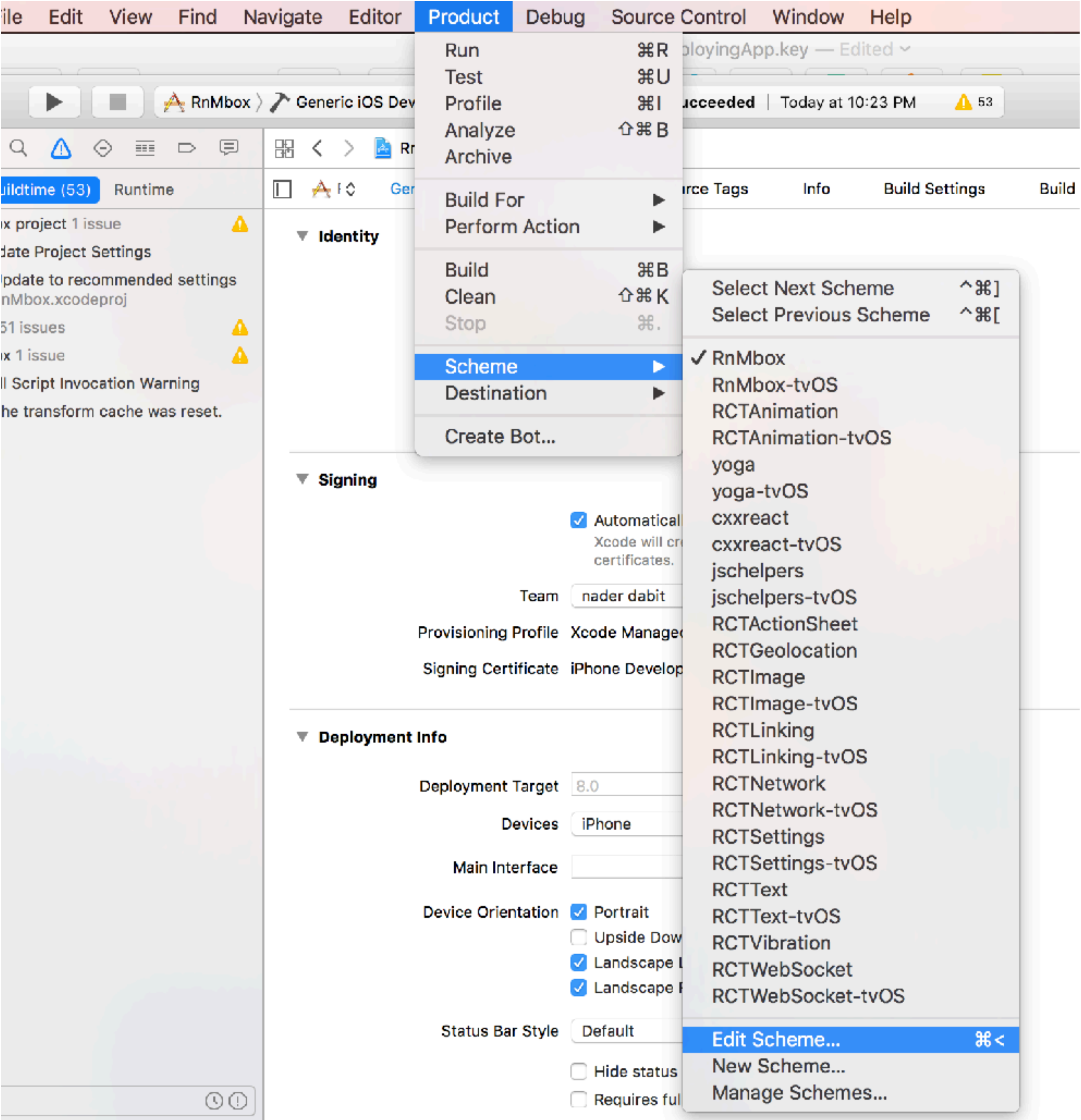


Deploying App

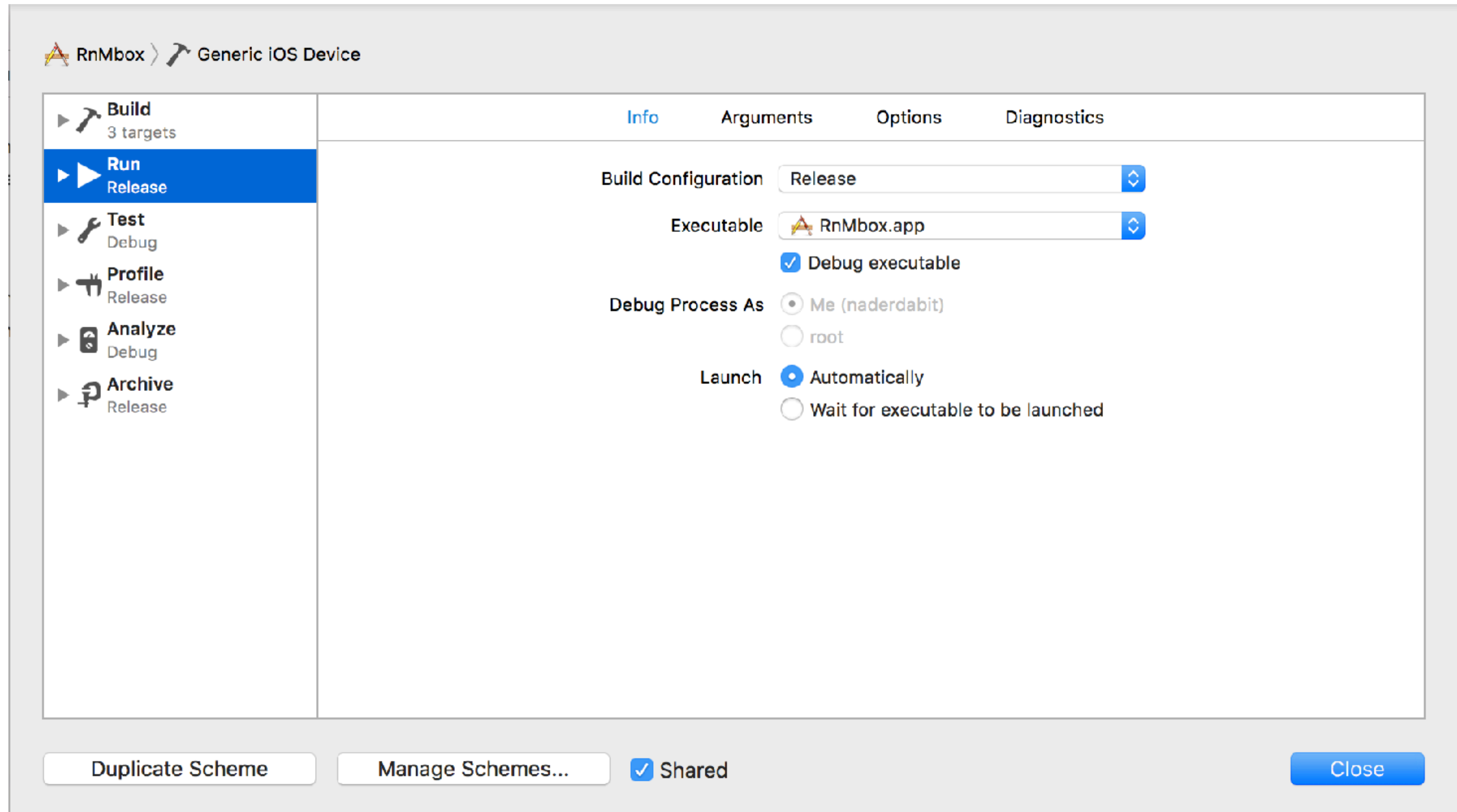
Deploying App - iOS

Step 1: change scheme to release (vs debug)

