*“IonicWoocommerce” Documentation by “VectorCoder” v2.4*

“IonicWoocommerce”

**Created: 09/March/2018   
By: VectorCoder   
Email:**[**vectorcoder@hotmail.com**](mailto:vectorcoder@hotmail.com)

Thank you for purchasing. If you have any questions that are beyond the scope of this help file, please feel free contact support team through our website [support panel](http://support.vectorcoder.com/help-center). Thanks so much!

Table of Contents

[**1) Installation – Wordpress Plugins** -  2](#_Toc508386805)

[**2) Installation – Ionic Mobile App** -  9](#_Toc508386806)

**3)**[**Generate a Google Server API Key** 10](#_Toc508386807)

[a. Create a Firebase project 11](#_Toc508386808)

[b. Getting your Firebase Cloud Messaging token and Sender ID 12](#_Toc508386809)

[c. Configure your OneSignal app's Android platform settings 13](#_Toc508386810)

**4)**[**Generate an iOS Push Certificate** 15](#_Toc508386811)

[**5) Install Tools and Deploy app 24**](#_Toc508386819)

[a. Build Android App for Testing On Windows 26](#_Toc508386820)

[b. Setting up you Mac for testing app and Deploy App 31](#_Toc508386821)

[**Build iOS app for Testing** 31](#_Toc508386822)

**6) Upload to play store**

**7)Upload to app store**

[**8) FAQ’S** -  41](#_Toc508386833)

**1) Installation – Wordpress Plugins** -

Please follow following steps to setup wordpress environment.

**1.1) Wordpress Requirements** -

* WooCommerce 2.6+.
* WordPress 4.4+.
* Pretty permalinks in Settings > Permalinks so that the custom endpoints are supported. **Default permalinks will not work.**
* You may access the API over either HTTP or HTTPS, but *HTTPS is recommended where possible*.

If you use ModSecurity and see 501 Method Not Implemented errors, see [this issue](https://github.com/woocommerce/woocommerce/issues/9838) for details.

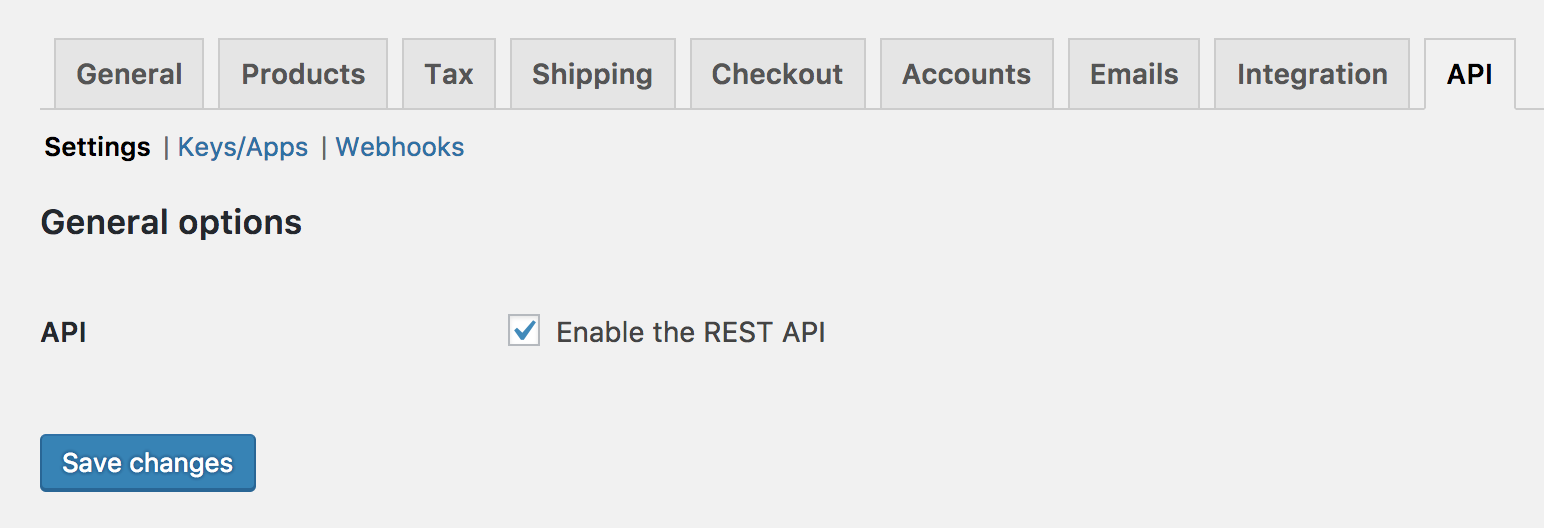
**1.1.1) Import Woocommerce Sample Data** -

