[**android获取IP地址和MAC地址的方法**](http://blog.csdn.net/z_guijin/article/details/21460469)

2014-03-18 15:04 3913人阅读 [评论](http://blog.csdn.net/z_guijin/article/details/21460469#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/z_guijin/article/details/21460469#report)

获取Mac地址实际项目中测试了如下几种方法:  
(1)设备开通Wifi连接，获取到网卡的MAC地址(但是不开通wifi，这种方法获取不到Mac地址，这种方法也是网络上使用的最多的方法)

//根据Wifi信息获取本地Mac

public static String getLocalMacAddressFromWifiInfo(Context context){

WifiManager wifi = (WifiManager) context.getSystemService(Context.WIFI\_SERVICE);

WifiInfo info = wifi.getConnectionInfo();

return info.getMacAddress();

}

(2)调用Linux的busybox，通过linux命令来获取

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

//根据busybox获取本地Mac

public static String getLocalMacAddressFromBusybox(){

String result = "";

String Mac = "";

result = callCmd("busybox ifconfig","HWaddr");

//如果返回的result == null，则说明网络不可取

if(result==null){

return "网络出错，请检查网络";

}

//对该行数据进行解析

//例如：eth0 Link encap:Ethernet HWaddr 00:16:E8:3E:DF:67

if(result.length()>0 && result.contains("HWaddr")==true){

Mac = result.substring(result.indexOf("HWaddr")+6, result.length()-1);

Log.i("test","Mac:"+Mac+" Mac.length: "+Mac.length());

/\*if(Mac.length()>1){

Mac = Mac.replaceAll(" ", "");

result = "";

String[] tmp = Mac.split(":");

for(int i = 0;i<tmp.length;++i){

result +=tmp[i];

}

}\*/

result = Mac;

Log.i("test",result+" result.length: "+result.length());

}

return result;

}

private static String callCmd(String cmd,String filter) {

String result = "";

String line = "";

try {

Process proc = Runtime.getRuntime().exec(cmd);

InputStreamReader is = new InputStreamReader(proc.getInputStream());

BufferedReader br = new BufferedReader (is);

//执行命令cmd，只取结果中含有filter的这一行

while ((line = br.readLine ()) != null && line.contains(filter)== false) {

//result += line;

Log.i("test","line: "+line);

}

result = line;

Log.i("test","result: "+result);

}

catch(Exception e) {

e.printStackTrace();

}

return result;

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

(3)调用android 的API： NetworkInterface. getHardwareAddress ()  
该API的level为9，只有android 2.3以上才有该接口

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

//根据IP获取本地Mac

public static String getLocalMacAddressFromIp(Context context) {

String mac\_s= "";

try {

byte[] mac;

NetworkInterface ne=NetworkInterface.getByInetAddress(InetAddress.getByName(getLocalIpAddress()));

mac = ne.getHardwareAddress();

mac\_s = byte2hex(mac);

} catch (Exception e) {

e.printStackTrace();

}

return mac\_s;

}

public static String byte2hex(byte[] b) {

StringBuffer hs = new StringBuffer(b.length);

String stmp = "";

int len = b.length;

for (int n = 0; n < len; n++) {

stmp = Integer.toHexString(b[n] & 0xFF);

if (stmp.length() == 1)

hs = hs.append("0").append(stmp);

else {

hs = hs.append(stmp);

}

}

return String.valueOf(hs);

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

其中getLocalIpAddress是获取本地IP地址

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

//获取本地IP

public static String getLocalIpAddress() {

try {

for (Enumeration<NetworkInterface> en = NetworkInterface

.getNetworkInterfaces(); en.hasMoreElements();) {

NetworkInterface intf = en.nextElement();

for (Enumeration<InetAddress> enumIpAddr = intf

.getInetAddresses(); enumIpAddr.hasMoreElements();) {

InetAddress inetAddress = enumIpAddr.nextElement();

if (!inetAddress.isLoopbackAddress()) {

return inetAddress.getHostAddress().toString();

}

}

}

} catch (SocketException ex) {

Log.e("WifiPreference IpAddress", ex.toString());

}

return null;

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

获取本地IP地址  
在网络上搜索一下，一般就有如下的代码:

