

PREREQUISITES

To set the iOS app up, we need to download and install few framework that are used to generate the Transit iOS application.

you need a Mac computer (macOS 10.11.5 or later) running the latest version of Xcode.

Download the latest version of Xcode on your Mac free from the App Store.

To download the latest version of Xcode

1. Open the App Store app on your Mac (by default it's in the Dock).
2. In the search field in the top-right corner, type Xcode and press the Return key.
3. The Xcode app shows up as the first search result.
4. Click Get and then click Install App.
5. Enter your Apple ID and password when prompted.
6. Xcode is downloaded into your /Applications directory.

INSTALLATION

Transit User Application:

1. Extract the iOS zip file.
2. Open the Transit User App in Xcode and wait till the app launches successfully.
3. On the project explorer window of Xcode, find and edit the following files
 - a. In General ->Identity -> Bundle Identifier
 - i. applicationId "com.appoets.transit.user" -> change this to your app's Bundle ID
 - ii. Display name -> name of your app
 - b. In the navigation area search for constant.h file (usually that may in services group)
 - i. #define SERVICE_URL = "<http://demo.transit.co>"; -> Change this to your app's base URL.
 - ii. #define ClientID = 2; -> Change this to your app's client_id.
 - iii. #define Client_SECRET =
"yVnKCIKDHPcDIqqO1V05RtDRdvtrVHfvjlfasfdaa"; -> Change this to your app's client_secret.
 - iv. #define Stripe_KEY= "pk_test_0G4SKYM246m8dK6kgayCPwKWtXy"; -> Change this to your app's Strip token.

To login google developer console you have to login with your gmail credentials

- v. `#define GOOGLE_API_KEY = @"Some key"` -> change this to your app's google api key that you get from the google developer console
 - vi. `#define GMSMAP_KEY = @"Some key"` -> change this to your app's google maps key that you get from the google maps console
 - vii. `#define GMSPLACE_KEY= @"Some key"` -> change this to your app's google places key that you get from the google maps console
 - viii. `#define Google_Client_ID= @"Some key"` -> change this to your app's google key that you get from the google maps console
-
- 4. Change the splash screen and icon for your own brand.
 - 5. The theme color of the app should be change in the colors.h in the css group folder inside the app -> `#define BLACKCOLOR RGB(18, 18, 18)` (give the RGB value of your theme color)

Open the project folder from the your mac and follow the below mentioned path:

User-> user->Assets.xcassets

you must save the icons and splash screens in these folders according to the device size.

Now hit on the 'run' button the application will run successfully in the Simulator.

If you want to run the build in your device(iPhone) you must enroll in the apple developer account to register your device

Tranxit Driver Application:

Extract the iOS zip file.

Open the Tranxit User App in Xcode and wait till the app launches successfully.

On the project explorer window of Xcode, find and edit the following files

- a. In General ->Identity -> Bundle Identifier
 - i. applicationId "com.appoets.tranxit.provider" -> change this to your app's Bundle ID
 - ii. Display name -> name of your app
- b. In the navigation area search for config.h file (usually that may in services group)
 - i. `#define SERVICE_URL = "http://demo.tranxit.co";` -> Change this to your app's base URL.

To login google developer console you have to login with your gmail credentials

- ii. `#define GOOGLE_API_KEY = @"Some key"` -> change this to your app's google api key that you get from the google developer console
- iii. `#define GSMAP_KEY = @"Some key"` -> change this to your app's google maps key that you get from the google maps console
- iv. `#define GMSPLACE_KEY= @"Some key"` -> change this to your app's google places key that you get from the google maps console

Change the splash screen and icon for your own brand.

The theme color of the app should be change in the colors.h in the css group folder inside the app ->`#define BLACKCOLOR RGB(18, 18, 18)` (give the RGB value of your theme color)

Open the project folder from the your mac and follow the below mentioned path:

User-> user->Assets.xcassets

you must save the icons and splash screens in these folders according to the device size.

Now hit on the 'run' button the application will run successfully in the Simulator.

If you want to run the build in your device(iPhone) you must enroll in the apple developer account to register your device

To run your app in the simulator

1. In the Scheme pop-up menu in the Xcode toolbar, choose iPhone 7.
2. The Scheme pop-up menu lets you choose which simulator or device you'd like to run your app on. Make sure you select the iPhone 7 Simulator, not an iOS device.
3. Click the Run button, located in the top-left corner of the Xcode toolbar.
4. Watch the Xcode toolbar as the build process completes.
5. Xcode displays messages about the build process in the [activity viewer](#), which is in the middle of the toolbar.

After Xcode finishes building your project, the simulator starts automatically. It may take a few moments to start up the first time.

The simulator opens in the iPhone mode you specified and then launches your app. Initially, the simulator displays your app's launch screen, and then it transitions to your app's main interface.

The launch image of your app can be change in the LaunchScreen.storyboard or add the different size of launch images in the Assets.xcassets -> launchscreens

You can install iOS App files on devices using Xcode.

To install an app on a device using Xcode

1. Connect the device to your Mac.
2. In Xcode, choose Window > Devices and select the device under Devices.
3. In the Installed Apps table, click the Add button (+) below the table.
4. In the dialog that appears, choose the iOS App file and click Open.

For more tasks you can perform in the Devices window, read [Managing Apps on Devices](#).

To install the app on an iOS device using iTunes

1. Connect the testing device to a Mac running iTunes.
2. If possible, don't use a Mac that you use for development. For watchOS apps, connect an iPhone paired with an Apple Watch.
3. Double-click the iOS App file that you created earlier.
4. In iTunes, click the device in the upper-left corner of the window.
5. Click the Apps button.
6. The app appears in the iTunes Apps list.
7. Under Apps, choose "Sort by Name" or "Sort by Kind" from the pop-up menu.
8. An Install or Remove button appears adjacent to the app.
9. If an Install button appears, click it.
10. The button text changes to Will Install.

11. Click the Apply button or the Sync button in the lower-right corner to sync the device.
12. The app is uploaded to the device so that the user can start testing.

If you are planning to upload the app in the Apple Store, you must generate the archive build. To do so,

Archiving Your App

Next, create an archive of your app. Xcode stores this archive in the Archives organizer.

To create an archive

1. In the project navigator, choose a target from the Scheme toolbar menu, and click Run. Always test the app before you export it.
2. Choose Product > Archive. The Archives organizer appears and displays the new archive.

To export a development certificate signed app using the team provisioning profile

1. In the Archives organizer, select the archive and click Export.
2. Select “Export a Development-signed Application” and click Next.
3. In the dialog that appears, choose a team from the pop-up menu and click Choose.
4. If necessary, Xcode creates the needed signing identity and provisioning profile for you.
5. In the dialog that appears, review the app, its entitlements, and the provisioning profile, and click Export.
6. The team provisioning profile should begin with the text Mac Team Provisioning Profile:
7. In the dialog that appears, enter a filename and choose a location, and click Export.