If you don't have setup your woocommerce store, then setup it first. You can download woocommerce sample data from this link.

https://docs.woocommerce.com/document/importing-woocommerce-sample-data/

https://github.com/woocommerce/woocommerce/tree/master/sample-data

**1.2) Enable REST API & Generate API keys**

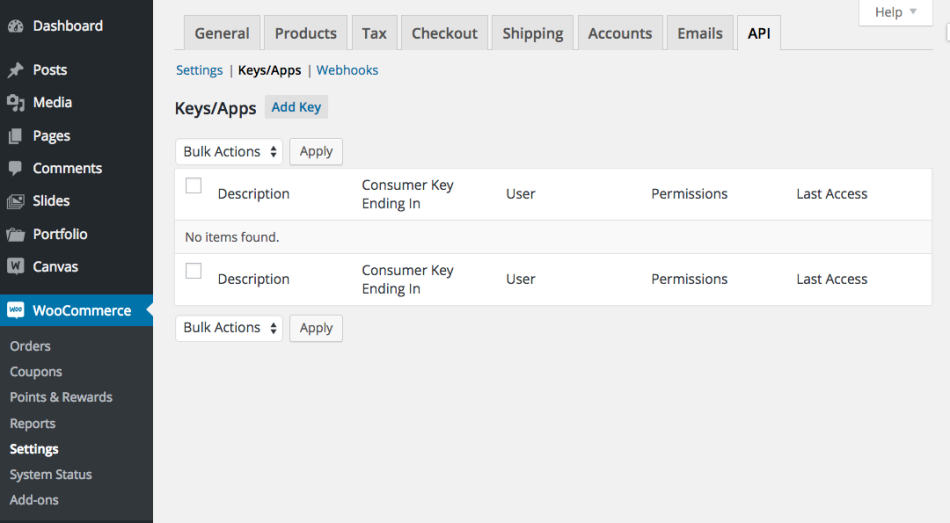
[](https://docs.woocommerce.com/wp-content/uploads/2015/07/woo-api-settings.png)

To enable the REST API within WooCommerce, go to **WooCommerce > Settings > API** and tick the **Enable REST API** checkbox.

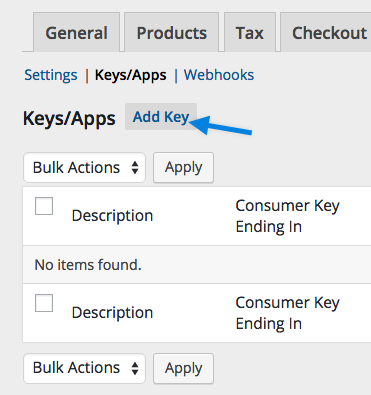
The WooCommerce REST API works on a key system to control access. These keys are linked to WordPress users on your website.

To create or manage keys for a specific WordPress user:

1/ Go to: **WooCommerce > Settings > API > Keys/Apps**.

[](http://docs.woocommerce.com/wp-content/uploads/2014/02/Screen-Shot-on-2015-09-09-at-17-36-06.png)

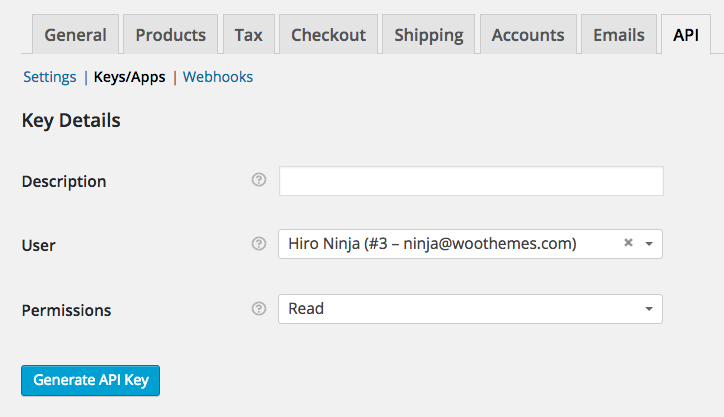
2/ Select**Add Key**

[](http://docs.woocommerce.com/wp-content/uploads/2014/02/Screen-Shot-2015-09-09-at-17.38.02.png)

