

[Open in app](#)[Get started](#)

Published in Nerd For Tech



Amr Essam

[Follow](#)May 30, 2021 · 10 min read · [Listen](#)[Save](#)

# How to install your Expo React Native application on iOS without using Xcode or macOS

Ever wondered how can you install or deploy Expo react native application natively on iOS without having to use Xcode or a macOS? If you just wondered or tried but didn't have the luck to find out how, guess what? This tutorial is for you.

In this article, We'll go through step by step until you open your Expo Managed Workflow application on your iOS device by generating and downloading a ".ipa" file on an iOS device.

As mentioned before, this article is best suited for users with Expo Managed Workflow, for more information about the difference between the managed and bare workflows you can refer to this [Expo article](#).

The steps are performed on Ubuntu, but it won't change if you're using another Linux distribution or even Windows.

We'll be going through 11 steps:

- In the first 8 steps, we configure your Apple developer account and create files needed to build your application.
- In Step 9, we modify your Expo project files.
- In Step 10 we build your application using the Expo command



[Open in app](#)[Get started](#)

how to do so or just use the official one, unfortunately, you can't skip this step, there is no other way to download your application on your ios device without having an apple developer account.



62



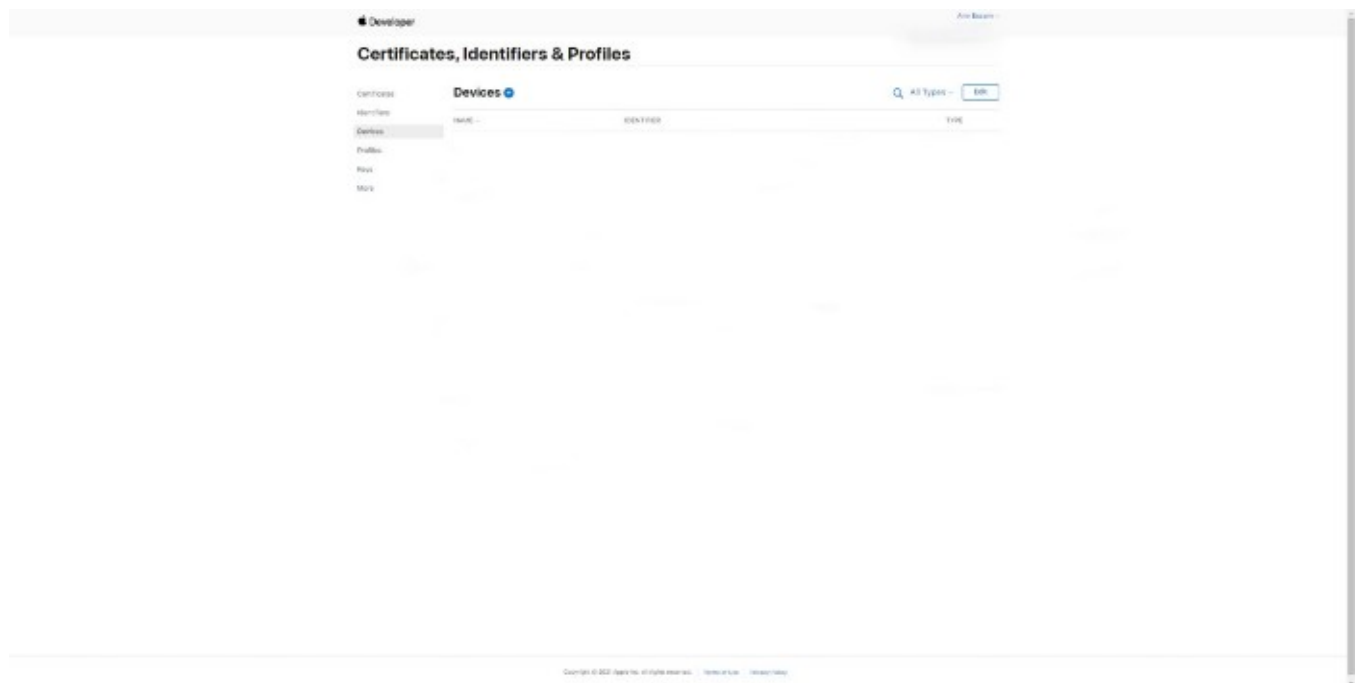
4

## Step 2, Register your Testing Devices

As part of Apple's security for preventing developers from distributing their applications in unverified stores, Apple requires that you register the devices that you're planning to use for testing.

