

## 第 10 章 流量统计

第 10 章 流量统计模块.....	2
10.1 模块概述.....	2
10.2 流量统计.....	2
10.2.1 流量统计界面 UI.....	2
10.2.2 流量统计界面 ListView 条目 UI.....	3
10.3 流量统计功能.....	5
10.3.1 流量统计之应用信息实体类.....	5
10.3.2 流量统计之初始化数据.....	5
10.3.3 流量统计之数据适配器.....	8
10.4 本章小结.....	10

## 第 10 章 流量统计模块

### ◆ 了解流量统计模块功能

### ◆ 掌握如何获取流量信息

在日常生活中，大家使用手机上网时用的基本都是 3G/4G 网络，虽然速度比 2G 网络要快很多，但也消耗了大量手机流量，并且 Android 中有些 APP 还会在后台偷偷的联网耗费流量，而每个月手机流量都是固定的，如果超过了上限会造成一定的经济损失，因此流量使用也是大家非常关心的问题，本章将针对流量统计模块进行详细讲解。

### 10.1 模块概述

流量统计模块的主要作用就是统计当月手机应用的总上传流量、本月下载流量。具体如图 10-1 所示。



流量统计		
微信	64.71 MB	
4.61 MB	60.10 MB	
天气	147 KB	
59.34 KB	88.10 KB	
WPS Office	7.66 KB	
1.93 KB	5.73 KB	
星期六 26 日历	8.69 KB	
4.33 KB	4.36 KB	
蘑菇街	63.69 MB	
2.30 MB	61.39 MB	
360手机助手	17.80 MB	
14.21 MB	3.59 MB	
音乐	287 KB	
174 KB	113 KB	
设置	588 KB	
110 KB	478 KB	

图 10-1 流量统计界面

### 10.2 流量统计

#### 10.2.1 流量统计界面 UI

观察流量统计的效果图如图 10-2 可知，对应的布局文件 `activity_traffic_manager.xml` 的具体代码如下文件【10-1】所示。



图 10-2 流量统计主界面

【文件 10-1】 res/layout/activity\_traffic\_manager.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical" >
6.     <TextView
7.         style="@style/TitleBarTextView"
8.         android:text="流量统计" />
9.     <ListView
10.        android:id="@+id/list_view"
11.        android:layout_width="match_parent"
12.        android:layout_height="match_parent" >
13.     </ListView>
14. </LinearLayout>
```

其中，TextView 是当前界面的标题，ListView 用于展示当前手机里面的所有应用程序。

## 10.2.2 流量统计界面 ListView 条目 UI

ListView 的条目如图 10-3 所示，布局文件为 item\_traffic.xml，具体的代码如文件【10-2】所示。

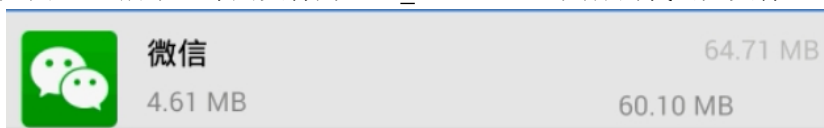


图 10-3 ListView 条目界面 UI

**【文件 10-2】res/layout/item\_traffic.xml**

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.      android:layout_width="match_parent"
4.      android:layout_height="wrap_content" >
5.      <ImageView
6.          android:id="@+id/iv_icon"
7.          android:layout_width="50dp"
8.          android:layout_height="50dp"
9.          android:layout_margin="10dp"
10.         android:src="@drawable/ic_launcher" />
11.     <TextView
12.         android:id="@+id/tv_app_name"
13.         android:layout_width="wrap_content"
14.         android:layout_height="wrap_content"
15.         android:layout_marginLeft="5dp"
16.         android:layout_marginTop="10dp"
17.         android:layout_toRightOf="@id/iv_icon"
18.         android:text="应用的名字"
19.         android:textColor="#000"
20.         android:textSize="16sp" />
21.     <TextView
22.         android:id="@+id/tv_app_snd"
23.         android:layout_width="wrap_content"
24.         android:layout_height="wrap_content"
25.         android:layout_below="@id/tv_app_name"
26.         android:layout_marginLeft="5dp"
27.         android:layout_marginTop="5dp"
28.         android:layout_toRightOf="@id/iv_icon"
29.         android:text="上传的流量"
30.         android:textColor="#66000000"
31.         android:textSize="14sp" />
32.     <TextView
33.         android:id="@+id/tv_total"
34.         android:layout_width="wrap_content"
35.         android:layout_height="wrap_content"
36.         android:layout_alignParentRight="true"
37.         android:layout_marginRight="5dp"
38.         android:layout_marginTop="10dp"
39.         android:text="总流量" />
40.     <TextView
41.         android:id="@+id/tv_app_rcv"
42.         android:layout_width="wrap_content"
43.         android:layout_height="wrap_content"
```

```
44.         android:layout_alignParentRight="true"
45.         android:layout_below="@id/tv_total"
46.         android:layout_marginRight="50dp"
47.         android:layout_marginTop="10dp"
48.         android:text="下载流量"
49.         android:textColor="#66000000" />
50. </RelativeLayout>
```