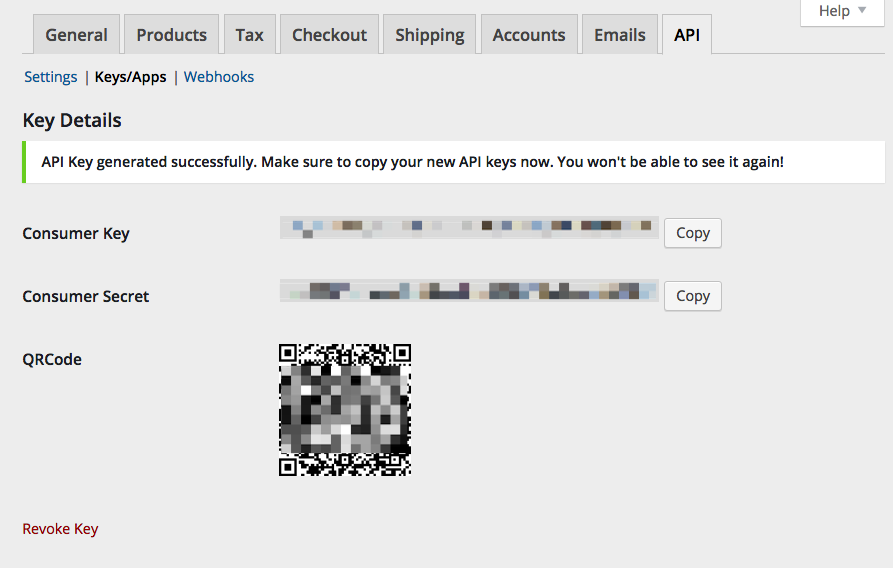
3/ Select the User you would like to generate a key for in the User field and add a Description.

4/ Choose the level of access for this API key, which can be **Read** access, **Write** access or **Read/Write** access.

5/ Select the **Generate API Key** button, and WooCommerce will generate API keys for that user.

[](http://docs.woocommerce.com/wp-content/uploads/2014/02/Generate-Key.png)

Now that keys have been generated, you should see two new keys, a QRCode, and a Revoke API Key button. These two keys are your **Consumer Key** and **Consumer Secret**.

[](http://docs.woocommerce.com/wp-content/uploads/2014/02/API-keys.png)

6/ Place your site link, consumer key and consumer secret in the mobile application config file that uses the WooCommerce API.

**1.3) Setup Plugins**

* Upload the json-api folder to the /wp-content/plugins/ directory or install directly through the plugin installer. Activate the plugin through the ‘Plugins’ menu in WordPress or by using the link provided by the plugin installer.
* Upload the ionic-ecommerce plugin to /wp-content/plugins/ directory and activate the plugin through `plugins` menu in WordPress.
* After that activate the controllers "AppUsers" and "AppSettings" through the /settings/json API/

http://yoursite/wp-admin/options-general.php?page=json-api

**App Settings**

After **wordpress**Login go to the **IonicEcommerce > App Setting**section and change the settings according to your requirements.

http://yoursite/wp-admin/admin.php?page=ionic-ecommerce

* Select App Home Page Style
* Select Category Page Style
* Display Footer Menu
* Display Add to Cart Button on Home Page
* iOS app share link
* App Name
* Your Website
* Currency Symbol
* New Product Tag Duration
* Contact/Inquiry Email
* Your Store Information (Phone Number, Address, City, State, Zip, Country, Latitude, Longitude)
* Side Menu Management
* Local Notification title, text and duration
* OneSignal Notifications Credentials
* Facebook Login Credentials
* AdMob Credential for your Android & iOS
* Manage Content Pages (About us, refund, privacy policy and Terms of usage**)**

**Banner Settings**

Go to the **IonicEcommerce > Banner Settings**section and add banners for your app home screen.

http://yoursite/wp-admin/admin.php?page=banner-settings

**Language Settings**

Go to the **IonicEcommerce > Language Settings**section and select language for app or add new one.

http://yoursite/wp-admin/admin.php?page=language-settings

**Lables Settings**

Go to the **IonicEcommerce > Lables Settings**section and manage all app labels.

http://yoursite/wp-admin/admin.php?page=labels-settings

**2) Installation – Ionic Mobile App**

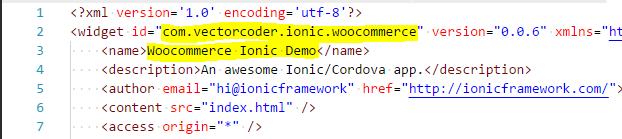
After wordpress plugins installation and setup, please follow below instructions to set up your mobile application.

