

# InterviewQuestions

Mock interview questions as part of Udacity's iOS nanodegree program

## Udacity Mock Interview Questions

Paul ReFalo 25 July 2017

---

### 1. What have you learned recently about iOS development? How did you learn it? Has it changed your approach to building apps?

*I recently learned about segmented controls to sort data for the user.*

*I also learned to use `NSFetchedResultsController` with a `UISearchBar` and a predicate to filter results for display in a `UIView`. Thinking of ways to sort and organize information can make an iOS app shine. I also learned about using `IBDesignables` from watching youtube tutorials to override basic storyboard types to make some tasks easier. For examples, you can create a custom class that inherits from `UIView` and then add some inspectable properties so that they show up in the storyboard. This can make*

*some tasks easier. Here is an example on UIView border properties:*

```
@IBInspectable var cornerRadius: CGFloat = 0 {
    didSet {
        layer.cornerRadius = cornerRadius
        layer.masksToBounds = cornerRadius > 0
    }
}

@IBInspectable var borderWidth: CGFloat = 0 {
    didSet {
        layer.borderWidth = borderWidth
    }
}

@IBInspectable var borderColor: UIColor? {
    didSet {
        layer.borderColor = borderColor?.CGColor
    }
}
```

This changes how I write apps because you can quickly include custom designable files into your project to make tasks quicker and easier. This is the kind of approach that I think would work well for you company, esri. In addition to loving GIS, I am also taking online Masters level courses in Data Analytics at Oregon State University. This will help be to get and clean data on its way to becoming a better data model for apps at esri.

## 2. Can you talk about a framework that you've used recently (Apple or third-party)? What did you like/dislike about the framework?

*I recently learned about Google's Firebase pods for iOS. I used Firebase as a backend for the inventory data in my Capstone Project. Firebase allows you to manage objects in a similar way to Core Data, as managed objects. I wrote manual functions to seed and store data so users can keep their inventories in sync across devices. I look forward to using Firebase and getting better at managing data persistence.*

## 3. Describe how you would construct a Twitter feed application (here is an example of Udacity's Twitter feed) that at minimum can display a company's Twitter page. Please include information about any classes/structs that you would use in the app. Which classes/structs would be the model(s), the controller(s), and the view(s)?

*I would start by trying the following:*

- Install Twitter's API Twitter Kit with Cocoa Pods

- Use Twitter's Login for authentication
- Construct a GET request, check for results != nil and parse the resulting json
- Create a struct TweetsModel to hold the resulting tweets
- Create a class Tweets as a UITableView to display the tweets to the user

## **4. Describe some techniques that can be used to ensure that a UITableView containing many UITableViewCell is displayed at 60 frames per second.**

*First try reusing cells with dequeueReusableCellWithIdentifier.*

*Next try using AsyncDisplayKit and preload more of the data off of the main thread if possible.*

## **5. Imagine that you have been given a project that has this ActorViewController. The ActorViewController should be used to display information about an actor. However, to send information to other ViewControllers, it uses**

# NSUserDefaults. Does this make sense to you? How would you send information from one ViewController to another one?

*No, UserDefaults is not the best place for a one-to-many relationship as needed here. UserDefaults is best for simple key-dictionary relationships and not a good substitute for an Actor class or struct. With UserDefaults you can write basic types such as Bool, Float, Double, Int, String, or URL. Also, you could use Core Data and the other ViewControllers can access data as managed objects as they need it.*

Another approach would be to override prepareForSegue and pass an object like this on segue:

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if segue.identifier == "segueID" {
        if let destination = segue.destinationViewController as? SecondController {
            destination.structActorObject = self.structActorObject
        }
    }
}
```

**6. Imagine that you have been given a project that has this GithubProjectViewController. The GithubProjectViewController should be used to display high-level information about a GitHub project. However, it's also responsible for finding out if there's network connectivity, connecting to GitHub, parsing the responses and persisting information to disk. It is also one of the biggest classes in the project. Follow-up question:: How might you improve the design of this view controller?**

*Importing SystemConfiguration you can use*

*SCNetworkReachabilityCreateWithAddress and the check the resulting array contains .reachable in order to check for connectivity. Then create a GET request, check results != nil and parse the results.*

*I would break up this ViewController into a Core Data model to handle the high-level information. Also break out the network connectivity, request and parsing to a GitHub API class. In this way the original GithubProjectViewController only needs to import the managed objects and display them with appropriate navigation.*

## 7. If you were to start your iOS developer position today, what would be your goals a year from now?

*I would want to spend a year building with the basics in MVC architecture (e.g. UITableViews, UICollectionViews, Core Data, MKMapView, OAuth, RESTful APIs, etc.). Then I would want to hone my skills at UI with UIView.animate. Customizing the UI is optimizing the UI in my opinion. Getting the right information and view to the user's fingertips is iOS development. I am also excited about utilizing @IBDesignable/Inspectable. I just watched WWDC17 section on the new coreML and am excited to explore ways to improve the user experience by anticipating some of the workflow tasks (e.g. uses autocomplete to sync up with topic at hand). CoreML could also help at esri to customize the map experience for your users thereby making their user experience smoother and more pleasant.*