Deploying App - iOS



Deploying App - iOS

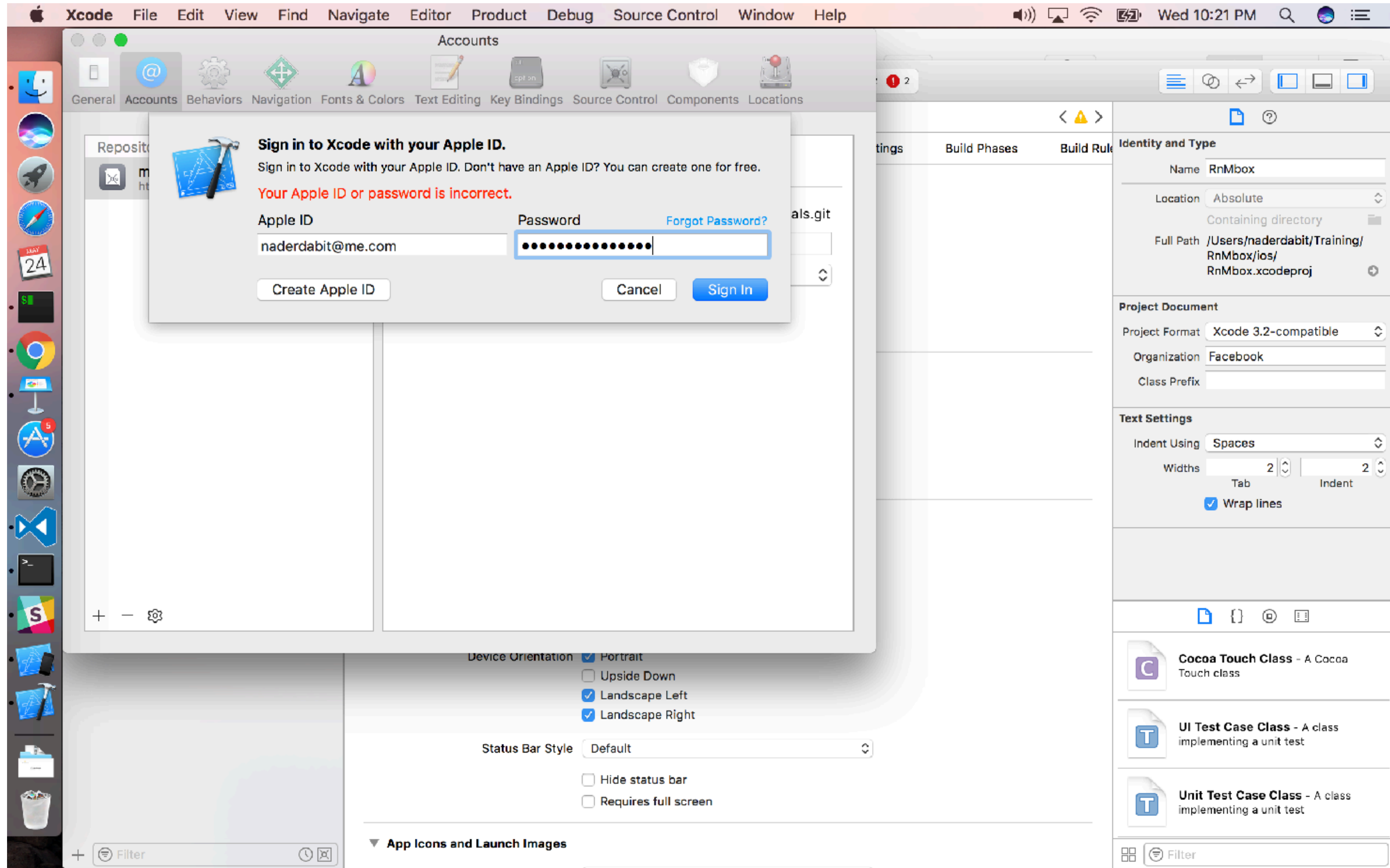


Deploying App - iOS

Step 2: sign in with apple id that is registered as an apple developer