**2.1) Ionic Source Code Setup**

Now open the folder name ( **App Source Code**) inside the downloaded package.

Inside the folder you will find all the ionic application code.

Open your project folder and open file “ **config.xml**“. This file has all the configuration that is used to built your **apk** and **ipa** file.

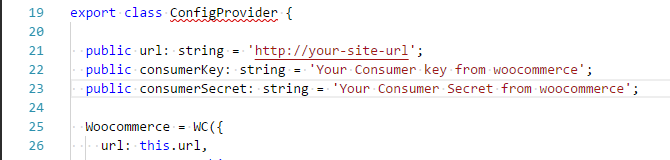


The id inside widget tag contains **package name** of the application that must be unique for every application in world (**com.vectorcoder.ionic.woocommerce**) and the other one is your **application name** that will be shown when application is installed (Woocommerce **Ionic demo**).

Now open **project folder/src/providers/config.ts** file and change the application

* [**http://your-site-url**](http://your-site-url)**(**It will be the url of your domain**)**
* **your consumer key and**
* **consumer secret**

.. It will load CMS products data into your application.



Now we are ready to go further.

**2.2) Cordova Plugins Setup**

Before moving further you have to add some plugins with your credentials.

**Facebook Login Plugin**  
Add Plugin Using below code Using your app id and app name

Add Plugin Using below code Using your app id and app name ionic cordova plugin add cordova-plugin-facebook4 –variable APP\_ID=**“yourAppId”**–variable APP\_NAME=**“yourAppName”**

**You have to enable it From CMS.**

For more detail please visit <https://ionicframework.com/docs/native/facebook/>

**2.3) Config OneSignal for Android**

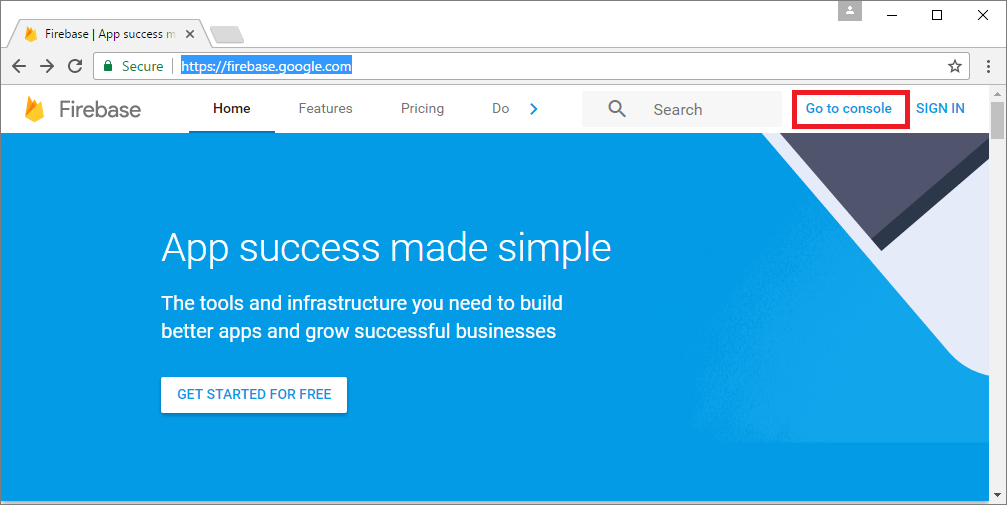
# Generate a Google Server API Key

Your **Google Server API Key** and **Google Project Number** are used to send push notifications to **ANDROID**devices.

