1. 网络通信

提到iOS网络编程层次模型分为三层：

* *Cocoa层：NSURL，Bonjour，Game Kit，WebKit*
* *Core Foundation层：基于 C 的 CFNetwork 和 CFNetServices*
* *OS层:基于 C 的 BSD socket*
  1. 获取网络状态

SystemConfiguration.framework，Reachability

* 1. CFNetwork

CFNetwork。CFNetwork 只是对 BSD socket 的进行了轻量级的封装，但在 iOS 中使用 CFNetwork 有一个显著的好处，那就是 CFNetwork 与系统级别的设置（如：天线设置）以及 run-loop 结合得很好

CFSocket类

* 1. Cocoa

AsyncSocket类是支持TCP和UDP的。  
AsyncSocket是封装了CFSocket和CFSteam的TCP/IP socket网络库。它提供了异步操作，本地cocoa类的基于delegate的完整支持。

主要有以下特性：

* 队列的非阻塞的读和写，而且可选超时。你可以调用它读取和写入，它会当完成后告知你。
* 自动的socket接收。如果你调用它接收连接，它将为每个连接启动新的实例，当然，也可以立即关闭这些连接。
* 委托（delegate）支持。错误、连接、接收、完整的读取、完整的写入、进度以及断开连接，都可以通过委托模式调用。
* 基于run loop的，而不是线程的。虽然可以在主线程或者工作线程中使用它，但你不需要这样做。它异步的调用委托方法，使用NSRunLoop。委托方法包括socket的参数，可让你在多个实例中区分。
* 同时也提供了基于GCD的实现

其实相对于前面来说，这个就是封装了底层繁琐的函数调用，通过指定委托，并让委托对象实现相关协议，就能让事件到来时调用响应的函数。而不用想CFSocket那样，要去指定回调函数。

* 1. NSULRConnection

使用NSURLConnection实现http通信的方式。NSURLConnection提供了异步请求、同步请求两种通信方式