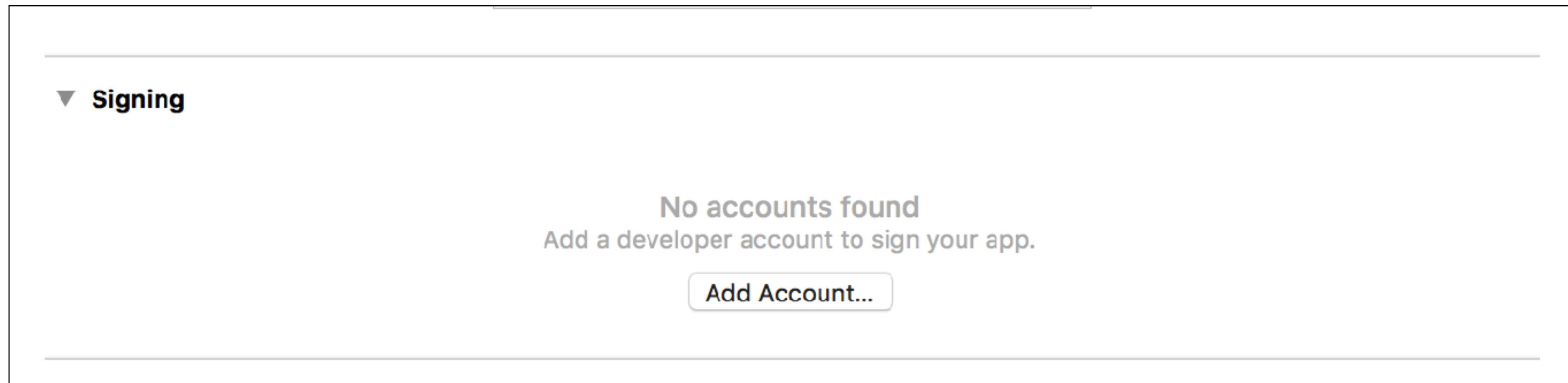
Deploying App - iOS

Step 2: sign in with apple id that is registered as an apple developer