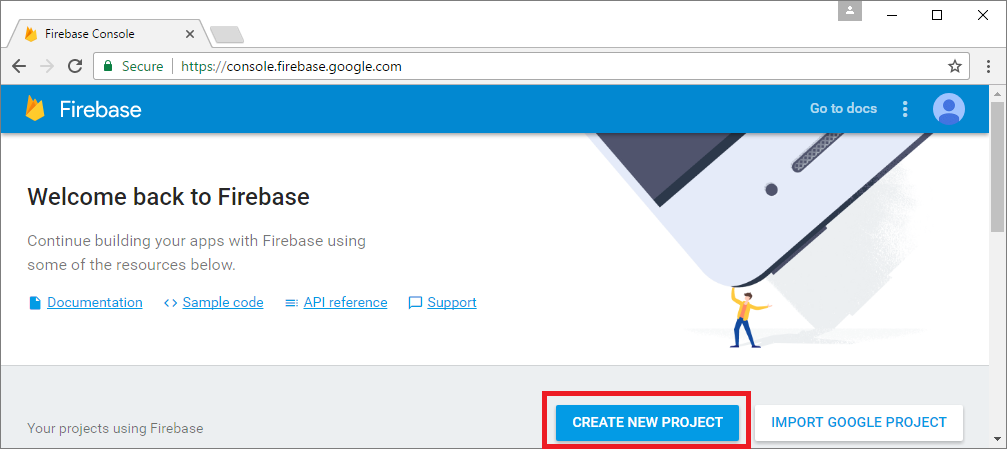
To begin, we'll obtain a Google Server API Key and Google Project Number. These keys allow OneSignal to use Google's web push services for your notifications.

## 1.Create a Firebase project

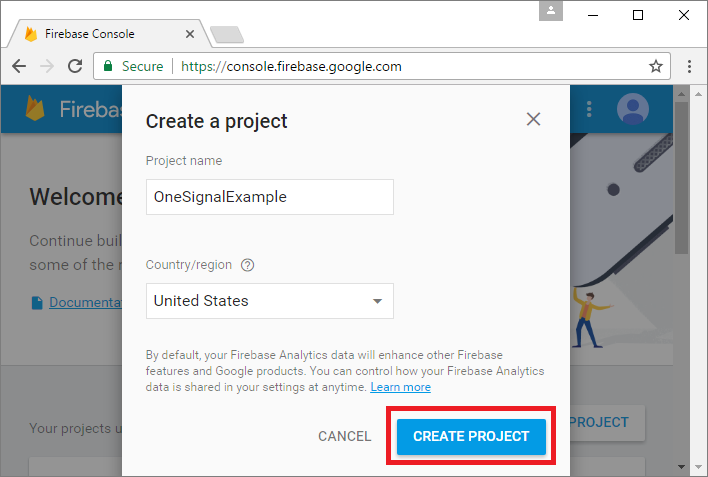
**1.1** Visit the [Firebase Console](https://firebase.google.com/) and sign in with your Google account.

[](https://files.readme.io/1a62b77-FireBase-1.1-go-to-console.png)

**1.2** Press "CREATE NEW PROJECT" or select an existing one below.

[](https://files.readme.io/ef03430-FireBase-1.2-create-new-project.png)

**1.3** Enter a project name and press "CREATE PROJECT".

[](https://files.readme.io/3fdb533-FireBase-1.3-create-project.png)

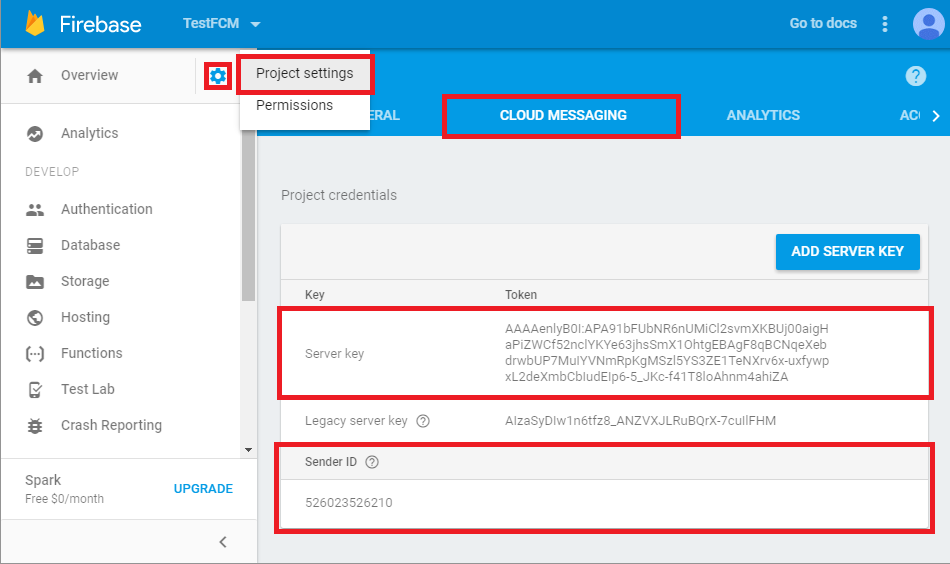
## 2. Getting your Firebase Cloud Messaging token and Sender ID

