

I'm doing something similar in my android app; I update some data in my interface every 10 seconds. There are many ways to do this, but I chose to use a `Handler` because it's very simple to implement:

```
Thread timer = new Thread() {
    public void run () {
        for (;;) {
            // do stuff in a separate thread
            uiCallback.sendEmptyMessage(0);
            Thread.sleep(3000);    // sleep for 3 seconds
        }
    }
};
timer.start();
...
private Handler uiCallback = new Handler () {
    public void handleMessage (Message msg) {
        // do stuff with UI
    }
};
```

11 down
vote
accepted

As you may know, you cannot run periodic functions like this in the UI thread, because it will block the UI. This creates a new `Thread` that sends a message to the UI when it is done, so you can update your UI with the new results of whatever your periodic function does.

If you do not need to update the UI with the results of this periodic function, you can simply ignore the second half of my code example, and just spawn a new `Thread` as shown. Beware, however: if you are modifying variables shared by this new `Thread` and the UI, you are going to run into problems if you don't synchronize. In general, threading is not an area where you want to ignore "good programming practices" because you'll get strange, non-predictable errors and you'll be cursing your program.

-tjw