Tyler Kaye

1. Open XCode --> open a new project
2. In terminal navigate to the project directory

```
~ sudo gem install cocoapods
```

3. Open Podfile and add in: pod 'StitchSDK', '~> 4.0.1'

```
'StitchSDK', '~> 4.0.1'
```

```
~ pod install
```

4. Open the .xcworkspace file and do remaining work in here
5. Setup Facebook Authentication

```
https://developers.facebook.com/docs/apps/
```

   A. Setup A New App In Facebook: go to
      i. Log in to your Facebook Developer account. If you do not have a Facebook developer account or even a Facebook account, see Facebook - Register and Configure an App.
      ii. Add a new Facebook application. You will be redirected to the Product Setup page (if not then click Add Product in your newly created app)
      iii. In Product Setup, click Get Started for Facebook Login. The screen should display the Quickstart instructions.
         a. Choose iOS for your platform.
         b. Add pod 'FBSDKLoginKit' to Podfile and run ~ pod install
         c. For the Add your Bundle Identifier step, enter the following and click Save:

```
com.mongo.MongoDBSample
```

         d. Optional. You can enable/disable Single Sign On and continue to the next step.
         e. In your project right-click on Info.plist and open the file as source code
         f. At the bottom of the file (right above </dict> </plist>) paste the two sections of code presented in the facebook setup page
            - Note the Facebook App ID in the top code box in the Facebook Setup page
            - You will use this during the setup step of the ToDo iOS project in this tutorial.

```
<key>FacebookAppID</key>
<string>1111111881399056</string>
```

iv. Click Settings under Facebook Login in the left-hand navigation menu.
a. Under Valid OAuth redirect URIs, add the following entry:

https://stitch.mongodb.com/api/client/v2.0/auth/callback

b. Save changes.
v. Click Settings for your application (i.e. not the Settings under Facebook Login). Note the App ID and App Secret. You will use the information in the Users section of the MongoDB Stitch Admin Console.
B. Configure Stitch for Facebook Authentication.
i. Select Users from the left-side navigation.
ii. Select the Providers tab.
iii. For Facebook, click the Edit button.
iv. In the Facebook provider settings:
v. Switch the Provider Status toggle to enabled.
vi. Enter your new Facebook App ID in the Client ID field and Facebook App Secret in Client Secret field.
vii. Then Click Save.
6. Setup Google Authentication
A. Create the iOS OAuth 2.0 client in Google Cloud Platform.
i. Open the Google Cloud Platform console:

https://console.cloud.google.com/

ii. From the projects list, select an existing project or create a new one.
iii. If the APIs & services page isn't already open, open the left-side menu and select APIs & services.
iv. In the left menu, click Credentials.
v. Click the New Credentials dropdown, and then select OAuth client ID.
vi. You may have to fill in some information about the consent screen such as your application name
vii. Select iOS.
viii. Provide a name for your application
ix. Use the following value for the Bundle ID and then click Create
x. In the list of OAuth 2.0 client IDs, click the client ID that has the name you specified and make note of the iOS URL scheme value; you will add this to your iOS project

com.mongo.MongoDBSample

later in this tutorial.
B. Create the Web App client credentials.
i. From the Google Cloud Platform Console console, click the New Credentials dropdown, and then select OAuth client ID.
ii. Select Web application.
iii. Provide a name for your application.
iv. For Authorized JavaScript origins, enter the following value:

https://stitch.mongodb.com

v. For Authorized redirect URIs, enter the following value:

https://stitch.mongodb.com/api/client/v2.0/auth/callback

vi. Click Create.
vii. Note the client ID and secret. You will use these values in the Users section of the MongoDB Stitch Console in the next section of this tutorial.

C. Configure Stitch Admin Console for Google Authentication
i. Select Users from the left-side navigation.
ii. Select the Providers tab.
iii. For Google, click the Edit button.
iv. In the Google provider settings
v. Switch the Provider Status toggle to enabled.
vi. Enter your new Client ID and Client Secret from the Google Web App client credentials.
vii. Click Save.

D. Configure Application for Google Authentication: (think there may be an easier way)
i. Google Sign-in requires a custom URL Scheme to be added to your project. To add the custom scheme:
ii. Open your project configuration: double-click the project name in the left tree view. Select your app from the TARGETS section, then select the Info tab, and expand the URL Types section.
iii. Click the + button, and add your reversed client ID as a URL scheme.
iv. The reversed client ID is your client ID with the order of the dot-delimited fields reversed. For example: com.googleusercontent.apps.ABCDEFGH……

com.googleusercontent.apps.ABCDEFGH……

v. Open Podile and add: pod ‘GoogleSignIn'
vi. From the terminal: ~ pod install

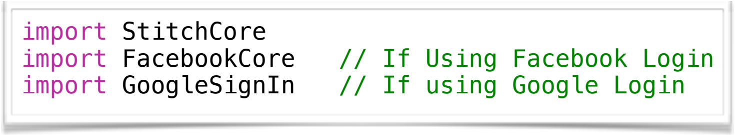
7. Writing the Actual Application:

A. Verify your Podfile has a section that looks like this

```
# Pods for TodoStitch
 pod 'StitchSDK', '~> 4.0.1'
 pod ‘FacebookLogin’  # If using Facebook Auth
 pod ‘FacebookCore'   # If using Facebook Auth
 pod ‘GoogleSignIn’   # If using Google Auth
```

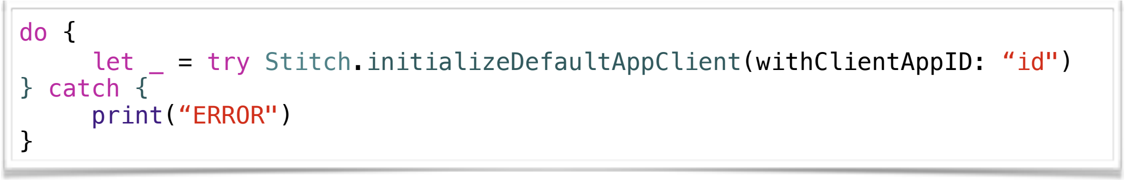
i. If it does not, then edit it and run ~ pod install

B. Open the AppDelegate.swift file and make sure we are importing Stitch by inserting



```
import StitchCore
import FacebookCore   // If Using Facebook Login
import GoogleSignIn   // If using Google Login
```

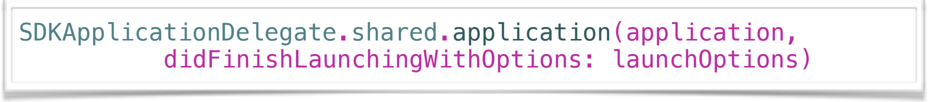
C. First thing to do is to initialize a default app client. While we can connect to multiple Stitch applications, setting a default app client sets a global variable that enables us to call Stitch.defaultAppClient from anywhere within our project and get the default Stitch client. Thus, in AppDelegate, edit the didFinishLaunchingWithOptions() function to:



```
do {
     let _ = try Stitch.initializeDefaultAppClient(withClientAppID: “id")
} catch {
     print(“ERROR")
}
```

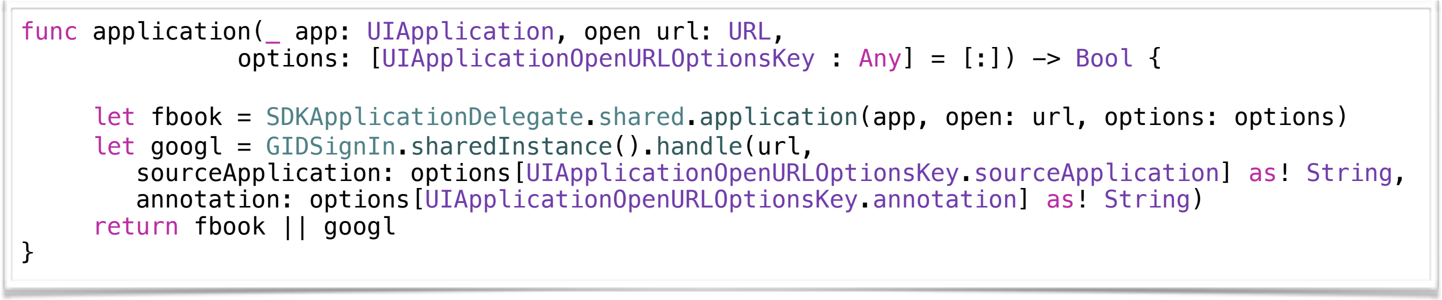
D. Next if using Facebook / Google Authentication you deed to update two of the methods in App Delegate:

i. In didFinishLaunchingWithOptions, we need to let the Facebook SDK know, so insert the following line (either above or below the Stitch default client initialization)



```
SDKApplicationDelegate.shared.application(application,
          didFinishLaunchingWithOptions: launchOptions)
```

ii. Insert a new function in app delegate. If you are using both Facebook and Google Authentication then copy the code below into AppDelegate.swift. If using one or the other, then just keep the relevant initialization and return only that variable.



```
func application(_ app: UIApplication, open url: URL,
               options: [UIApplicationOpenURLOptionsKey : Any] = [:]) -> Bool {

     let fbook = SDKApplicationDelegate.shared.application(app, open: url, options: options)
     let googl = GIDSignIn.sharedInstance().handle(url,
        sourceApplication: options[UIApplicationOpenURLOptionsKey.sourceApplication] as! String,
        annotation: options[UIApplicationOpenURLOptionsKey.annotation] as! String)
     return fbook || googl
}
```

E. Login View Controller Code:

A. TodoTableViewController