* **无线网络接入类型**
* 中国联通
* 3gnet 中国联通拨号网络: 可以直接连接互联网
* 3gwap 沃3G手机上网:通过代理连接互联网
* 3gnet 沃3G连接互联网:同第一个
* 3gwap 中国联通彩信: 彩信连接
* 中国移动
* cmwap 移动梦网
* cmnet 中国移动连接互联网
* cmwap 中国移动彩信
* 中国电信
* ctwap
* ctnet
* ctwap

Wap网络类型:

中国移动,联通,电信的wap网络连接,通常都使用代理连接.

移动:10.0.0.172 端口80

联通:10.0.0.172 端口80

电信:10.0.0.200 端口80

Net网络类型:

直接连接互联网,不需要代理.

* 基于以上,在Android平台上通过2/3G访问网络,需要进行一定的处理:
  + Step1 判断连接类型是否为wap
  + Step2 如果是wap,根据不同的移动商, 设置连接代理
  + Step3 连接网络

**程序事例**

|  |
| --- |
| //在访问网络时根据当前连接类型,设置不同代理.  **DefaultHttpClient client = new DefaultHttpClient();**  //判断是否是中国电信wap上网  **if (*checkNetworkType(con) == TYPE\_CT\_WAP) {***  HttpHost proxy = **new HttpHost("10.0.0.200", 80);**  client.getParams().setParameter(ConnRoutePNames.*DEFAULT\_PROXY,*  proxy);  }  //判断是否是中国移动或者联通上网  **else if (*checkNetworkType(con) == TYPE\_CM\_CU\_WAP) {***  HttpHost proxy = **new HttpHost("10.0.0.172", 80);**  client.getParams().setParameter(ConnRoutePNames.*DEFAULT\_PROXY,*  proxy);  } |

|  |
| --- |
| **public** **static** **final** String *CTWAP* = "ctwap";  **public** **static** **final** String *CMWAP* = "cmwap";  **public** **static** **final** String *WAP\_3G* = "3gwap";  **public** **static** **final** String *UNIWAP* = "uniwap";  **public** **static** **final** **int** *TYPE\_NET\_WORK\_DISABLED* = 0;// 网络不可用  **public** **static** **final** **int** *TYPE\_CM\_CU\_WAP* = 4;// 移动联通wap10.0.0.172  **public** **static** **final** **int** *TYPE\_CT\_WAP* = 5;// 电信wap 10.0.0.200  **public** **static** **final** **int** *TYPE\_OTHER\_NET* = 6;// 电信,移动,联通,wifi 等net网络  **public** **static** Uri *PREFERRED\_APN\_URI* = Uri  .*parse*("content://telephony/carriers/preferapn");  /\*\*\*  \* 判断Network具体类型（联通移动wap，电信wap，其他net）  \*  \* \*/  **public** **static** **int** checkNetworkType(Context mContext) {  **try** {  **final** ConnectivityManager connectivityManager = (ConnectivityManager) mContext  .getSystemService(Context.*CONNECTIVITY\_SERVICE*);  **final** NetworkInfo mobNetInfoActivity = connectivityManager  .getActiveNetworkInfo();  **if** (mobNetInfoActivity == **null** || !mobNetInfoActivity.isAvailable()) {  // 注意一：  // NetworkInfo 为空或者不可以用的时候正常情况应该是当前没有可用网络，  // 但是有些电信机器，仍可以正常联网，  // 所以当成net网络处理依然尝试连接网络。  // （然后在socket中捕捉异常，进行二次判断与用户提示）。  //Log.i("", "=====================>无网络");  **return** *TYPE\_OTHER\_NET*;  } **else** {  // NetworkInfo不为null开始判断是网络类型  **int** netType = mobNetInfoActivity.getType();  **if** (netType == ConnectivityManager.*TYPE\_WIFI*) {  // wifi net处理  // Log.i("", "=====================>wifi网络");  **return** *TYPE\_OTHER\_NET*;  } **else** **if** (netType == ConnectivityManager.*TYPE\_MOBILE*) {  // 注意二：  // 判断是否电信wap:  // 不要通过getExtraInfo获取接入点名称来判断类型，  // 因为通过目前电信多种机型测试发现接入点名称大都为#777或者null，  // 电信机器wap接入点中要比移动联通wap接入点多设置一个用户名和密码,  // 所以可以通过这个进行判断！  **boolean** isType\_Ct\_Wap = **false**;  **final** Cursor c = mContext.getContentResolver().query(  *PREFERRED\_APN\_URI*, **null**, **null**, **null**, **null**);  **if** (c != **null**) {  c.moveToFirst();  **final** String user = c.getString(c  .getColumnIndex("user"));  **if** (!TextUtils.*isEmpty*(user)) {  // Log.i("",  // "=====================>代理："  // + c.getString(c  // .getColumnIndex("proxy")));  **if** (user.startsWith(*CTWAP*)) {  //Log.i("", "=====================>电信wap网络");  isType\_Ct\_Wap = **true**;  }  }  }  c.close();  **if** (isType\_Ct\_Wap) {  **return** *TYPE\_CT\_WAP*;  }  // 注意三：  // 判断是移动联通wap:  // 其实还有一种方法通过getString(c.getColumnIndex("proxy")获取代理ip  // 来判断接入点，10.0.0.172就是移动联通wap，10.0.0.200就是电信wap，但在  // 实际开发中并不是所有机器都能获取到接入点代理信息，例如魅族M9 （2.2）等...  // 所以采用getExtraInfo获取接入点名字进行判断  String netMode = mobNetInfoActivity.getExtraInfo();  //Log.i("", "netMode ================== " + netMode);  **if** (netMode != **null**) {  // 通过apn名称判断是否是联通和移动wap  netMode = netMode.toLowerCase();  **if** (netMode.equals(*CMWAP*) || netMode.equals(*WAP\_3G*)  || netMode.equals(*UNIWAP*)) {  //Log.i("", "=====================>移动联通wap网络");  **return** *TYPE\_CM\_CU\_WAP*;  }  }  }  }  } **catch** (Exception ex) {  ex.printStackTrace();  **return** *TYPE\_OTHER\_NET*;  }  **return** *TYPE\_OTHER\_NET*;  } |