**Asynctask**

为什么 : Android3.0开始，系统要求网络访问必须在子线程中进行，否则网络访问将会失败并抛出NetworkOnMainThreadException这个异常，这样做是为了避免主线程由于耗时操作所阻塞从而出现ANR现象

主要技术点:

AsyncTask封装了线程池和Handler。AsyncTask有两个线程池：SerialExecutor和THREAD\_POOL\_EXECUTOR。前者是用于任务的排队，默认是串行的线程池：后者用于真正的执行任务。

AsyncTask还有一个Handler，叫InternalHandler， AsyncTask内部就是通过InternalHandler来发送任务执行的进度以及执行结束等消息

AsyncTask的对象一般是执行execute（）方法来启动任务,调用async源码中的executeOnExecuto, onPreExecute()方法执行在主线程中，然后讲参数赋给了Worker对象，

结合<https://github.com/GeniusVJR/LearningNotes/blob/master/Part1/Android/Asynctask%E6%BA%90%E7%A0%81%E5%88%86%E6%9E%90.md>和<http://www.qingpingshan.com/rjbc/az/212154.html>

* *THREAD\_POOL\_EXECUTOR*   
  这个是用于任务执行的线程池 ，所有的任务都是在这个线程池中完成调用
* *SerialExecutor*   
  这个线程池用于用于任务的排队，默认实现也能看出来，它是串行执行任务。
* *FutureTask*   
  FutureTask一个可取消的异步计算, 使用子线程计算返回的结果Result作为泛型，并且将WorkerRunnable作为构造参数传入
* *WorkerRunnable*   
  真正的子线程执行单位