**Handler**

A Handler allows you to send and process [Message](https://developer.android.com/reference/android/os/Message.html) and Runnable objects associated with a thread's [MessageQueue](https://developer.android.com/reference/android/os/MessageQueue.html). Each Handler instance is associated with a single thread and that thread's message queue. When you create a new Handler, it is bound to the thread / message queue of the thread that is creating it -- from that point on, it will deliver messages and runnables to that message queue and execute them as they come out of the message queue.

Handler is the message processor on the worker thread.

There are two main uses for a Handler: (1) to schedule messages and runnables to be executed as some point in the future; and (2) to enqueue an action to be performed on a different thread than your own.

Scheduling messages is accomplished with the [post(Runnable)](https://developer.android.com/reference/android/os/Handler.html#post(java.lang.Runnable)), [postAtTime(Runnable, long)](https://developer.android.com/reference/android/os/Handler.html#postAtTime(java.lang.Runnable, long)), [postDelayed(Runnable, long)](https://developer.android.com/reference/android/os/Handler.html#postDelayed(java.lang.Runnable, long)), [sendEmptyMessage(int)](https://developer.android.com/reference/android/os/Handler.html#sendEmptyMessage(int)), [sendMessage(Message)](https://developer.android.com/reference/android/os/Handler.html#sendMessage(android.os.Message)), [sendMessageAtTime(Message, long)](https://developer.android.com/reference/android/os/Handler.html#sendMessageAtTime(android.os.Message, long)), and [sendMessageDelayed(Message, long)](https://developer.android.com/reference/android/os/Handler.html#sendMessageDelayed(android.os.Message, long)) methods.

When a process is created for your application, its main thread is dedicated to running a message queue that takes care of managing the top-level application objects (activities, broadcast receivers, etc) and any windows they create. You can create your own threads, and communicate back with the main application thread through a Handler.