What Every Junior iOS Developer Needs to Know

Gabe Nadel
writes on September 2, 2015

Developing apps for iOS has been a growing sector for some time. The versatility and ubiquity of the iPhone and iPad gave scores of developers a chance to bring their wild ideas to life and parade them in front of an enormous – and relatively wealthy – captive audience. In the initial gold rush and for some time after, there was real money to be made making technically simple apps which might appeal to niche audiences or just be good for a laugh. Indie developers were trying their hands at all sort of apps and entrepreneurs of all stripes needed coders to bring *their* next big thing to life. Ah, the salad days.

As the competition on the App Store grew fiercer and the initial infatuation with the devices began to fade, more and more of the lucrative iOS development work came from big businesses requiring a mobile presence. The upshot of this for an aspiring iOS developer is that now, your first substantial iOS job will most likely come through a more traditional channel – a recruiter, direct hire at a large firm, digital agency and the like. This no doubt comes with its benefits – a steady paycheck, for starters – though the interview and vetting process may be far more rigorous than just a few years ago.

This presents a somewhat unique problem for iOS developers as there is so much breadth to the subject. Since an iPhone can be used for so many different types of things, it's often hard to know which skills to hone first. Should I learn Core Audio before Core Data? If I want to learn about animation, which of the many options should I choose? No mere mortal can tackle it all, so where to spend one's limited time?

In this article, I hope to outline the menu of priorities a Jr. iOS Developer should focus on, so you can walk *confidently* into a junior-level job interview.

NOTE: Before anyone gets up in arms about the placement or omission of items on this list, bear in mind this is for a theoretical job, working on a theoretical project. The actual skills needed could vary wildly. We are knowingly painting with a wide brush, but hopefully will cover all the important areas, not simply the bare minimum. Also, in an effort to be even-handed, I've primarily provided links to Apple documentation – you may however find it much easier to learn this material with the guidance of a code school or via countless online tutorials. Treehouse of course does offer robust iOS tracks in both Objective-C and Swift.

The Must Haves

These are topics you should have hands-on experience with and be comfortable talking about and implementing. That hands-on work needn't be paid work and you won't be expected to recite documentation from memory, but you should have at least moderate fluency in the following topics.

- Xcode & Interface Builder
- Building to Devices
- Distributing an App (preferably for an app on the App Store)

- Fluency with Swift OR Objective-C
- Cocoa Touch
- UIKit
- Auto Layout
- Understanding of MVC
- Debugging in Xcode
- Instruments
- Xib's, Storyboards, Segues and related presentation/transition
- Core Data
- TableViews
- CollectionViews
- Parsing JSON
- Notifications, Delegation, Key Value Observation
- REST and/or SOAP
- Git, Subversion or other version control
- Understanding of Clean Coding Principles
- Understanding of SOLID
- Threading and Concurrency
- Unit Testing, XCTest (not an expert, but some experience)
- Have built for both iPhone and iPad, ideally in a universal app

The Nice-to-Haves

All of these topics and frameworks won't be expected for every project, but in all likelihood, a few will. Understanding what they are used for, where they live in the iOS landscape and, where possible, having hands-on experience, will show hiring managers that you have gone beyond the bare minimum.

- If Swift is your primary language, at least some knowledge of Objective-C
- If Obj-C is your primary language, at least some knowledge of Swift
- arc, Familiarity of the non-arc practices
- In-App Purchase

At Least A Few Of

Above and Beyond

These items are definitely not iOS 101, though can help separate you from other junior candidates and do come up in interviews and job postings. If they are skills you already have it's worth mentioning them or if they are of interest to you, it might be worth taken a little time getting your hands dirty.

Now remember, hiring managers know that you won't be able to solve all of their problems and fix all of their bugs on day one. No developer can; especially not a relative rookie. They care much more that you have the critical thinking skills to work through problems and are cognisant of your limits, yet eager to expand them. You will potentially be a custodian of a very expensive codebase or brand and they need to be able to trust that you will treat it with care and caution.

Lastly, I find devs by-and-large to be protective of their workspace and environment – and less in terms of territoriality than climate. Team dynamics are felt in a very real way, especially on highly collaborative teams. Your future teammates care about your skills, but they also care a great deal that you will be pleasant to work with and a good teammate. Don't forget to showcase *that* side of your personality in a rush to demonstrate what you know.

Now go get em.

- getting started
- junior iOS developer