In this step, we'll be registering the devices required in testing.

In your developer account navigate to the "Devices" tab.



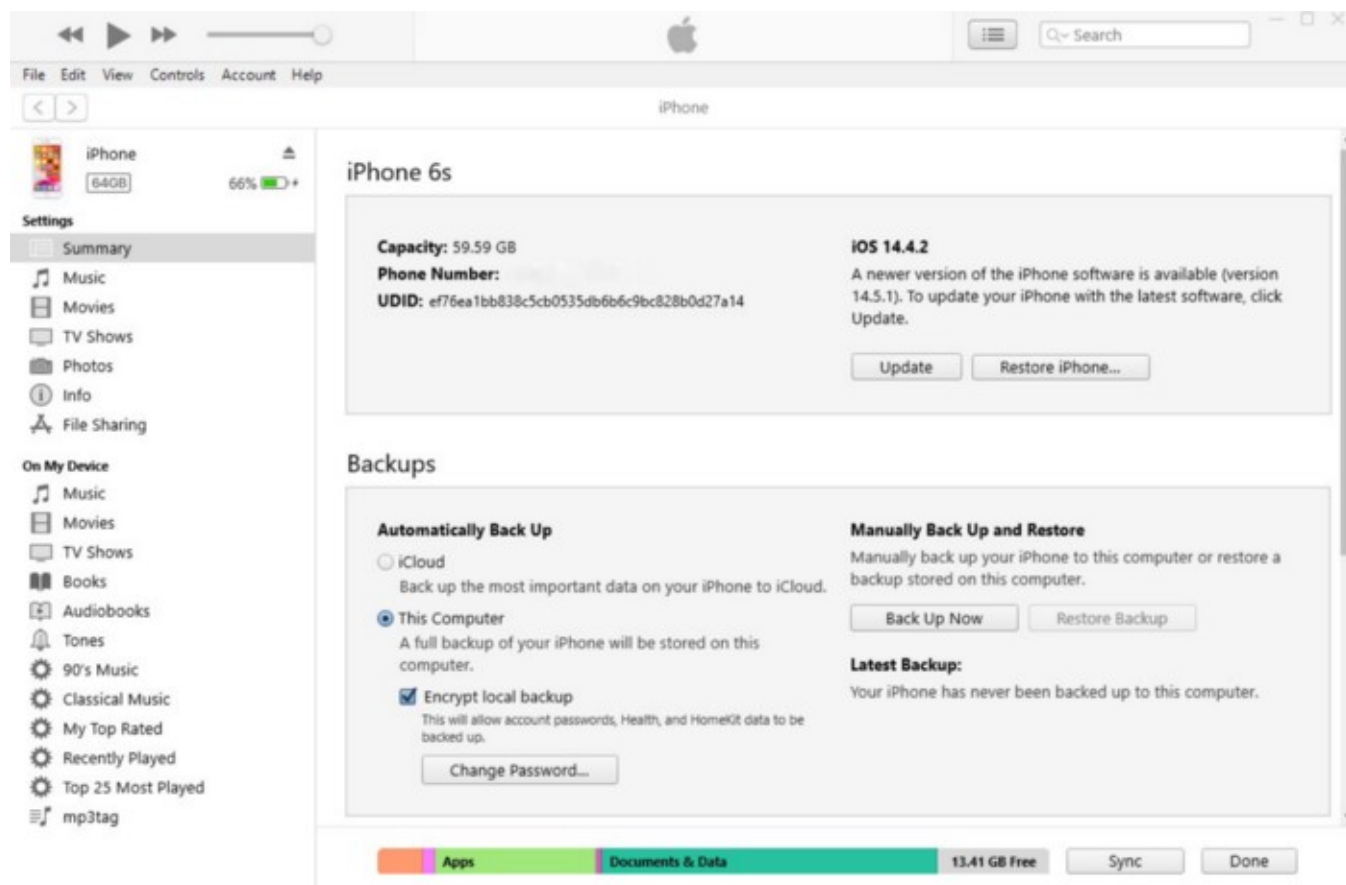
Devices tab on Apple developer account

- 1- After you navigate to the Devices tab, click on the "+" icon to add a new device.
- 2- Choose your platform, which is "iOS, tvOS, watchOS" if you're intending to test your application on your iOS device.
- 3- Write your device name, you can find it easily on your iOS device by going to Settings > General > About.




