山东大学 计算机科学与技术 学院

计算机网络 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201400130051 | 姓名：曹德福 | | 班级：2014级3班 |
| 实验题目：Android上的网络应用开发。 | | | |
| 实验学时：4 | | 日期： 2016年11月5日 | |
| 实验内容：  使用Android SDK开发环境和工具，开发一个网络程序，能把附近wifi热点的MAC地址和当前的GPS位置提交到该网址(http://to.do.com/)，要求运行在自己的手机（或者android手机模拟器）上。 | | | |
| 硬件环境：  PC机，安卓智能手机 | | | |
| 软件环境： Windows10系统，[Android Studio](http://www.baidu.com/link?url=DFw94i92-KCPVErYtcul66Ql2hFpaciIe3oJElg2AX27pm8uA5frZOVvACJ-pcGV) | | | |
| 实验步骤与内容：  一．实验内容  使用Android SDK开发环境和工具，开发一个网络程序，能把附近wifi热点的MAC地址和当前的GPS位置提交到该网址(http://to.do.com/)，要求运行在自己的手机（或者android手机模拟器）上。  二．实验过程  （1） 需求分析：  根据实验题目的要求可以把这个问题分为以下几个部分：  （a）安装Android Studio，搭建好环境。学习安卓的开发过程  （b） 获得wifi的mac地址  （c） 定位，要求只输出经纬度  （d） 通过get方式，将位置信息提交到指定的网址  （e） 图像用户界面的设计  （2） 解决方案：  根据上述的分析可以分步，分模块的解决问题：   1. 搭建实验环境:   之前有安装并使用过安卓开发环境，所以此处总结安装过程如下：  (1). 安装JDK  (2). 配置Windows上JDK的变量环境  (3). 下载安装Android Studio   1. 获取wifi的mac地址：   在android中有WifiManager类，可以通过创建相应对象并调用相应方法获得wifi接入点的各类信息。  通过调用getSystemService方法（参数为Context.WIFI\_SERVICE）来获取WifiManager的实例；  打开wifi : setWifiEnabled(true);  通过调用getScanResults()获取扫描结果,并存入list<ScanResult>中；   1. 将地址转换成位置(定位)：   由于只需要输出经纬度，故不需要引入百度地图等专用包，直接使用安卓内部的类就可以实现：通过调用getSystemService方法（参数为Context.LOCATION\_SERVICE），来获取LocationManager实例；   1. 信息的提交：   将信息整合成字符串，与我之前学习的get方法不同，Android后期不再允许与网络相关的操作添加到主线程中。也就是说，Android在与服务器实现数据访问和处理的时候，直接在UI线程内部处理是不允许的，这样会导致请求失败;因此我选择重新new了一个新线程来专门发送数据；  （e）图形用户界面的设计：  利用Android Studio的可视化编程工具对图形用户界面进行设计  （3）具体实现步骤  （a）创建project  在新生成的project中，打开activity\_main.xml，选择图  形化布局，通过鼠标的拖动可以将按钮等组件拖动到模拟屏  幕中，而在代码类中会生成相应的代码。可以在图形化界面  或者代码中对相应组件的设置进行更改。可以把图片通过拖  动添加到res/drawable文件夹下，这样就能设置相关图片:    （b）建立相关类,是响应功能:  每一个android程序都是继承自Activity，Activity是一个容器，里面可以放各种组件，并把它显示在手机屏幕上。  (C) 获取wifi MAC地址:  1、 打开wifi : setWifiEnabled(true);  2、 通过调用getScanResults()获取扫描结果,并存入list<ScanResult>中；  3、 给ListView设置适配器，展示扫描结果，并获取mac地址。  我选择是是ArrayList<ScanResult>，可以通过xml文件来显示每个item,比较自由。  具体代码如下:  **public void** GetWifi(){  **final** WifiManager wifi= (WifiManager) getSystemService(Context.WIFI\_SERVICE);  *//WifiInfo wifiInfo=wifi.getConnectionInfo();* **final** ArrayList<ScanResult> list; *//存放周围wifi热点对象的列表* list = (ArrayList<ScanResult>) wifi.getScanResults();  **for** (**int** n1=0;n1<list.size();n1++){  macString=macString+**"name"**+list.get(n1).SSID+**"mac"**+list.get(n1).BSSID;  }  **if** (list == **null**) {  Toast.makeText(**this**, **"wifi未连接！"**, Toast.LENGTH\_LONG).show();  }**else** {  listView.setAdapter(**new** MyAdapter(MainActivity.**this**,list));  }  listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  @Override  **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  ScanResult scanResult = list.get(position);  WifiInfo wifiInfor = wifi.getConnectionInfo();  wifiInfor.getMacAddress();  Toast.makeText(MainActivity.**this**,  **"Mac地址："**+scanResult.BSSID, Toast.LENGTH\_LONG).show();  }  }); }  最关键代码:  WifiManager wifi= (WifiManager) getSystemService(Context.WIFI\_SERVICE);  ArrayList<ScanResult> list = (ArrayList<ScanResult>) wifi.getScanResults();  在这里可以实现按照WIFI的信号强弱进行排序,或者选择更合适的展示相应的MAC信息的方式,我的操作截图如下:    (d)将地址转换成位置(定位)：  GPS定位有自带的一些方法，可以提供地理坐标，经度纬度,满足题目要求,所以不需要引入专用包等.  1、 通过调用getSystemService方法（参数为Context.LOCATION\_SERVICE），来获取LocationManager实例；  2、 完成LocationListener监听器需要实现的方法（主要是onLocationChanged(Location location)方法）：  通过调用Location类的getLatitude()和getLongitude()获得经纬度，并显示在定位界面中。  