7.0 Lesson - Firebase 101

Paul Solt - Paul@SuperEasyApps.com
SuperEasyApps.com

7.1 Lecture - Firebase 101

In this lesson you will learn how to get started with the basics of Firebase, which you can use to store data in the cloud. It is a JSON based cloud storage, which is very fast at storing JSON based data. However, Firebase is not a relational database.

Links

- Firebase

7.2 Tutorial - Create a Firebase Account and Create Your First Project

1. Go to http://www.firebase.com to create your own Firebase account
2. Click on `Manage App` for your Your First App in the Firebase Dashboard (or create a new app)
3. Use the code download “7.2 - Firebase 101 - Begin” or start a new Xcode project
4. Copy/paste the URL for your Firebase app, as a comment in your Xcode project

```
https://<Some_Value_Here>.firebaseio.com
```

7.3 Tutorial - Setup Firebase Using Cocoapods

1. Use the Firebase Quickstart and Tutorials for Swift
2. Open Terminal (Spotlight)
3. Install Cocoapods using Ruby `sudo gem install cocoapods` in Terminal

```
Pauls-Mac-Pro:Firebase Demo paulsolt$ sudo gem install cocoapods
Password:
Fetching: nap-1.1.0.gem (100%)
Successfully installed nap-1.1.0
Fetching: cocoapods-core-0.39.0.gem (100%)
Successfully installed cocoapods-core-0.39.0
Fetching: claide-0.9.1.gem (100%)
Successfully installed claide-0.9.1
Fetching: xcodeproj-0.28.2.gem (100%)
Successfully installed xcodeproj-0.28.2
Fetching: cocoapods-downloader-0.9.3.gem (100%)
Successfully installed cocoapods-downloader-0.9.3
Fetching: cocoapods-search-0.1.0.gem (100%)
Successfully installed cocoapods-search-0.1.0
Fetching: cocoapods-stats-0.6.2.gem (100%)
Successfully installed cocoapods-stats-0.6.2
Fetching: cocoapods-try-0.5.1.gem (100%)
Successfully installed cocoapods-try-0.5.1
Fetching: cocoapods-trunk-0.6.4.gem (100%)
Successfully installed cocoapods-trunk-0.6.4
Fetching: molinillo-0.4.5.gem (100%)
Successfully installed molinillo-0.4.5
Fetching: cocoapods-0.39.0.gem (100%)
Successfully installed cocoapods-0.39.0
Parsing documentation for nap-1.1.0
Installing ri documentation for nap-1.1.0
Parsing documentation for cocoapods-core-0.39.0
Installing ri documentation for cocoapods-core-0.39.0
Parsing documentation for claide-0.9.1
Installing ri documentation for claide-0.9.1
Parsing documentation for xcodeproj-0.28.2
Installing ri documentation for xcodeproj-0.28.2
Parsing documentation for cocoapods-downloader-0.9.3
Installing ri documentation for cocoapods-downloader-0.9.3
Parsing documentation for cocoapods-search-0.1.0
Installing ri documentation for cocoapods-search-0.1.0
Parsing documentation for cocoapods-stats-0.6.2
Installing ri documentation for cocoapods-stats-0.6.2
Parsing documentation for cocoapods-try-0.5.1
Installing ri documentation for cocoapods-try-0.5.1
Parsing documentation for cocoapods-trunk-0.6.4
Installing ri documentation for cocoapods-trunk-0.6.4
Parsing documentation for molinillo-0.4.5
Installing ri documentation for molinillo-0.4.5
Parsing documentation for cocoapods-0.39.0
Installing ri documentation for cocoapods-0.39.0
11 gems installed
```

4. Create or start with an Xcode project in your Projects folder
5. Change directories in Terminal using the `cd directoryName` command (drag and drop Finder folder into Terminal)
6. The `pwd` command will print the working directory (current directory)
7. Using Terminal in your Xcode project directory run the commands

```
Pauls-Mac-Pro:Firebase Demo paulsolt$ pod init
Pauls-Mac-Pro:Firebase Demo paulsolt$ open -a Xcode Podfile
```