[Open in app](#)
[Get started](#)

switched to “UDID” and the text beside it is your “UDID”.



UDID in iTunes

5- Finally click on continue and you have finished step 2 successfully.

Before we leave this step, you can register multiple devices by uploading a file with the information you’ve just added manually, you can find a sample file that Apple supplies after you click on the “+” button.

### Step 3, Create an Identifier

In this step, we’ll be creating a unique identifier for your application, so the first thing you need to do is to navigate to the “Identifiers” tab.

1- After you navigate to the “Identifiers” tab, click on the “+” icon to add a new identifier.

2- Choose “App IDs” and then click continue.





Identifiers registration page on the Apple developer account

3- Select App and click “Continue”.



## Identifiers type page on the Apple developer account

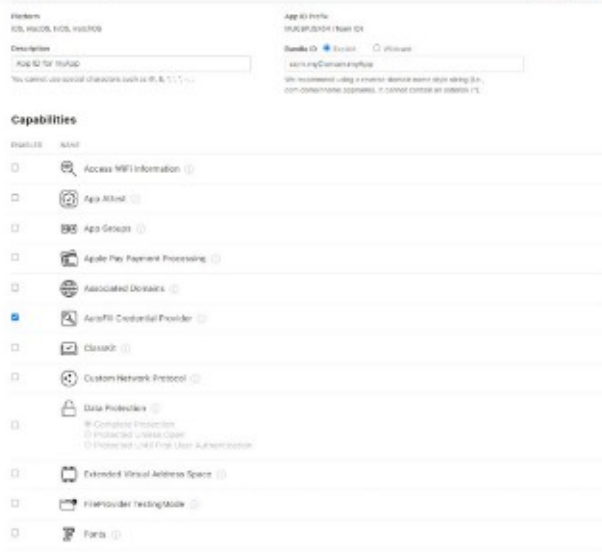
4- Write a description for your application identifier, for example, if your app name is “myApp”, you can use “App ID for myApp”.

5- Write your bundle ID, you'll find that Apple recommends using a reverse-domain name style string, for example, if your domain name is "myDomain.com" and your application name is "myApp" then the recommended name is



Open in app

Get started



Identifiers capabilities page on the Apple developer account

6- Choose the capabilities required in your application, for example, you can choose “AutoFill Credential Provider” and “Push Notifications”.

## Step 4, Create a CSR (Certificate Signing Request)

In this step, we'll be generating a CSR which will be used in step 5 to create a certificate in the Apple developer account.

So before we create a CSR, CSR is the process of requesting the certificate from the Certificate Authority (Apple), so that Apple can verify the details of who is requesting an issue of the developer certificate. To know more about this topic, you can refer to [this](#) article.

1- Open a terminal, and write this command to generate a key:

```
openssl genrsa -out mykey.key 2048
```

2- After generating a key, we'll be using it to generate a certificate, so assuming my email is “amr.essam95@hotmail.com” and my name is “Amr” and I'm living in Egypt, then the command will be:





Open in app

Get started

file because as you can see we're using it in the second command.

