

# 技术报告

14331008 蔡嘉彬

## 1. 网络访问

执行网络交互的线程里面，我们可以使用 `URLConnection` 来执行一个 GET 类型的操作并下载数据 函数解读：`setReadTimeout()` 将读超时设置为指定的超时值，以毫秒为单位 `setConnectTimeout()` 设置连接超时时间 `setRequestMethod()` 设定 HTTP 请求的方法 `setDoInput()` 设置是否从 `URLConnection` 读入，默认情况下是 `true` `setDoOutput()` 设置是否向 `URLConnection` 输出，默认情况下是 `false` `getResponseCode()` 会返回连接的状态码（status code）。这是一种获知额外网络连接信息的有效方式。其中，状态码是 200 则意味着连接成功。

```
try {
    URL url = new URL(myurl);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(10000);
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("GET");
    conn.setDoInput(true);
    conn.connect();
    int response = conn.getResponseCode(); //状态码为200表示连接成功
    Log.d(DEBUG_TAG, "The response is: " + response);
    is = conn.getInputStream(); //获取输入流
    String contentAsString = readIt(is, len); //转换为string
    return contentAsString;
}
finally {
    if (is != null) {
        is.close();
    }
}
```

`InputStream` 是一种可读的 byte 数据源。如果我们获得了一个 `InputStream`，通常会进行解码（decode）或者转换为目标数据类型：

```

    InputStream in= connection.getInputStream();
    BufferedReader reader= new BufferedReader(new
InputStreamReader(in));
    StringBuilder response= new StringBuilder();
    String line;
    while ((line= reader.readLine())!= null) {
        response.append(line);
    }
    JSONObject obj= new JSONObject(response.toString());

```

网络操作会遇到不可预期的延迟。为了避免造成不好的用户体验 ,总是在 UI 线程之外单独的线程中执行网络操作。AsyncTask 类提供了一种简单的方式来处理这个问题:

```

private class GetList extends AsyncTask<Void, Void, Boolean> {

    @Override
    protected Boolean doInBackground(Void... integers) {...}

    @Override
    protected void onPostExecute(Boolean Success) {...}
}

```

## 2. SwipeRefreshLayout: 下拉刷新控件

SwipeRefreshLayout 只能有一个 child, 我们只需要在容器里包裹一个 ListView 就好了:

```
<android.support.v4.widget.SwipeRefreshLayout
    android:id="@+id/srl"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ListView
        android:id="@+id/lv"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</android.support.v4.widget.SwipeRefreshLayout>
```

该布局文件对应的 Activity 或其他类中获取布局 id，先设置 ListView 显示的适配器，然后再设置 SwipeRefreshLayout。

```
swipeRefreshLayout.setProgressBackgroundColorSchemeResource(android.R.color.white);
swipeRefreshLayout.setColorSchemeResources(R.color.colorAccent, R.color.colorPrimary, R.color.colorPrimaryDark);
swipeRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {
        new GetList().execute();
    }
});
```