## 10.3 流量统计功能

在流量统计界面中，在界面上会显示应用程序上传和下载的流量信息，本节将针对流量统计功能进行详细讲解。

### 10.3.1 流量统计之应用信息实体类

为了便于存储应用程序信息，这里我们定义了一个实体类用于封装应用程序信息，具体代码如文件【10-3】所示。

**【文件 10-3】** com.itheima.mobilesafe\_sh2.bean/TrafficInfo

```
1. public class TrafficInfo {
2.     /**
3.      * 应用的图标
4.      */
5.     public Drawable icon;
6.     /**
7.      * 应用的名字
8.      */
9.     public String appName;
10.    /**
11.     * 下载的流量
12.     */
13.    public long rcv;
14.    /**
15.     * 上传的流量
16.     */
17.    public long snd;
18. }
```

### 10.3.2 流量统计之初始化数据

流量统计界面 TrafficManagerActivity.java 主要用于显示应用程序的上传和下载流量，首先需要进行一些初始化的操作，具体的如【文件 10-4】所示。

**【文件 10-4】 activity\_trafficmonitoring**

```
1.     public class TrafficManagerActivity extends Activity {
2.         private ListView mListView;
3.         private PackageManager mPackageManager;
4.         private List<TrafficInfo> list;
5.         @Override
6.         protected void onCreate(Bundle savedInstanceState) {
7.             // TODO Auto-generated method stub
8.             super.onCreate(savedInstanceState);
9.             initView();
10.            initData();
11.        }
12.        private void initView() {
13.            setContentView(R.layout.activity_traffic_manager);
14.            mListView = (ListView) findViewById(R.id.list_view);
15.        }
16.        private void initData() {
17.            mPackageManager = getPackageManager();
18.            ScanTask task = new ScanTask();
19.            // 开启线程
20.            task.execute();
21.            // start;
22.        }
23.        /**
24.         * 第一个参数：一般表示 url 地址。String
25.         * 第二个参数：一般表示进度条。Integer
26.         * 第三个参数：一般表示结果
27.         * 最后都会封装成一个 bean 对象
28.         */
29.        private class ScanTask extends AsyncTask<Void, Void, Void> {
30.            /**
31.             * 子线程。用来加载数据
32.             */
33.            @Override
34.            protected Void doInBackground(Void... params) {
35.                // 获取到手机上面所有的应用程序
36.                List<PackageInfo> installedPackages = mPackageManager
37.                    .getInstalledPackages(0);
38.                list = new ArrayList<TrafficInfo>();
39.                // 迭代当前的集合
40.                for (PackageInfo packageInfo : installedPackages) {
41.                    TrafficInfo info = new TrafficInfo();
42.                    Drawable icon = packageInfo.applicationInfo
43.                        .loadIcon(mPackageManager);
```

```
44.         String appName = packageInfo.applicationInfo.loadLabel(  
45.             mPackageManager).toString();  
46.         int uid = packageInfo.applicationInfo.uid;  
47.         // 某个用户程序的下载流量  
48.         // long uidRxBytes = TrafficStats.getUidRxBytes(uid);  
49.         // 获取到当前的下载流量  
50.         long uidRxBytes = getRcv(uid);  
51.         // 某个应用程序的上传流量  
52.         // long uidTxBytes = TrafficStats.getUidTxBytes(uid);  
53.         // 获取到上传流量  
54.         long uidTxBytes = getSnd(uid);  
55.         // if (uidRxBytes > 0 || uidTxBytes > 0) {  
56.         //     info.rcv = uidRxBytes;  
57.         //  
58.         //     info.snd = uidTxBytes;  
59.         // }  
60.         if(uidRxBytes != 0 && uidTxBytes != 0){  
61.             info.icon = icon;  
62.             info.appName = appName;  
63.             info.rcv = uidRxBytes;  
64.             info.snd = uidTxBytes;  
65.             list.add(info);  
66.         }  
67.     }  
68.     return null;  
69. }  
70. /**  
71.  * 回到主线程  
72.  */  
73. @Override  
74. protected void onPostExecute(Void result) {  
75.     // TODO Auto-generated method stub  
76.     super.onPostExecute(result);  
77.     TrafficManagerAdapter adapter = new TrafficManagerAdapter();  
78.     mListView.setAdapter(adapter);  
79. }  
80. }  
81. /**  
82.  * 获取下载流量  
83.  * @param uid  
84.  * @return  
85.  */  
86. public long getRcv(int uid) {
```

```
87.         File file = new File("/proc/uid_stat/" + uid + "/tcp_rcv");
88.         try {
89.             BufferedReader reader = new BufferedReader(new FileReader(file));
90.             String readLine = reader.readLine();
91.             return Long.parseLong(readLine);
92.         } catch (Exception e) {
93.             // TODO Auto-generated catch block
94.             e.printStackTrace();
95.         }
96.         return 0;
97.     }
98.     /**
99.     * 获取到上传的流量
100.    * @param uid
101.    * @return
102.    */
103.    public long getSnd(int uid) {
104.        File file = new File("/proc/uid_stat/" + uid + "/tcp_snd");
105.        try {
106.            BufferedReader reader = new BufferedReader(new FileReader(file));
107.            String readLine = reader.readLine();
108.            return Long.parseLong(readLine);
109.        } catch (Exception e) {
110.            // TODO Auto-generated catch block
111.            e.printStackTrace();
112.        }
113.        return 0;
114.    }
```