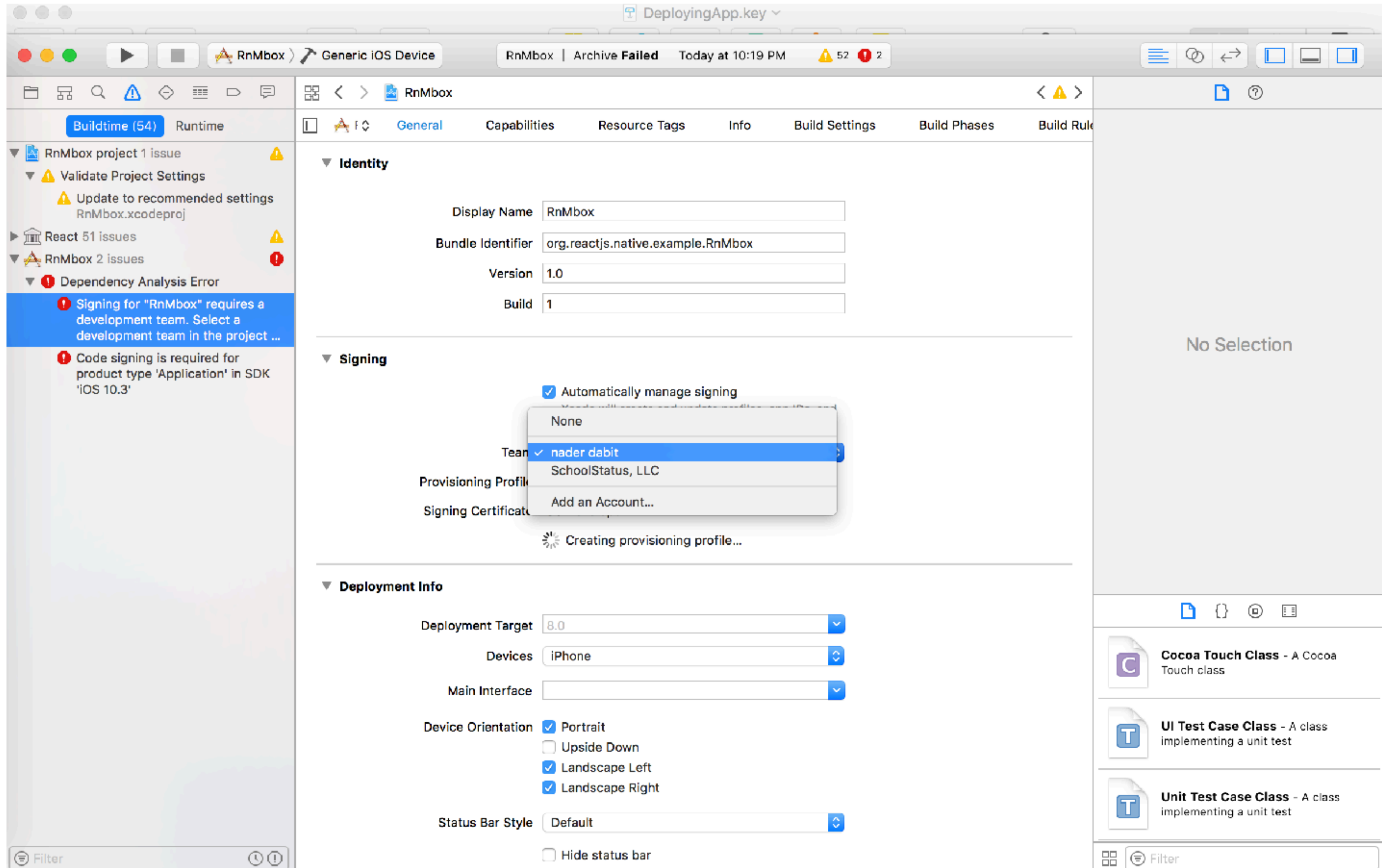


Deploying App - iOS

Step 3: Make sure you have a signing identity chosen



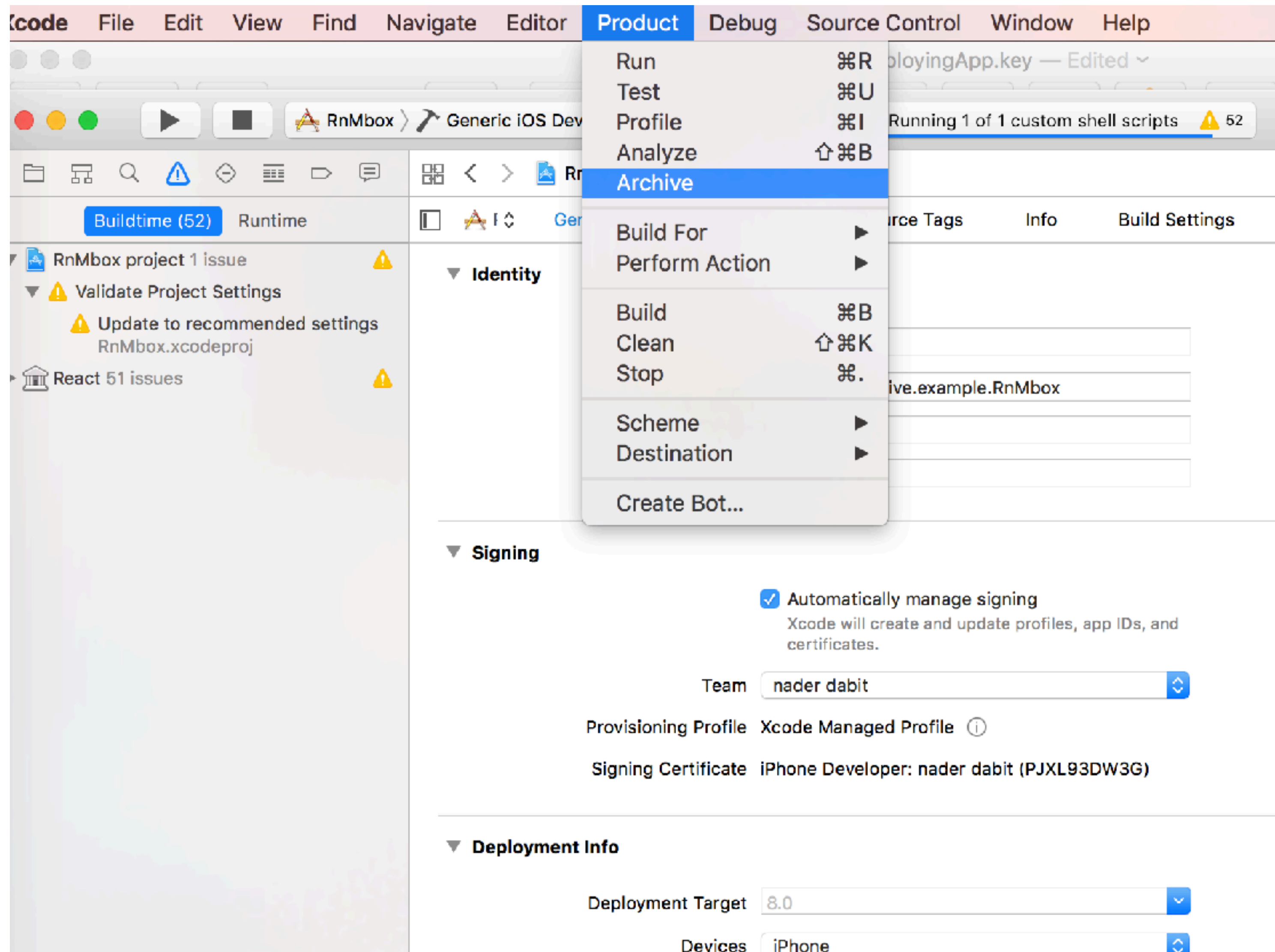
Deploying App - iOS - signing identity



