



链滴

利用 Handler 机制切换到主线程中更新 UI

作者: [xynling](#)

原文链接: <https://ld246.com/article/1542776542419>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

首先在主线程中定义Handler，Handler mainHandler = new Handler(); (必须要在主线程中定义才操作主线程，如果想在其他地方定义声明时要这样写Handler mainHandler = new Handler(Looper.getMainLooper())，来获取主线程的 Looper 和 Queue)

获取到 Handler 后就很简单了，用handler.post(Runnable r)方法把消息处理放在该 handler 依附消息队列中（也就是主线程消息队列）。

(1)：假如该方法是在子线程中

```
Handler mainHandler = new Handler(Looper.getMainLooper());
mainHandler.post(new Runnable() {
    @Override
    public void run() {
        //已在主线程中，可以更新UI
    }
});
```

Handler还有下面的方法：

- 1.postAtTime(Runnable r, long uptimeMillis); //在某一时刻发送消息
- 2.postAtDelayed(Runnable r, long delayMillis); //延迟delayMillis毫秒再发送消息

(2)：假设在主线程中

```
Handler myHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        switch(msg.what) {
            case 0:
                //更新UI等
                break;
            case 1:
                //更新UI等
                break;
            default:
                break;
        }
    }
};
```

之后可以把 mainHandler 当做参数传递在各个类之间，当需要更新UI时，可以调用sendMessage一系列方法来执行handleMessage里的操作。

假设现在在子线程了

```
/**
 *获取消息，尽量用obtainMessage()方法，查看源码发现，该方法节省内存。
 *不提倡用Messenger msg=new Messenger()这种方法，每次都去创建一个对象，肯定不节省内
 啦！
 *至于为什么该方法还存在，估计还是有存在的必要吧。（留作以后深入研究）
 */
Message msg = myHandler.obtainMessage();
```

```
msg.what = 0; //消息标识
myHandler.sendMessage(msg); //发送消息
```

如上代码，只是发送了个消息标识，并没有传其他参数。

如果想传递参数，可以这样：

```
msg.what = 1; //消息标识
msg.arg1=2; //存放整形数据，如果携带数据简单，优先使用arg1和arg2，比Bundle更节省内存。
msg.arg2=3; //存放整形数据
Bundle bundle=new Bundle();
bundle.putString("dd","adfasd");
bundle.putInt("love",5);
msg.setData(bundle);
msg.obj=bundle; //用来存放Object类型的任意对象
myHandler.sendMessage(msg); //发送消息
```

总结：msg.obj它的功能比较强大一下，至于它和利用Bundle传递数据，那个会效率高一些，更节省内存一些，个人认为从传递数据的复杂程度看，由简单到复杂依次使用，arg1，setData(), obj。会比较好一些。

作者：da_caoyuan

来源：CSDN

原文：https://blog.csdn.net/da_caoyuan/article/details/52931007

版权声明：本文为博主原创文章，转载请附上博文链接！