**Android耗时操作的ProgressDialog等待界面**

有时我们的程序需要运行一些耗费时间的操作，当进行这些操作时，我们不能让界面假死，不然用户会以为是死机或者卡住了。这时候，我们就需要通过运行 一个ProgressDialog来告诉用户：“你看，我没死，我还活着，我还能转！”。 当然，还要有Thread的配合，才能完成这个任务。  
下面是一开始我自己写的代码，虽然思路对了，但是有些繁琐：

|  |  |
| --- | --- |
| 01 | //声明变量 |
| 02 | private ProgressDialog pDialog = null; | |

|  |  |  |
| --- | --- | --- |
| 03 | Boolean startView = false; | |
| 04 | private Handler handler; |

|  |  |  |
| --- | --- | --- |
| 05 | private Thread mThread; | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | public void onCreate(Bundle savedInstanceState) { | |
| 08 | super.onCreate(savedInstanceState); |

|  |  |  |
| --- | --- | --- |
| 09 | setContentView(R.layout.main); | |
| 10 | //显示Dialog |

|  |  |
| --- | --- |
| 11 | pDialog = new ProgressDialog(this);� |
| 12 | pDialog.setProgressStyle(ProgressDialog.STYLE\_SPINNER); | |

|  |  |  |
| --- | --- | --- |
| 13 | pDialog.setMessage("Loading....");� | |
| 14 | pDialog.show(); |

|  |  |
| --- | --- |
| 15 |  |
| 16 | //定义Handler对象 | |

|  |  |
| --- | --- |
| 17 | handler = new Handler() { |
| 18 | public void handleMessage(Message msg) { | |

|  |  |  |
| --- | --- | --- |
| 19 | startView = msg.getData().getBoolean("start"); | |
| 20 | if(startView) { |

|  |  |
| --- | --- |
| 21 | //关闭Dialog |
| 22 | pDialog.dismiss(); | |

|  |  |  |
| --- | --- | --- |
| 23 | } | |
| 24 | } |

|  |  |
| --- | --- |
| 25 | }; |
| 26 | //启动线程 | |

|  |  |  |
| --- | --- | --- |
| 27 | mThread=new Thread(this); � | |
| 28 | mThread.start(); |

|  |  |
| --- | --- |
| 29 | } |
| 30 | //在下面的线程中运行耗时方法 | |

|  |  |  |
| --- | --- | --- |
| 31 | public void run() { | |
| 32 | //耗时方法 |

|  |  |  |
| --- | --- | --- |
| 33 | longTimeMethod()； | |
| 34 |  |

|  |  |
| --- | --- |
| 35 | //handler传回“准备好”的信息 |
| 36 | Message msg = handler.obtainMessage(); | |

|  |  |
| --- | --- |
| 37 | Bundle data = new Bundle(); |
| 38 | data.putBoolean("start", true); | |

|  |  |
| --- | --- |
| 39 | msg.setData(data); |
| 40 | handler.sendMessage(msg); | |

|  |  |
| --- | --- |
| 41 | } |

这样写的好处是Message可以传回不同类型的值，可以应对不同需要。后来在网上看见了一种简便方法，顿时觉得自己这只菜鸟还要好好学习啊。  
代码是这样的：

|  |  |
| --- | --- |
| 01 | //声明变量 |
| 02 | private Button b1; | |

|  |  |  |
| --- | --- | --- |
| 03 | private ProgressDialog pd; | |
| 04 |  |

|  |  |
| --- | --- |
| 05 | //定义Handler对象 |
| 06 | private Handler handler =new Handler(){ | |

|  |  |
| --- | --- |
| 07 | @Override |
| 08 | //当有消息发送出来的时候就执行Handler的这个方法 | |

|  |  |  |
| --- | --- | --- |
| 09 | public void handleMessage(Message msg){ | |
| 10 | super.handleMessage(msg); |

|  |  |  |
| --- | --- | --- |
| 11 | //只要执行到这里就关闭对话框 | |
| 12 | pd.dismiss(); |

|  |  |  |
| --- | --- | --- |
| 13 | } | |
| 14 | }; |

|  |  |
| --- | --- |
| 15 |  |
| 16 | public void onCreate(Bundle savedInstanceState) { | |

|  |  |  |
| --- | --- | --- |
| 17 | super.onCreate(savedInstanceState); | |
| 18 | setContentView(R.layout.main); |

|  |  |
| --- | --- |
| 19 | } |
| 20 |  |

|  |  |  |
| --- | --- | --- |
| 21 | private void processThread(){ | |
| 22 | //构建一个下载进度条 |

|  |  |  |
| --- | --- | --- |
| 23 | pd= ProgressDialog.show(MainHandler.this, "Load", "Loading…"); | |
| 24 | new Thread(){ |

|  |  |
| --- | --- |
| 25 | public void run(){ |
| 26 | //在新线程里执行长耗时方法 | |

|  |  |
| --- | --- |
| 27 | longTimeMethod(); |
| 28 | //执行完毕后给handler发送一个空消息 | |

|  |  |  |
| --- | --- | --- |
| 29 | handler.sendEmptyMessage(0); | |
| 30 | } |

|  |  |  |
| --- | --- | --- |
| 31 | }.start(); | |
| 32 | } |

这样的写法比较简洁，运行效率也高。