**2.1** Click the Gear icon in the top left and select "Project settings".

**2.2** Select the "CLOUD MESSAGING" tab.

**2.3** Save the two values listed:

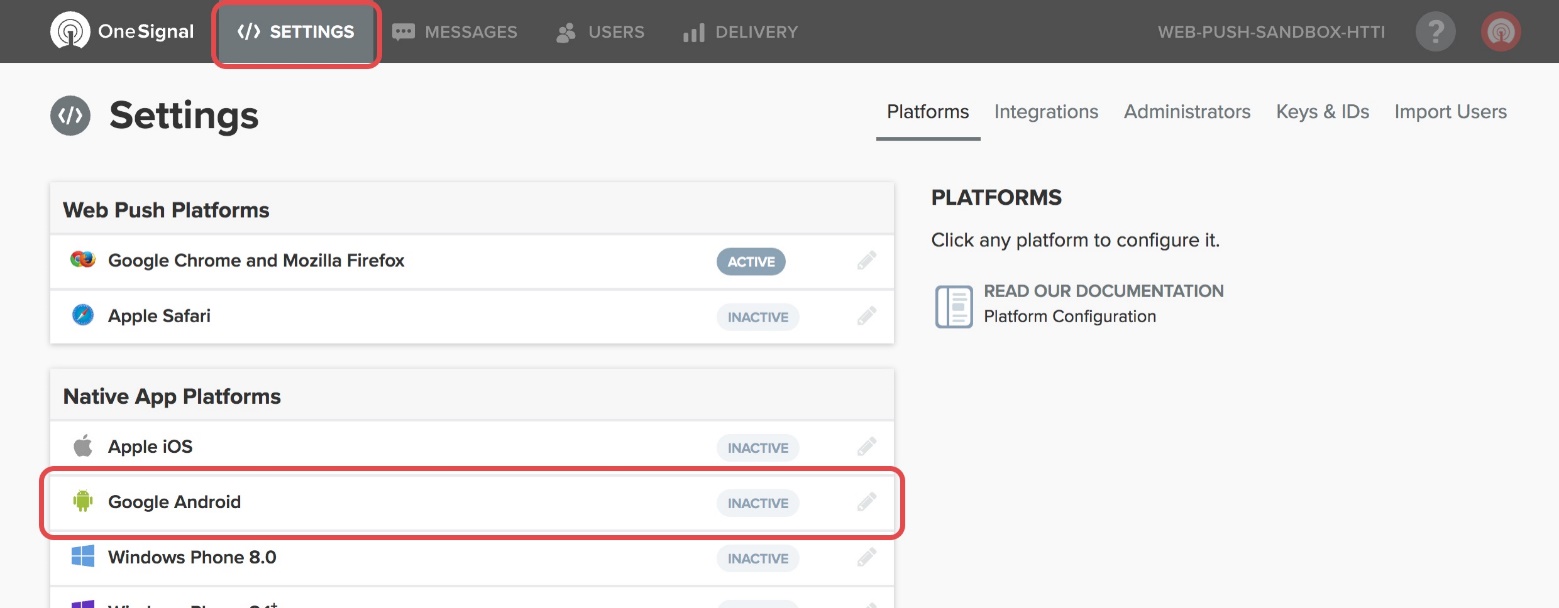
* You'll need your **Server key**, also known as the **Google Server API key**.
* You'll need your **Sender ID**, also known as the **Google Project Number**, later as well

[](https://files.readme.io/992ce63-FCM_Project_keys.png)