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

//获取本地IP

public static String getLocalIpAddress() {

try {

for (Enumeration<NetworkInterface> en = NetworkInterface

.getNetworkInterfaces(); en.hasMoreElements();) {

NetworkInterface intf = en.nextElement();

for (Enumeration<InetAddress> enumIpAddr = intf

.getInetAddresses(); enumIpAddr.hasMoreElements();) {

InetAddress inetAddress = enumIpAddr.nextElement();

if (!inetAddress.isLoopbackAddress()) {

return inetAddress.getHostAddress().toString();

}

}

}

} catch (SocketException ex) {

Log.e("WifiPreference IpAddress", ex.toString());

}

return null;

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

但是经过测试该方法在android2.3, 2.2...较老版本有效，但是在android较新版本(例如4.0等)获取的数据不正确。  
获取到了类似fe80::b607:f9ff:fee5:487e..这样的IP地址。经过一番努力，终于找出原因。  
上面的IP地址是IPV6的地址形式（大概这个意思，具体没有太深入研究）。解决方法是，在上面代码中的最内层的for循环的if语句中对 inetAddress进行格式判断，只有其是IPV4格式地址时，才返回值。修改后代码如下：(下面的方法也是网络上的方法，没有结果验证)

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

public String getLocalIpAddress() {

try {

String ipv4;

List nilist = Collections.list(NetworkInterface.getNetworkInterfaces());

for (NetworkInterface ni: nilist)

{

List ialist = Collections.list(ni.getInetAddresses());

for (InetAddress address: ialist){

if (!address.isLoopbackAddress() && InetAddressUtils.isIPv4Address(ipv4=address.getHostAddress()))

{

return ipv4;

}

}

}

} catch (SocketException ex) {

Log.e(LOG\_TAG, ex.toString());

}

return null;

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

网络上还有一种方法来获取本地IP地址(不过是在wifi状态下)  
通过WifiManager， DhcpInfo获取IP地址以及网关等信息（在android4.0等版本也适用）

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

package com.jason.demo.androidip;

import android.content.Context;

import android.net.DhcpInfo;

import android.net.wifi.WifiInfo;

import android.net.wifi.WifiManager;

import android.text.format.Formatter;

public class IPAddress {

public String getIPAddress(Context ctx){

WifiManager wifi\_service = (WifiManager) ctx.getSystemService(Context.WIFI\_SERVICE);

DhcpInfo dhcpInfo = wifi\_service.getDhcpInfo();

WifiInfo wifiinfo = wifi\_service.getConnectionInfo();

System.out.println("Wifi info----->"+wifiinfo.getIpAddress());

System.out.println("DHCP info gateway----->"+Formatter.formatIpAddress(dhcpInfo.gateway));

System.out.println("DHCP info netmask----->"+Formatter.formatIpAddress(dhcpInfo.netmask));

//DhcpInfo中的ipAddress是一个int型的变量，通过Formatter将其转化为字符串IP地址

return Formatter.formatIpAddress(dhcpInfo.ipAddress);

}

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

加入permission  
<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>

不过我自己在做项目过程中，用另外一种方法也解决了android4.0获取IP错误的问题:

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

//获取本地IP

public static String getLocalIpAddress() {

try {

for (Enumeration<NetworkInterface> en = NetworkInterface

.getNetworkInterfaces(); en.hasMoreElements();) {

NetworkInterface intf = en.nextElement();

for (Enumeration<InetAddress> enumIpAddr = intf

.getInetAddresses(); enumIpAddr.hasMoreElements();) {

InetAddress inetAddress = enumIpAddr.nextElement();

if (!inetAddress.isLoopbackAddress() && !inetAddress.isLinkLocalAddress()) {

return inetAddress.getHostAddress().toString();

}

}

}

} catch (SocketException ex) {

Log.e("WifiPreference IpAddress", ex.toString());

}

return null;

}

[复制代码](http://blog.csdn.net/z_guijin/article/details/21460469)

参考博文:

http://www.cnblogs.com/Amandaliu/archive/2011/11/06/2238177.html  
Android获取Mac地址

http://blog.csdn.net/ccf0703/article/details/7451274  
解决安卓4.0获取本地IP地址问题。

http://blog.csdn.net/garybook/article/details/7874456  
通过WifiManager,DhcpInfo获取android IP地址及网关等信息(两种方式)

http://blog.csdn.net/lizzydarcymsp/article/details/5623302  
利用InetAddress类确定特殊IP地址 (isLinkLocalAddress,isLoopbackAddress等)