

获取设备联网状态

1.1 知识解析

Android 中提供了 Apache HttpClient 库用于网络访问。同时，我们也可以使用 Java 中的网络库来访问网络，但 Android 也会将其转换成 Apache HttpClient 库来使用（deprecated from API 22）。

对于 Android 2.2 以上版本，Android 提供了 `android.net.AndroidHttpClient` 类用于网络访问，它支持 SSL 连接，并针对 Gzip 压缩做了优化。（deprecated from API 22）

另外，要让你的应用可以访问网络，必须赋予 `android.permission.INTERNET` 的权限

另外还需要注意，如果你访问的本地（localhost），需要注意应该使用 10.0.2.2 这个 IP 来访问，否则这里的 localhost 会被 Android 当成是它本身。

对于网络状态，可以通过 Android 提供的 `ConnectivityManager` 来判断。

ConnectivityManager

`ConnectivityManager` 的主要作用：

- 监控数据网络连接 (Wi-Fi、GPRS、 UMTS 等)
- 当网络连接状态发生改变时，发送广播消息
- 在连接中断的时候试图转移到其他网络连接
- 给应用提供一个查询网络状态是否可用的 API——可以提供粗粒或者细粒的消息
- 给应用提供一个请求和选择网络连接方式的 API

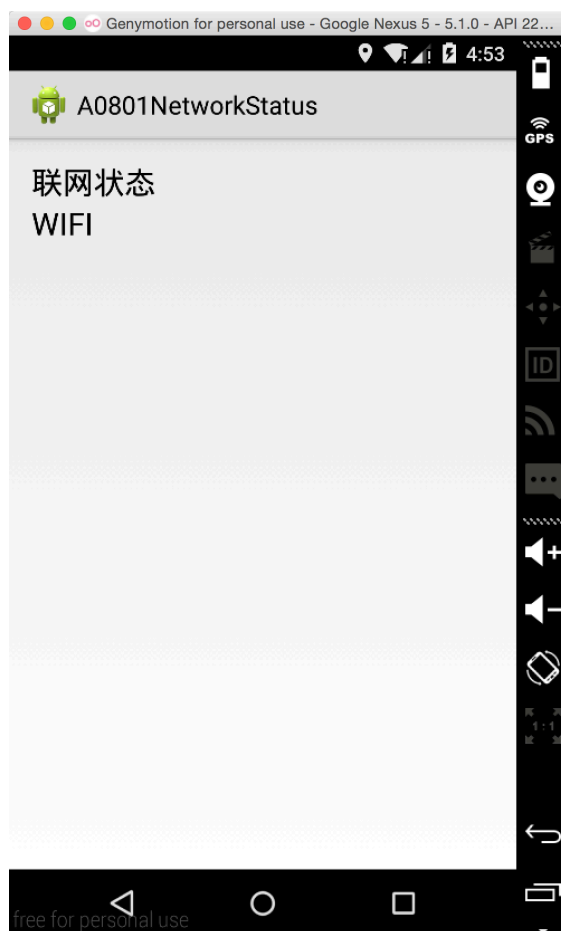
NetworkInfo

通过 `NetworkInfo`，可以得到当前所使用的网络的类型，例如：wifi 或者 mobile

另外还可以通过这个类，得到当前的网络状态，例如下图所示。

Detailed state	Coarse-grained state
IDLE	DISCONNECTED
SCANNING	CONNECTING
CONNECTING	CONNECTING
AUTHENTICATING	CONNECTING
CONNECTED	
DISCONNECTING	DISCONNECTING
DISCONNECTED	DISCONNECTED
UNAVAILABLE	DISCONNECTED
FAILED	DISCONNECTED

1.2 功能演示



1.3 实战操作

```
ConnectivityManager mgr = (ConnectivityManager)
this.getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo info = mgr.getActiveNetworkInfo();
if (info == null || !info.isConnected()) {
    tv1.setText("没有联网");
} else {
    tv1.setText("联网状态");
    String typeName = info.getTypeName();
    tv2.setText(typeName);
}
```

1.4 职业素质

检车网络状态可以应用与我们日常开发中，比如在我们打开 App 或者要请求网络的时候要检查一下是否有网络，获取大数据时候要检查网络是 3G 还是 wifi，提示用户注意流量，提升体验。