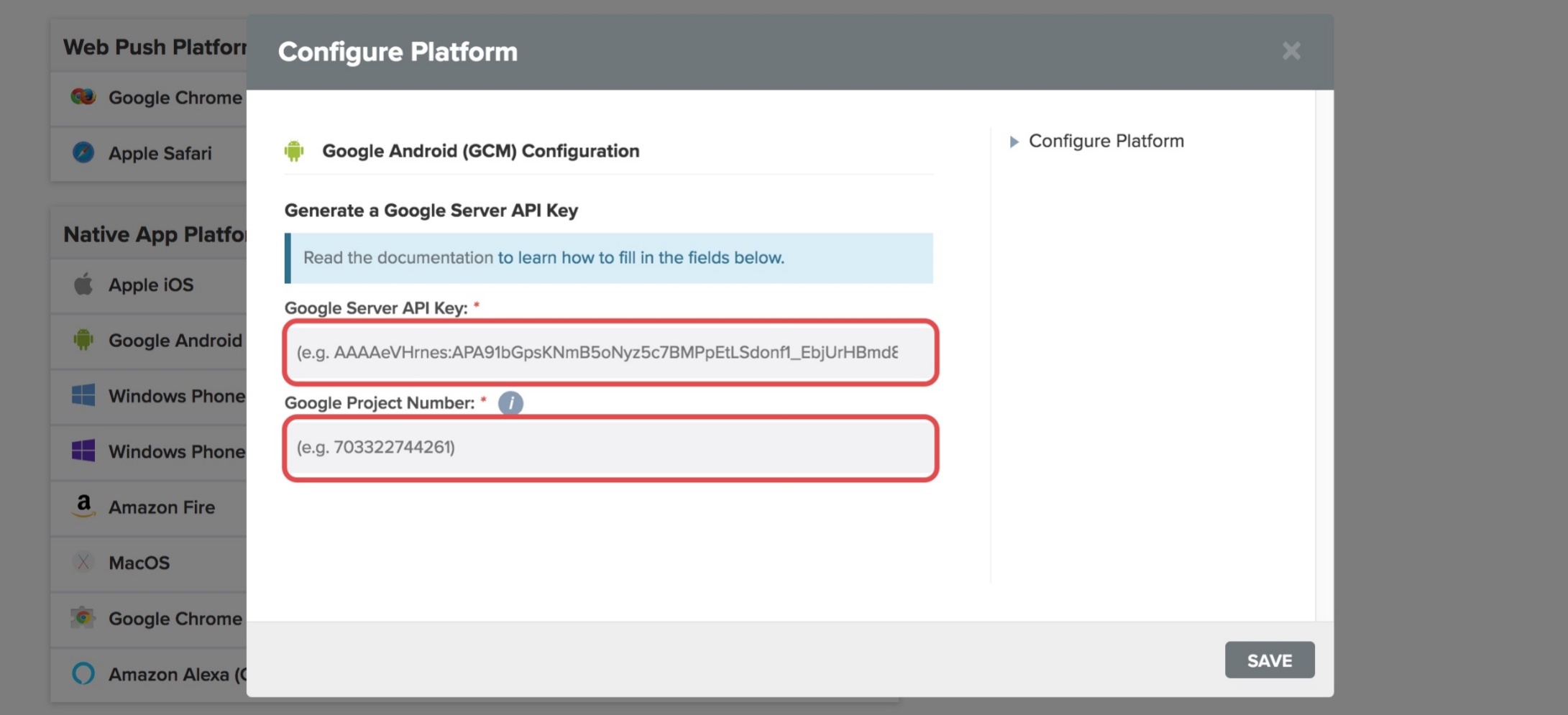
## 

## 3. Configure your OneSignal app's Android platform settings

**3.1** Go to Settings and press the Configure button to the right of Google Android.

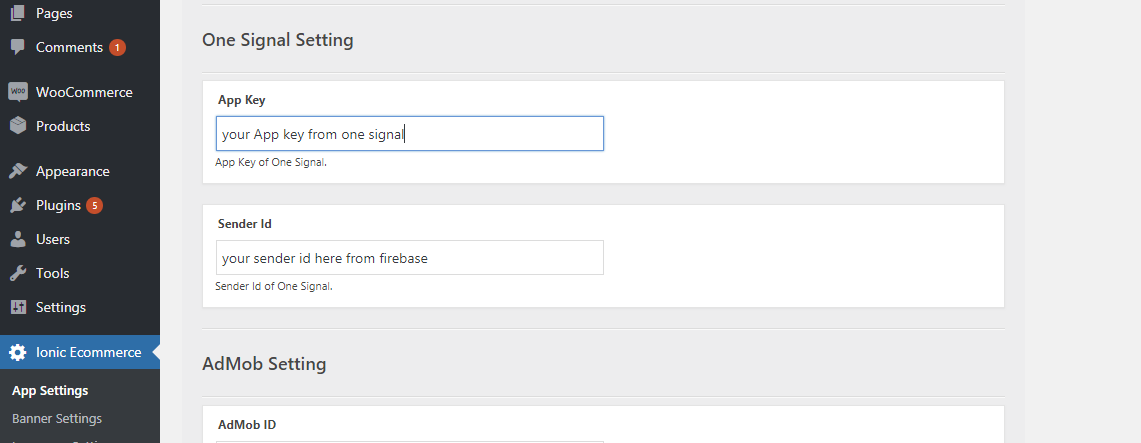
[](https://files.readme.io/e7700c4-settings-platforms-android.jpg)

**3.2** Paste your Google Server API Key and Google Project Number in here and press Save.

[](https://files.readme.io/d371be2-settings-platforms-android-configure.jpg)

After Creating this Google Server API Key and **Sender ID from the firebase project update them in your Admin plugin settings.**

**http://Your-site-url/wp-admin/admin.php?page=ionic-ecommerce**



**3.2) Config OneSignal for IOS**

# Generate an iOS Push Certificate

Required for all **iOS** apps.