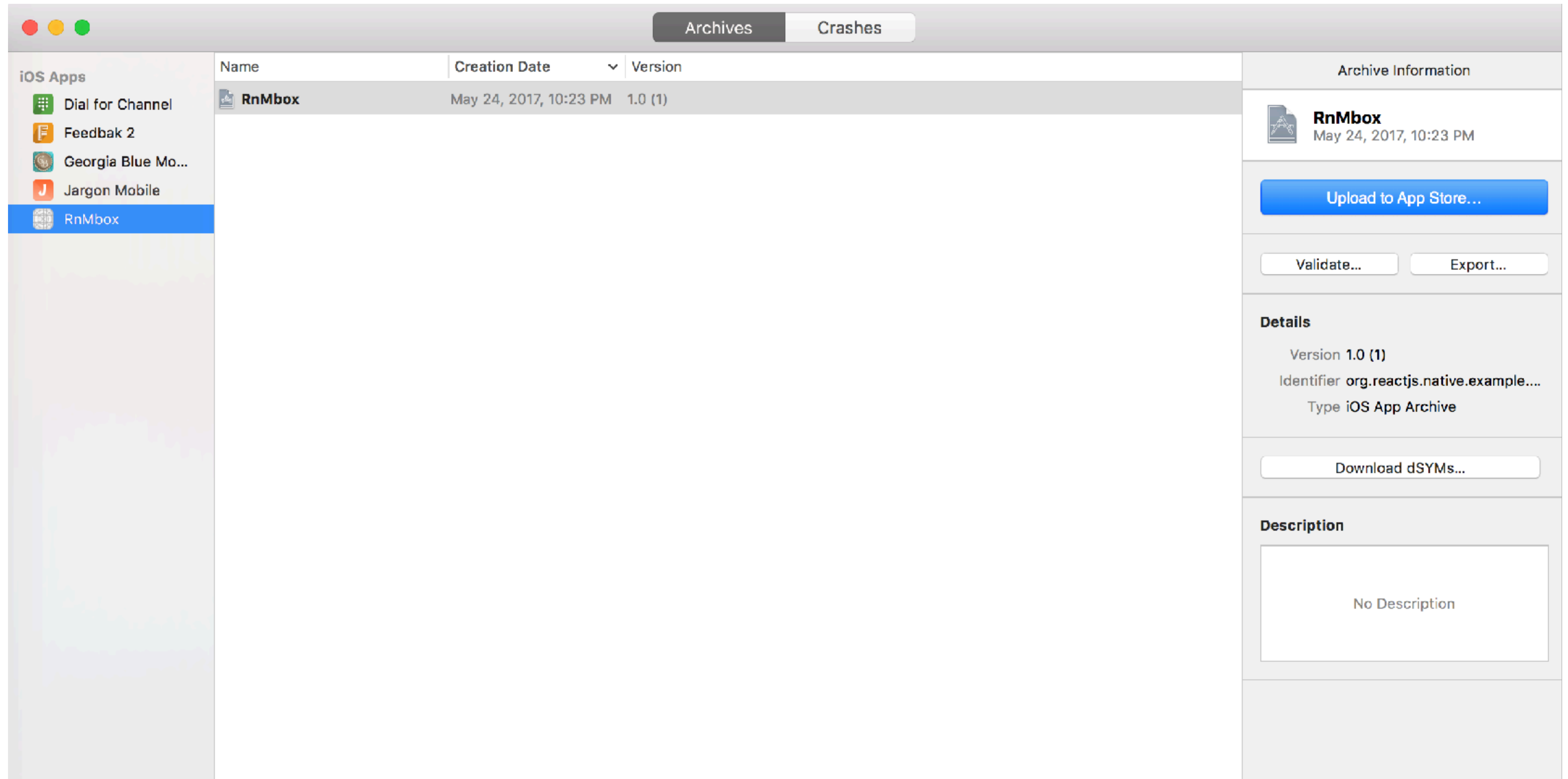
Deploying App - iOS

Step 4: Create .ipa file for app using archive
(make sure to choose generic device)