实现代码如下:  **public void** GetGps() {  locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);  **if** (ActivityCompat.checkSelfPermission(**this**, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(**this**, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  **return**;  }  Location location = locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER);  updateView(location);  locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 2000, 8, **new** LocationListener() {  @Override  **public void** onLocationChanged(Location location) {  updateView(location);  }  @Override  **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  }  @Override  **public void** onProviderEnabled(String provider) {  **if** (ActivityCompat.checkSelfPermission(MainActivity.**this**,  Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(MainActivity.**this**, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  **return**;  }*//获得权限* updateView(locationManager.getLastKnownLocation(provider));  }  @Override  **public void** onProviderDisabled(String provider) {  updateView(**null**);  }  });  } **public void** updateView(Location location){  **if**(location!=**null**){  StringBuffer sb = **new** StringBuffer();  sb.append(**"经度："**);  sb.append(location.getLongitude());  sb.append(**"\n纬度："**);  sb.append(location.getLatitude());  sb.append(**"\n高度："**);  sb.append(location.getAltitude());  sb.append(**" 速度："**);  sb.append(location.getSpeed());  sb.append(**"\n方向："**);  sb.append(location.getBearing());  sb.append(**" 精度："**);  sb.append(location.getAccuracy());  editText.setText(sb.toString());  gpsString=sb.toString();  }  **else** editText.setText(**"nothing"**); }  操作截图如下:    (e) 数据上传  Android向Web提交参数的共有4种方式:  1)基于http协议通过get方式提交参数；  2)基于http协议通过post方式提交参数  3)基于httpclient通过get提交参数  4)基于httpclient通过post提交参数  根据实验要求这里数据上传的实现是将按钮设置相应的监听然后用Get方法提交到网站即可。  实现代码如下:  **public void** PushToNet(){  **final** String url = **"http://121.250.223.36/submit"**;  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  String url2=url+**"?wifi-mac="**+macString.replace(**" "**, **""**)+**"&gps="**+gpsString.replace(**" "**, **""**)+**"&iam="**+**"caodefu-201400130051"**;  URL mURL = **null**;  Log.d(**"test"**,url2);  **try** {  mURL = **new** URL(url2);  HttpURLConnection conn;  conn = (HttpURLConnection) mURL.openConnection();  conn.setRequestMethod(**"GET"**);  conn.setReadTimeout(5000);  conn.setConnectTimeout(10000);   **int** responseCode = conn.getResponseCode();  **if** (responseCode == 200) {   } **else** {  Log.i(**"test"**, **"访问失败"** + responseCode);  }  } **catch** (Exception e) {  e.printStackTrace();  }   }  }).start(); }  同时为满足交互作用,添加提交按钮:  截图如下:    (f) 修改权限：获得上网与GPS权限  （4）结果展示：    上传地址界面： | | | |
| 结论分析与体会：  这次实验感觉任务量还是挺大的，对于重装系统还没有安装Android Studio的我来说，只是配置环境就花了足够长的时间。按照教程有的软件无法下载安装成功，比如添加PhoneGap库的加载在输入窗口会打印出“ERROR: Start Page at 'www/index.html' was not found”经过百度查找终于解决。下载安装时还遇到了各种各样的问题，比如版本问题，环境变量的配置等。  安装完环境后，得益于之前阅读的《第一行安卓代码》和大牛的众多博客，我开始上手实现相关方法，不得不说，合理的划分任务可以解决一时无从下手的问题。通过阅读大量代码，让我对网络中MAC和GPS的认识更进了一步。  这个实验紧跟住了现代科技的发展，它点燃了我对安卓这一领域的激情，这算是一个启蒙，让我对Android 先有了一个大概的了解，这个实验虽然不一定能包含 Android 的理念或者开发的全部，但是能够让我对 Android的设计开发有了更深的体验，让我产生浓厚的兴趣，我觉得收获还是很多的，对于我这个之前接触过安卓如今又重新上手的学生来叔哦，这个实验足够让我决心继续深入学习安卓网络相关知识。 | | | |