

# AsyncTask

## 1.1 知识解析

AsyncTask 和 Handler 目的一样，都是为了在执行耗时操作的时候，不阻塞主线程（UI 线程），并在适当的时候通知 UI 线程去更新 UI，这些都是通过异步处理来完成的。

要使用 AsyncTask，需要自己编写一个类，继承 AsyncTask 类，并且实现其中的方法

AsyncTask 定义了三种泛型类型 Params，Progress 和 Result，即 AsyncTask<Params, Progress, Result>，分别表示：

- Params 启动任务执行的输入参数，比如 HTTP 请求的 URL。
- Progress 后台任务执行的百分比。
- Result 后台执行任务最终返回的结果，比如 String。

在覆盖 AsyncTask 的时候，如果对应的泛型类型不需要指定，可以定义成 Void 类型。

AsyncTask 的执行分为四个步骤，每一步都对应一个回调方法，开发者需要实现至少一个方法（doInBackground(Params...)）。这些方法都是回调方法，在任务的执行过程中，这些方法被自动调用：

onPreExecute()：该方法将在执行实际的后台操作前被 UI thread 调用。可以在该方法中做一些准备工作，如在界面上显示一个进度条。

doInBackground(Params...)：将在 onPreExecute 方法执行后马上执行，该方法运行在后台线程中。这里将主要负责执行那些很耗时的后台计算工作。该方法是抽象方法，子类必须实现。在这个方法中可以调用 publishProgress() 方法来更新实时的任务进度。

onProgressUpdate(Progress...)：在 publishProgress 方法被调用后，UI thread 将调用这个方法从而在界面上展示任务的进展情况，例如通过一个进度条进行展示。

onPostExecute(Result)：在 doInBackground 执行完成后，onPostExecute 方法将被 UI thread 调用，后台的计算结果将通过该方法传递到 UI thread。

使用 AsyncTask 类，以下是几条必须遵守的准则：

- Task 的实例必须在 UI thread 中创建
- execute 方法必须在 UI thread 中调用
- 不要手动的调用 onPreExecute(), onPostExecute(Result), doInBackground(Params...), onProgressUpdate(Progress...)这几个方法
- 该 task 只能被执行一次，否则多次调用时将会出现异常

## 1.3 实战操作

```
@Override
```

```
protected Void doInBackground(Void... params) {  
    // TODO Auto-generated method stub  
    int i = 0;  
    while (i <= 100) {  
        try {  
            Thread.sleep(200);  
        } catch (InterruptedException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
        this.publishProgress(i++);  
    }  
    return null;  
}
```

```
@Override
```

```
protected void onPostExecute(Void result) {  
    // TODO Auto-generated method stub  
    Toast.makeText(getApplicationContext(), " 下 载 完 成 ！ ",
```

```
Toast.LENGTH_LONG).show();

        super.onPostExecute(result);
    }

    @Override
    protected void onPostExecute() {
        // TODO Auto-generated method stub
        super.onPostExecute();
    }

    @Override
    protected void onProgressUpdate(Integer... values) {
        // TODO Auto-generated method stub
        tv.setText("当前进度: " + values[0]);
        pb.setProgress(values[0]);
        super.onProgressUpdate(values);
    }
}
```

## 1.4 职业素质

Android的AsyncTask比Handler更轻量级一些，适用于简单的异步处理。

首先明确Android之所以有Handler和AsyncTask，都是为了不阻塞主线程（UI线程），且UI的更新只能在主线程中完成，因此异步处理是不可避免的。

Android为了降低这个开发难度，提供了AsyncTask。AsyncTask就是一个封装过的后台任务类，顾名思义就是异步任务。

AsyncTask直接继承于Object类，位置为android.os.AsyncTask。要使用AsyncTask工作我们要提供三个泛型参数，并重载几个方法（至少重载一个）。