Deploying App - iOS Choose Product -> Archive from menu




Deploying App - iOS access to generated .ipa file



The screenshot displays the Xcode 'Archives' window. On the left, a sidebar lists 'iOS Apps' with icons and names: 'Dial for Channel', 'Feedbak 2', 'Georgia Blue Mo...', 'Jargon Mobile', and 'RnMbox' (which is selected and highlighted in blue). The main area features a table with columns 'Name', 'Creation Date', and 'Version'. A single entry is shown: 'RnMbox' created on 'May 24, 2017, 10:23 PM' with version '1.0 (1)'. On the right, a detailed view for the selected archive is shown. It includes the app icon, name 'RnMbox', and creation date. Below this is a prominent blue button labeled 'Upload to App Store...'. Further down are 'Validate...' and 'Export...' buttons. A 'Details' section lists 'Version 1.0 (1)', 'Identifier org.reactjs.native.example....', and 'Type iOS App Archive'. At the bottom of this section is a 'Download dSYMs...' button. The final section, 'Description', contains a text box with the placeholder 'No Description'.

Name	Creation Date	Version
RnMbox	May 24, 2017, 10:23 PM	1.0 (1)

Archive Information

 **RnMbox**
May 24, 2017, 10:23 PM

[Upload to App Store...](#)

[Validate...](#) [Export...](#)

Details

Version 1.0 (1)
Identifier org.reactjs.native.example....
Type iOS App Archive

[Download dSYMs...](#)

Description

No Description

Deploying App - Android

Step 1: We will generate a private signing key using keytool.

Deploying App - Android

```
keytool -genkey -v -keystore some-key-name.keystore -alias some-key-alias -  
keyalg RSA -keysize 2048 -validity 10000
```

keep track of the alias you pass here

Deploying App - Android - creating keystore

```
cd
fundamentals-slides(master) x: cd
~: keytool -genkey -v -keystore some-key-name.keystore -alias some-key-alias -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: nader dabit
What is the name of your organizational unit?
[Unknown]: something
What is the name of your organization?
[Unknown]: somethingnew
What is the name of your City or Locality?
[Unknown]: jackson
What is the name of your State or Province?
[Unknown]: ms
What is the two-letter country code for this unit?
[Unknown]: us
Is CN=nader dabit, OU=something, O=somethingnew, L=jackson, ST=ms, C=us correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=nader dabit, OU=something, O=somethingnew, L=jackson, ST=ms, C=us
Enter key password for <some-key-alias>
(RETURN if same as keystore password):
[Storing some-key-name.keystore]
~: █
```


Deploying App - Android

Step 2: Place the my-release-key.keystore file under the android/app directory in your project folder.

Deploying App - Android

Step 3: Edit the file `~/.gradle/gradle.properties` and add the following (replace `*****` with the correct keystore password, alias and key password),

```
MYAPP_RELEASE_STORE_FILE=some-key-name.keystore
```

```
MYAPP_RELEASE_KEY_ALIAS=some-key-alias
```

```
MYAPP_RELEASE_STORE_PASSWORD=*****
```

```
MYAPP_RELEASE_KEY_PASSWORD=*****
```