. Delete everything in the Podfile

8. Add the text to your podfile

```
use_frameworks!
pod 'Firebase', '>= 2.5.1'
```

9. Run the pod installation command (wait a few minutes)

```
Pauls-Mac-Pro:Firebase Demo paulsolt$ pod install
Updating local specs repositories

CocoaPods 1.0.0.rc.1 is available.
To update use: `gem install cocoapods --pre`
[!] This is a test version we'd love you to try.

For more information see http://blog.cocoapods.org
and the CHANGELOG for this version http://git.io/BaH8pQ.

Analyzing dependencies
Downloading dependencies
Installing Firebase (2.5.1)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `Firebase Demo.xcworkspace` for this
project from now on.
Sending stats
Pod installation complete! There is 1 dependency from the Podfile and 1 total
pod installed.
```

10. IMPORTANT: Open the .xcworkspace file instead of the .xcodeproj file to edit your app project using Firebase
11. You will see your project and the Pod as a joint project. You can edit the Podfile inside Xcode from now on.

12. In ViewController.swift add the following to the top (lowercase b in Firebase!)

```
import Firebase
```

13. Clean your project (Command + K)
14. Run your project! You’ve just setup Cocoapods and installed your first Cocoapod. You can use this technique for any open source or 3rd party SDK that supports Cocoapods.

Links

- Firebase Quickstart
- Cocoapods

7.4 Important - Podfile Install to Run Sample Code

IMPORTANT: All sample code will not run until you install the podfile!

The podfile downloads a ton of files that take up 74MB, so I won’t be including the Pod directory in the sample code projects. When you download the sample code it will not run unless you use the command to install your podfile to get Firebase setup.

Pod Install Steps

1. Open Terminal (Command + Spacebar) and type: `Terminal` press `Enter`
2. Change directory (cd) into your downloaded sample code folder

```
cd /Users/paulsolt/Downloads/7.4\ -\ Firebase101
```

3. You should see something like (with your user account instead of paulsolt):

```
Pauls-Mac-Pro:7.4 - Firebase101 paulsolt$ cd /Users/paulsolt/Downloads/7.4\ -\ Firebase101
Pauls-Mac-Pro:7.4 - Firebase101 paulsolt$ pwd
/Users/paulsolt/Downloads/7.4 - Firebase101
```

4. Run the pod install command

```
pod install
```

5. Open the Workspace file (.xcworkspace)
6. Clean the project (Command + K) or `Xcode > Product > Clean`
7. Run the app (Command + R)

Terminal Commands

1. `cd` = Change to home Directory (~)
2. `cd /Users/paulsolt/Downloads` = change to paul’s Downloads directory
3. `pwd` = print working directory (current directory)
4. `open .` = open current directory with Finder
5. `ls` = list all files/folders in directory
6. `cd ..` = go up a directory
7. Tip: Use tab to autocomplete as you type with the cd command

7.5 Tutorial - Firebase Basic Commands

1. Create a Firebase connection in ViewController.swift at the top of the class

```
let firebaseRef = Firebase(url: "https://crackling-inferno-8669.firebaseio.com")
var listRef: Firebase!
```

2. In the `viewDidLoad()` method add

```
firebaseRef.setValue("Save your data to the cloud")
let testRef = firebaseRef.childByAppendingPath("test")

// Create and save a new value (JSON data)
let itemTest = ["title": "Eat Lunch", "done": false]
testRef.setValue(itemTest)
sleep(2)

// Update the done status
let itemUpdate = ["done": true]
testRef.updateChildValues(itemUpdate)
sleep(2)

// Remove the item using the reference
testRef.removeValue()
```

3. Explore your Firebase Dashboard as you work with these commands. The sleep() function will let you see delays between actions.