### 10.3.3 流量统计之数据适配器

前面的一些初始化操作做好之后，下面就需要将数据集合在 ListView 的数据适配器中进行展示出来，对应的代码如下所示。

```
1.     private class TrafficManagerAdapter extends BaseAdapter {
2.         private ViewHolder holder;
3.         @Override
4.         public int getCount() {
5.             return list == null ? 0 : list.size();
6.         }
7.         @Override
8.         public Object getItem(int position) {
9.             if (list != null) {
10.                 return list.get(position);
```



```
11.         }
12.         return null;
13.     }
14.     @Override
15.     public long getItemId(int position) {
16.         // TODO Auto-generated method stub
17.         return position;
18.     }
19.     @Override
20.     public View getView(int position, View convertView, ViewGroup parent) {
21.         if (convertView == null) {
22.             convertView = View.inflate(getApplicationContext(),
23.                 R.layout.item_traffic, null);
24.             holder = new ViewHolder();
25.             holder.iv_icon = (ImageView) convertView
26.                 .findViewById(R.id.iv_icon);
27.             holder.tv_app_name = (TextView) convertView
28.                 .findViewById(R.id.tv_app_name);
29.             // 上传流量
30.             holder.tv_app_snd = (TextView) convertView
31.                 .findViewById(R.id.tv_app_snd);
32.             // 总流量
33.             holder.tv_total = (TextView) convertView
34.                 .findViewById(R.id.tv_total);
35.             // 下载流量
36.             holder.tv_app_rcv = (TextView) convertView
37.                 .findViewById(R.id.tv_app_rcv);
38.             convertView.setTag(holder);
39.         } else {
40.             holder = (ViewHolder) convertView.getTag();
41.         }
42.         TrafficInfo trafficInfo = list.get(position);
43.         holder.iv_icon.setImageDrawable(trafficInfo.icon);
44.         holder.tv_app_name.setText(trafficInfo.appName);
45.         holder.tv_app_snd.setText("上传" + Formatter.formatFileSize(
46.             getApplicationContext(), trafficInfo.snd));
47.         holder.tv_app_rcv.setText("下载:" + Formatter.formatFileSize(
48.             getApplicationContext(), trafficInfo.rcv));
49.         holder.tv_total
50.             .setText(Formatter.formatFileSize(getApplicationContext(),
51.                 trafficInfo.rcv + trafficInfo.snd));
52.         return convertView;
53.     }
```

```
54.     }
55.     static class ViewHolder {
56.         ImageView iv_icon;
57.         TextView tv_app_name;
58.         TextView tv_app_snd;
59.         TextView tv_total;
60.         TextView tv_app_rcv;
61.     }
```

运行程序，效果图如下图 10-4 所示。



图 10-4 流量统计效果图

## 10.4 本章小结

本章主要是针对流量统计模块进行讲解，讲解了获取并展示流量监控的逻辑。该模块逻辑较为简单，编程者要慢慢体会并完全掌握。