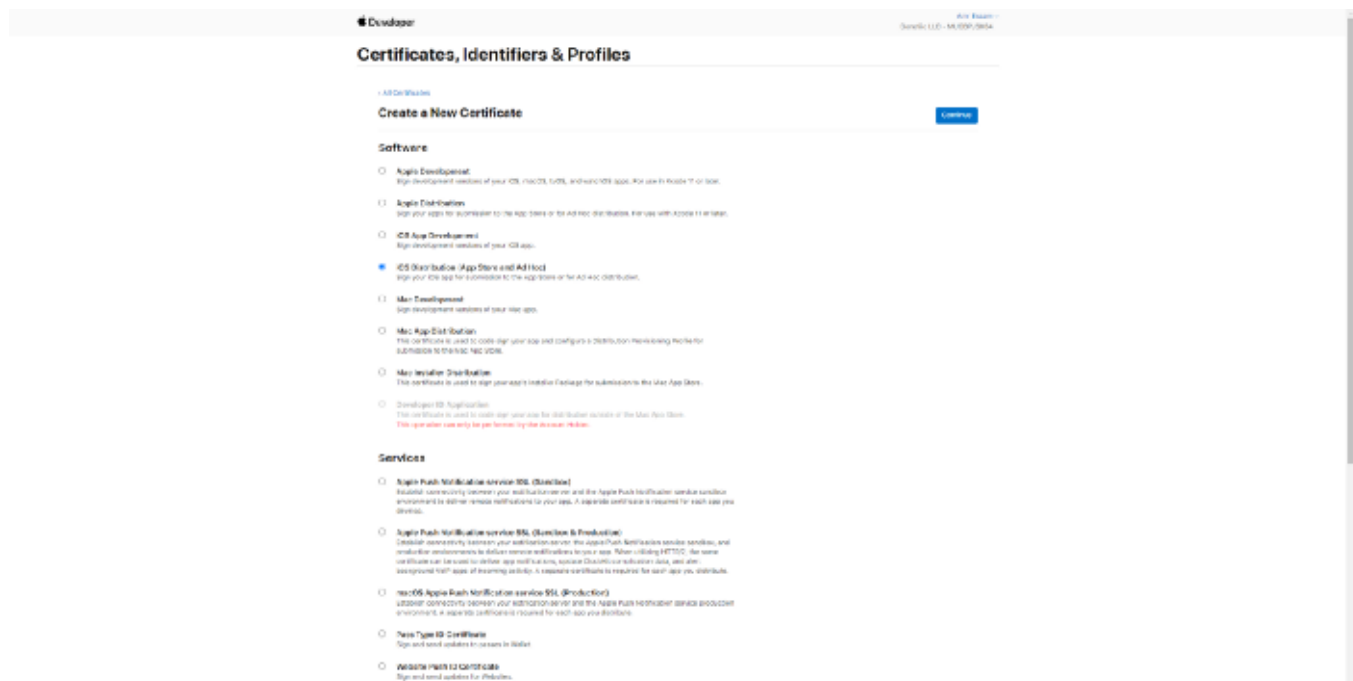
After running the command, a new file "ios\_distribution.csr" is generated.

## Step 5, Create a Certificate

In this step, we'll be creating a distribution certificate for you (the developer), so this certificate can be used for all your applications, for more details about the certificate you can refer to [this](#) Expo article about App Signing, it's a very good article.

1- After you navigate to the "Certificates" [tab](#), click on the "+" icon to add a new certificate.

2- In the "Software" part, choose "iOS Distribution (App Store and Ad Hoc)", which will allow you to test your application on your iOS device, you can skip the "Services" part if your use model doesn't require it.



New certificate page on Apple developer account

3- After you click "Continue", you'll be asked to upload your CSR file that we've just generated in the previous step.