These are going to be global gradle variables

Deploying App - Android

Step 4: open android/app/build.gradle in your project folder and add the signing config,

```
signingConfigs {
    release {
        if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
            storeFile file(MYAPP_RELEASE_STORE_FILE)
            storePassword MYAPP_RELEASE_STORE_PASSWORD
            keyAlias MYAPP_RELEASE_KEY_ALIAS
            keyPassword MYAPP_RELEASE_KEY_PASSWORD
        }
    }
}

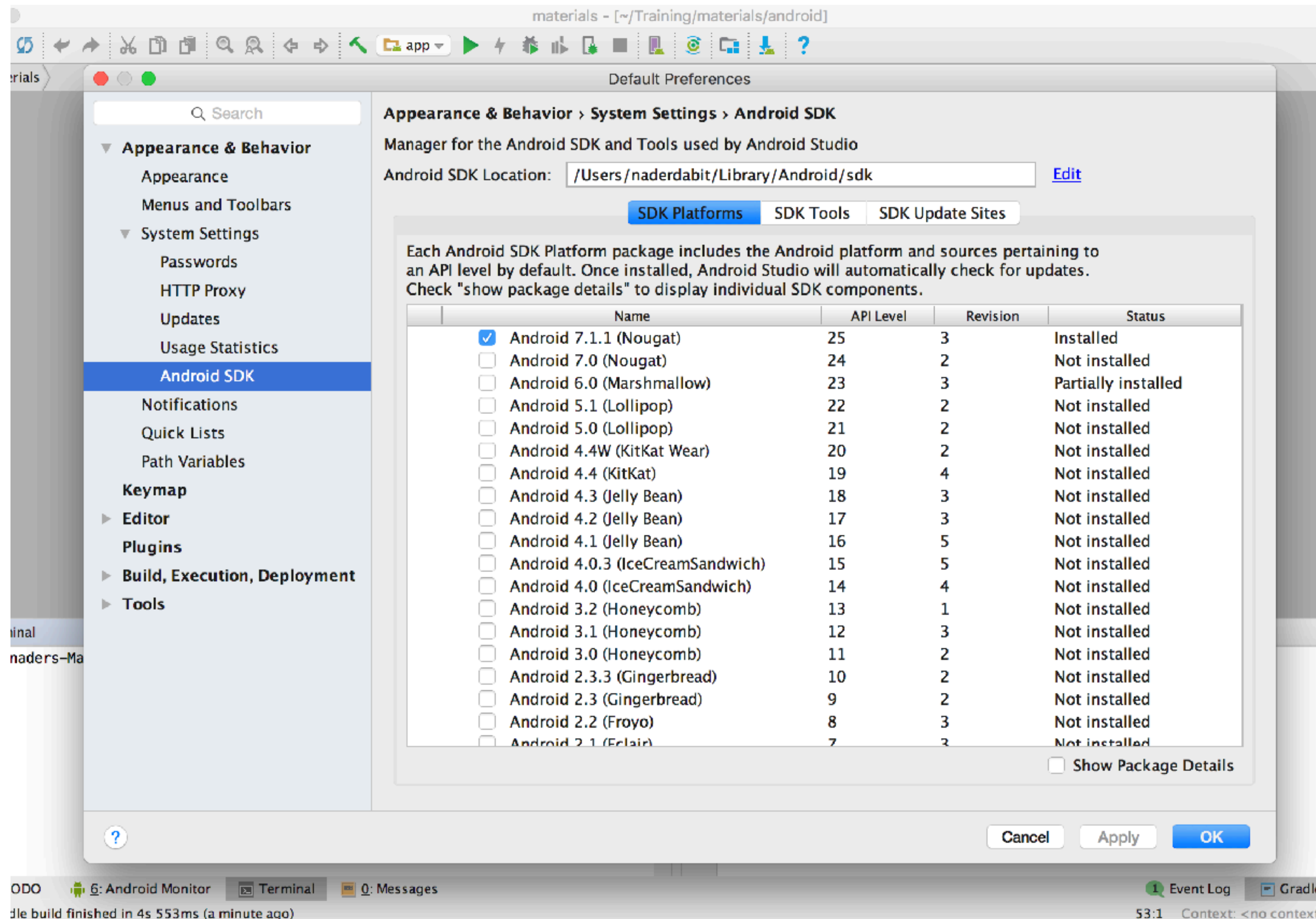
buildTypes {
    release {
        ...
        signingConfig signingConfigs.release
    }
}
```

Deploying App - Android

Step 5: Set SDK Variable in project

open tools/android/sdkmanager to get path to sdk

Deploying App - Android



Deploying App - Android

Step 6: Create `local.properties` file in android directory with the sdk directory

```
sdk.dir=/Users/naderdabit/Library/Android/sdk
```


Deploying App - Android

Generating the release APK

```
cd android && ./gradlew assembleRelease
```

Deploying App - Android

The generated APK can be found under `android/app/build/outputs/apk/app-release.apk`

Deploying App - Android

Testing the release build.

```
react-native run-android --variant=release
```