹配

**2.以下方法实现IP类型的子网掩码转为数字类型的子网掩码互相转换。**

public static int maskStr2InetMask(String maskStr)

public static String intToMask(int bitMask)

（在文末补充了所有IP类型掩码与数字类型掩码的对应关系，其实就是进制转换）

3.以下方法获取网段之内的所有IP。

public static List<String> parseIpMaskRange(String ip, String mask)

4. 以下方法存储了所有掩码关系，便于使用。

private static Map<String, String> maskBitMap()

## 补充

子网掩码是用来判断任意两台计算机的IP地址是否属于同一子网络的根据。最为简单的理解就是两台计算机各自的IP地址与子网掩码进行AND运算后，如果得出的结果是相同的，则说明这两台计算机是处于同一个子网内的。

eg:

I P 地址 192.168.0.1 子网掩码 255.255.255.0

转化为二进制进行运算

I P地址 11010000. 10101000. 00000000. 00000001

子网掩码 11111111. 11111111. 11111111. 00000000

AND运算 11010000. 10101000. 00000000. 00000000

192.168.0.0

前面为掩码计算后的数字（经常使用，其实就是进制转换），后面为IP格式的子网掩码

eg: 子网掩码为：11111111 11111111 11111111 10000000 即255.255.255.128。即25

为了方便，以下列出了所有的子网掩码。

maskBit.put("1", "128.0.0.0");

maskBit.put("2", "192.0.0.0");

maskBit.put("3", "224.0.0.0");

maskBit.put("4", "240.0.0.0");

maskBit.put("5", "248.0.0.0");

maskBit.put("6", "252.0.0.0");

maskBit.put("7", "254.0.0.0");

maskBit.put("8", "255.0.0.0");

maskBit.put("9", "255.128.0.0");

maskBit.put("10", "255.192.0.0");

maskBit.put("11", "255.224.0.0");

maskBit.put("12", "255.240.0.0");

maskBit.put("13", "255.248.0.0");

maskBit.put("14", "255.252.0.0");

maskBit.put("15", "255.254.0.0");

maskBit.put("16", "255.255.0.0");

maskBit.put("17", "255.255.128.0");

maskBit.put("18", "255.255.192.0");

maskBit.put("19", "255.255.224.0");

maskBit.put("20", "255.255.240.0");

maskBit.put("21", "255.255.248.0");

maskBit.put("22", "255.255.252.0");

maskBit.put("23", "255.255.254.0");

maskBit.put("24", "255.255.255.0");

maskBit.put("25", "255.255.255.128");

maskBit.put("26", "255.255.255.192");

maskBit.put("27", "255.255.255.224");

maskBit.put("28", "255.255.255.240");

maskBit.put("29", "255.255.255.248");

maskBit.put("30", "255.255.255.252");

maskBit.put("31", "255.255.255.254");

maskBit.put("32", "255.255.255.255");