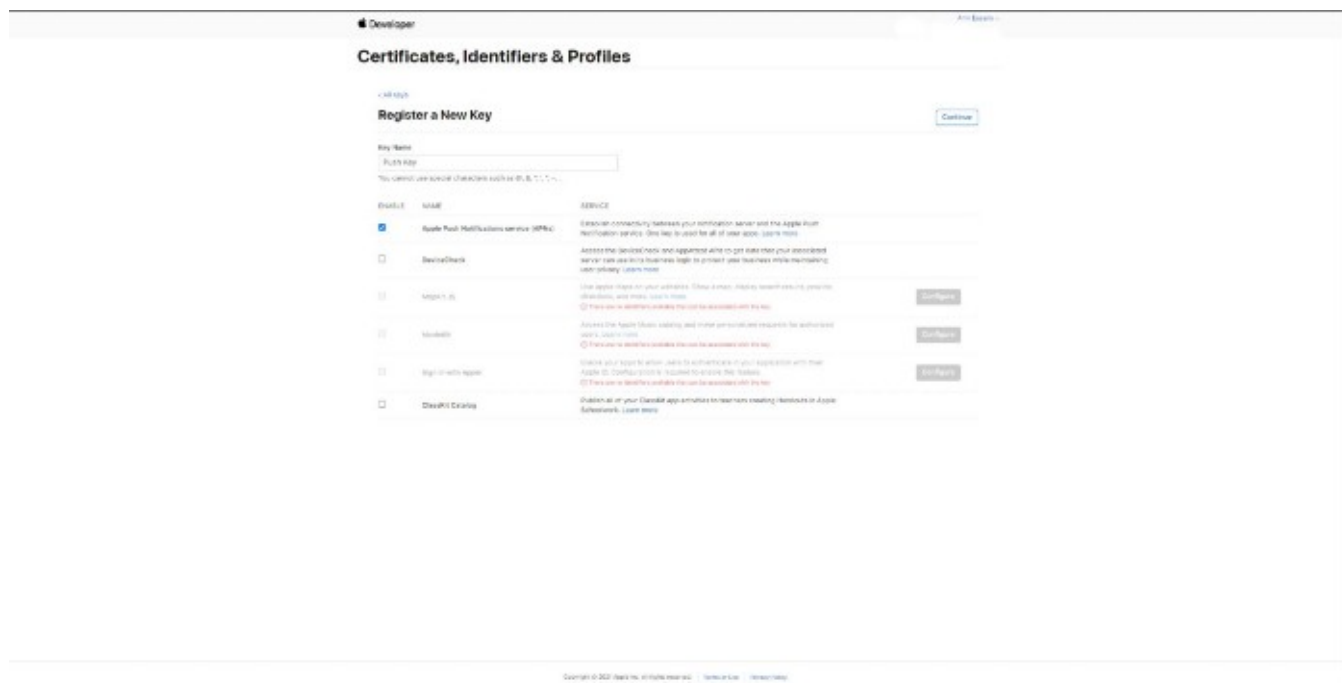
4- After you click "Continue", you've now successfully generated a certificate, you need




[Open in app](#)
[Get started](#)

step.

- 1- After you navigate to the “Keys” tab, click on the “+” icon to add a new Key.
- 2- Write the key Name, which can be anything for example you can name it “Push Key”.
- 3- Choose “Apple Push Notifications Service (APNs)” and click continue.



Push notifications registration page on Apple developer account

- 4- Download your generated key and save it as it'll be used in building your application (Step 10).

## Step 7, Create a Provisioning Profile

In this step, we'll be creating a provisioning profile, which is app-specific, you will have a provisioning profile for every app you submit to the App Store. Provisioning profiles expire after 12 months, but this won't affect apps in production. You will just need to create a new one the next time you build your app.

- 1- After you navigate to the “Profiles” tab, click on the “+” icon to add a new Profile.
- 2- Choose “Ad Hoc” as we're planning to distribute this app on an iOS device and click

“Continue”




[Open in app](#)
[Get started](#)

- Development**
- ☐ **iOS App Development**  
Create a provisioning profile to install development apps on test devices.
  - ☐ **iOS App Development**  
Create a provisioning profile to install development apps on test devices.
  - ☐ **macOS App Development**  
Create a provisioning profile to install development apps on test devices.
- Distribution**
- ☒ **Ad Hoc**  
Create a distribution provisioning profile to install your app on a limited number of registered devices.
  - ☐ **iOS Ad Hoc**  
Create a distribution provisioning profile to install your app on a limited number of registered iOS devices.
  - ☐ **App Store**  
Create a distribution provisioning profile to submit your app to the App Store.
  - ☐ **iOS App Store**  
Create a distribution provisioning profile to submit your iOS app to the App Store.
  - ☐ **Mac App Store**  
Create a distribution provisioning profile to submit your app to the Mac App Store.
  - ☐ **Developer ID**  
Create a Developer ID provisioning profile to use Apple services with your Developer ID signed applications.

Copyright © 2022 Apple Inc. All rights reserved. [Terms of Use](#) [Privacy Policy](#)

### Provisioning profile registration page on Apple developer account

3- You'll be asked for the identifier we've created in step 3, choose the one we've created before and click on "Continue".

4- You'll be asked for the certificate associated with the profile, choose the one we've created in step 5 and click on "Continue".

5- You'll be asked for the devices you wish to include, choose the devices we've created in step 2.

6- Finally you'll be asked for a name for this profile, you can choose any name you want, for example, you can use "Release Testing Profile".

7- Click on "Continue" and download your profile because we'll use it later.

## Step 8, Create a p12 Signed Certificate

In this step, we'll be creating a ".p12" file which is a file for holding the private keys.