7.6 Tutorial - Create an Item Class for Firebase

1. Firebase works with JSON data, the Item.swift code file will convert both ways.

```
import Foundation
import Firebase
class Item {
    // Optional properties if retrieved from firebase
    var firebaseRef: Firebase?
    var key: String?
    var title: String
    var done: Bool
    var date: NSDate
    // default initializer for local data
    init(title: String, done: Bool) {
        self.title = title
        self.done = done
        self.date = NSDate()
    }
    // default initializer for data being retrieved from firebase
    init(data: FDataSnapshot) {
        self.firebaseRef = data.ref
        self.key = data.key
        self.title = data.value["title"] as! String
        self.done = data.value["done"] as! Bool
        // Store NSDate as a number on firebase
        let timeInterval = data.value["date"] as! NSTimeInterval
        self.date = NSDate(timeIntervalSince1970: timeInterval)
    }
    func toDictionary() -> [String: AnyObject] {
        return [
            "title": title,
            "done": done,
            "date": date.timeIntervalSince1970,
        ]
    }
}
```

2. In ViewController.swift update the configureCell() method to set the textLabel for the current item

```
func configureCell(cell: UITableViewCell, indexPath: NSIndexPath) {
    let item = itemArray[indexPath.row]
    cell.textLabel?.text = item.title
}
```

7.7 Tutorial - Add and Save Items to Firebase

1. Listen for changes to firebase at the bottom of viewDidLoad() in ViewController.swift

```
firebaseRef.observeEventType(.Value) { (snapshot: FDataSnapshot!) in
    print("Data:\n\(snapshot.key) -> \(snapshot.value)")
}
```

2. Add methods to add and save items in ViewController.swfit

```
func addItem(title: String) -> Item {
    let item = Item(title: title, done: false)
    // create a unique firebase url ref
    item.firebaseRef = listRef.childByAutoId()
    // Save task JSON to firebase
    item.firebaseRef?.setValue(item.toDictionary())
    return item
}
func saveItem(item: Item) {
    item.firebaseRef?.updateChildValues(item.toDictionary())
}
```

3. Create a new Firebase reference in viewDidLoad()

```
listRef = firebaseRef.childByAppendingPath("list")
```

4. Add items and save changes to them

```
let item = addItem("Do homework")
// Add more Items
addItem("Wash Car")
addItem("Run!")

// Update an Item's status to true and save to firebase
item.done = true
saveItem(item)
```

7.8 Tutorial - Load TableView Data From Firebase

1. Add logic to convert data from Firebase into local data to display

```
func fetchItems() {
    listRef.queryOrderedByKey().observeEventType(.Value) { (snapshot: FDataSnapshot!) in
        print("Fetch:\n\(snapshot.key) -> \(snapshot.value)")

        var array = [Item]()
        for data in snapshot.children {
            let item = Item(data: data as! FDataSnapshot)
            array.append(item)
        }

        self.itemArray = array
        self.firebaseTableView.reloadData()
    }
}
```

2. Try adding a new Item using JSON on the Dashboard with the app running. You should see an update in the app.

NOTE: If you use the incorrect format, your app could crash from unexpected or missing data.

```
{ "title" : "Cook Dinner", "done": false, "date": 123}
```

7.9 Tutorial - Remove Data From Firebase and Moving Forward

1. In viewDidLoad() remove the logic to replace everything at the top level using `firebaseRef.setValue()` It is preventing our data from persisting between uses.
2. Implement the method removeItem() in ViewController.swift

```
func removeItem(item: Item) {
    item.firebaseRef?.removeValue()
}
```

3. The method is called when you swipe to delete

```
func tableView(tableView: UITableView, canEditRowAtIndexPath indexPath: NSIndexPath) -> Bool
{
    return true
}

func tableView(tableView: UITableView, commitEditingStyle editingStyle:
```

```
UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {
    if editingStyle == .Delete {
        let item = itemArray[indexPath.row]
        removeItem(item)
    }
}
```

4. Now your ready to get started with Firebase and use it in your apps if you need an easy server backend.

Links

- iOS Firebase Guide
- Structuring Data with Firebase