Extra-ordinary Networking

This thread has been locked by a moderator.

45

Most apps perform ordinary network operations, like fetching an HTTP resource with `URLSession` and opening a TCP connection to a mail server with Network framework. These operations are not without their challenges, but they're the well-trodden path.

Note If your app performs ordinary networking, see TN3151 Choosing the right networking API for recommendations as to where to start.

Some apps have extra-ordinary networking requirements. For example, apps that:

- Help the user configure a Wi-Fi accessory
- Require a connection to run over a specific interface
- Listen for incoming connections

Building such an app is tricky because:

- Networking is hard in general.
- Apple devices support very dynamic networking, and your app has to work well in whatever environment it's running in.
- Documentation for the APIs you need is tucked away in man pages and doc comments.
- In many cases you have to assemble these APIs in creative ways.

If you're developing an app with extra-ordinary networking requirements, this post is for you.

Note If you have questions or comments about any of the topics discussed here, put them in a new thread here on DevForums. Make sure I see it by tagging it with… well… tags appropriate to the specific technology you're using, like *Foundation*, *CFNetwork*, *Network*, or *Network Extension*.

Links, Links, and More Links

Each topic is covered in a separate post:

- The iOS Wi-Fi Lifecycle describes how iOS joins and leaves Wi-Fi networks. Understanding this is especially important if you're building an app that works with a Wi-Fi accessory.
- Network Interface Concepts explains how Apple platforms manage network interfaces. If you've got this far, you definitely want to read this.
- Network Interface Techniques offers a high-level overview of some of the more common techniques you need when working with network interfaces.
- Network Interface APIs describes APIs and core techniques for working with network interfaces. It's referenced by many other posts.
- Running an HTTP Request over WWAN explains why most apps should not force an HTTP request to run over WWAN, what they should do instead, and what to do if you really need that behaviour.
- If you're building an iOS app with an embedded network server, see Showing Connection Information in an iOS Server for details on how to get the information to show to your user so they can connect to your server.
- Many folks run into trouble when they try to find the device's IP address, or other seemingly simple things, like the name of the Wi-Fi interface. Don't Try to Get the Device's IP Address explains why these problems are hard, and offers alternative approaches that function correctly in all network environments.
- If you're building an app that works with a Wi-Fi accessory, see Working with a Wi-Fi Accessory.

There are also some posts that are not part of this series but likely to be of interest if you're working in this space:

- Local Network Privacy FAQ discusses iOS's local network privacy feature.
- Calling BSD Sockets from Swift does what it says on the tin, that is, explain how to call BSD Sockets from Swift. When doing weird things with the network, you often find yourself having to use BSD Sockets, and that API is not easy to call from Swift. The code therein is primarily for the benefit of test projects, oh, and DevForums posts like this one.
- TN3111 iOS Wi-Fi API overview is a critical resource if you're doing Wi-Fi specific stuff on iOS.
- Networking Resources has links to many other useful resources.

Share and Enjoy
—
Quinn "The Eskimo!" @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

Network

Reply

Posted 2 days ago by eskimo

Add a Comment