1- Navigate to the directory where you've saved the "ios\_distribution.cer" file you've downloaded in the 6th step and the key "mykey.key" you've generated in the 4th step.

2- Run the following two commands, the second command will ask you for a password so make sure you remember it as we'll be using it in one of the next steps. In the second command, we'll be generating "ios\_distribution.p12" file.





[Open in app](#)[Get started](#)

```
openssl pkcs12 -inkey mykey.key -in ios_distribution.pem -export -  
out ios_distribution.p12 -name "Amr Essam"
```

## Step 9, Modify your app.json file

One last step before starting to generate your “.ipa” file is modifying your “app.json” file with the “bundle ID” we’ve created before in the 3rd step.

1- Open your app.json file, you should find an “ios” object in the “expo” object like this, the “...” before and after “ios” refers to content we’re not going to cover in this step.

```
{  
  "expo": {  
    ...  
    "ios": {  
      "supportsTablet": true,  
      "icon": "./assets/icon.png",  
      "bundleIdentifier": "com.myDomain.myApp"  
    },  
    ...  
  }  
}
```

2- Write the bundle ID we’ve used in the 3rd step, in my case here it was “com.myDomain.myApp”.

Congratulations we’ve finished all the steps required before starting to generate our “.ipa” file which we’ll download on our iOS app to test our application.

## Step 10, Generate a “.ipa” file

In this step, we’ll be using Expo build command to generate a “.ipa” file.

1- Open your terminal and navigate to your Expo project, and run this command:

```
expo build:ios
```



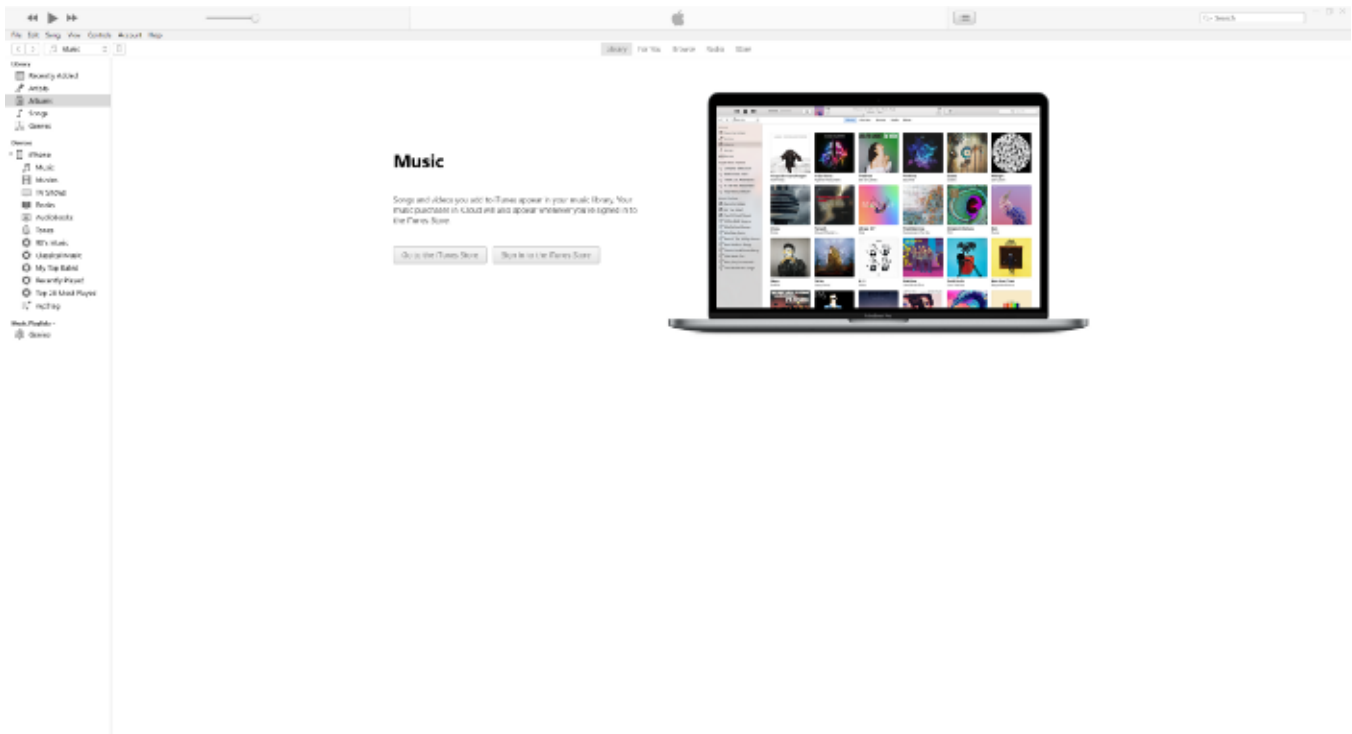
[Open in app](#)[Get started](#)

- 3- You'll be asked if you have an access to an Apple developer account, we're going to write "y" which is yes.
- 4- You'll be asked for the email and password of your Apple developer account.
- 5- After submitting your email and password, you'll be asked if you'll provide your own "Apple Distribution Certificate", we're going to choose "I want to upload my own file".
- 6- You'll be asked for the "P12" file path, we're going to write the path of the "P12" file we've generated in the 8th step, and then you'll be asked for the password we've chosen in the 8th step.
- 7- After entering your password, you'll see some commands loading like "Getting Distribution Certificate from Apple..." and then you'll be asked for the "Apple Push Notifications service key", we're going to choose "I want to upload my own file" and then write the path of the file we've downloaded in the 6th step which had the extension "p8".
- 8- After entering the path of the "p8" file, you'll be asked for the "Key ID", you can get your key ID by going to the "Keys" tab in the Apple developer account and click on the key we've created in the 6th step and we'll find the "Key ID" and then we're going to enter our "Key ID".
- 9- After entering your "Key ID", you'll be asked for the "Apple Provisioning Profile" which is our last step, we're going to choose "I want to upload my own", and then we'll write the path for the provisioning profile we've downloaded in the 7th step which had the extension ".mobileprovision".
- 10- After entering the path for your ".mobileprovision" file, you'll find that you're now building your Javascript bundle, and then your application will be built on Expo servers and you'll find a link to track your application building.
- 11- Open the link for monitoring the build and wait for your build to finish and then finally download your ".ipa" file on your computer.



[Open in app](#)[Get started](#)

1- Open iTunes and connect your iOS device, and then click on the “Device” icon under the “Account” tab.



iTunes

2- After clicking on the “Device” icon, your device will open, drag your “.ipa” file in the “On My Device” section until it’s in blue color, and then leave your mouse. After doing so, you’ll find that iTunes is now downloading the app on your device.

After iTunes finishes downloading, Congratulations, you’ll find that the app is now on your iOS device and you can open it like any other application downloaded from the App Store.

## Final Notes

- If you’re going to build your application again after the first build, the good news, you’ll find that all the certificates, keys, and provisioning profiles are saved and you won’t be going through these steps again, but if you want to start over again and clear all saved files, you can run the build command using the following command instead of the “expo build:ios” command.



[Open in app](#)[Get started](#)

- If you haven't configured your application icons, you can read [this](#) tutorial on how to easily configure your application icons using Expo.

Finally, if you had any problems with these steps, Feel free to contact me on my [LinkedIn account](#).

## References

[1] DZone, iOS Code Signing: Part 2, Certificate Signing Requests, [https://dzone.com/articles/ios-code-signing-part-2-certificate-signing-reques#:~:text=The%20Certificate%20Signing%20Request%20\(CSR\)%20is%20the%20process%20of%20requesting,from%20a%20local%20macOS%20machine.](https://dzone.com/articles/ios-code-signing-part-2-certificate-signing-reques#:~:text=The%20Certificate%20Signing%20Request%20(CSR)%20is%20the%20process%20of%20requesting,from%20a%20local%20macOS%20machine.)

[2] A.Hoefling, How To Develop iOS Without a Mac, <https://www.andrewhoefling.com/Blog/Post/how-to-develop-ios-without-a-mac>

[3] Expo, App Signing, <https://docs.expo.io/distribution/app-signing/>

---

## Sign up for NFT Weekly Digest

By Nerd For Tech

Subscribe to our weekly News Letter to receive top stories from the Industry Professionals around the world [Take a look](#).

Your email



Get this newsletter



[Open in app](#)[Get started](#)[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