### Required For Setup

* [A OneSignal Account](https://onesignal.com/) if you do not already have one
* Your OneSignal App Auth Key, available in [**Keys & IDs**](https://documentation.onesignal.com/docs/accounts-and-keys#section-keys-ids)
* A Mac with Xcode 8+. If you are using Xcode 7, PhoneGap Build, or Cordova, [follow this guide instead](https://documentation.onesignal.com/docs/generate-an-ios-push-certificate-alternate).

## 

## 1. Provisioning

### Option A: Try out our Automatic Provisioning Tool

We recently released a tool to automate this process!

Make sure your role for the team is Admin inside your Apple Developer account > Membership section. Otherwise you will not be able to use this tool.

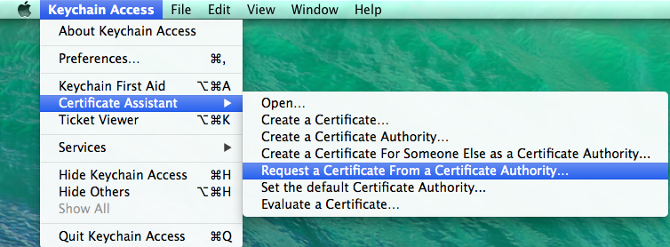
Simply follow the directions on [OneSignal's Provisionator Tool](https://onesignal.com/provisionator) then [continue to Step 4](https://documentation.onesignal.com/docs/generate-an-ios-push-certificate#section-4-upload-your-push-certificate-to-onesignal).

**-- OR --**

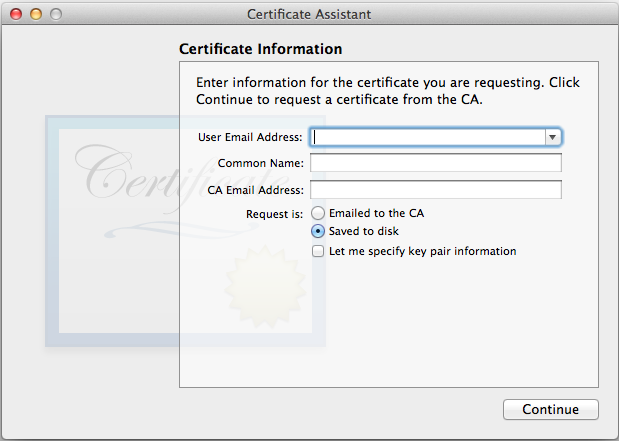
### Option B: Create Certificate Request Manually

**1.1** Open Keychain Access on your Mac OS X system. It may be located in "Applications" > "Utilities" > "Keychain Access"

**1.2** Select "Keychain Access">"Certificate Assistant">"Request a Certificate From a Certificate Authority..."

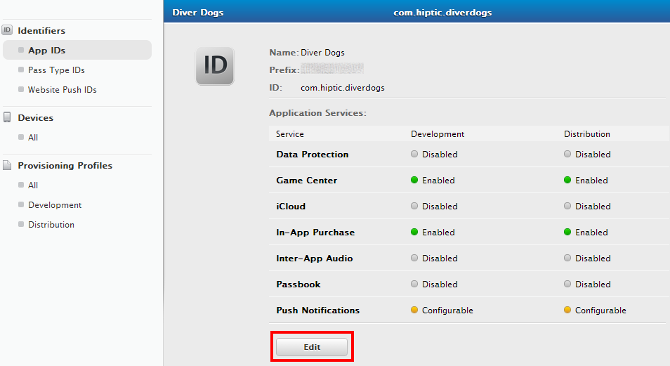
[](https://files.readme.io/sc7fGG3gRgWaL168mmS4_SC_KeychainAccess.png)

**1.3** Select the "Save to disk" option and enter your information in the required fields. This creates a certification request file that will be used later.

[](https://files.readme.io/Xl1qWa7fSVKf7Hhjhucr_SC_CertificateAssistant.png)

## 

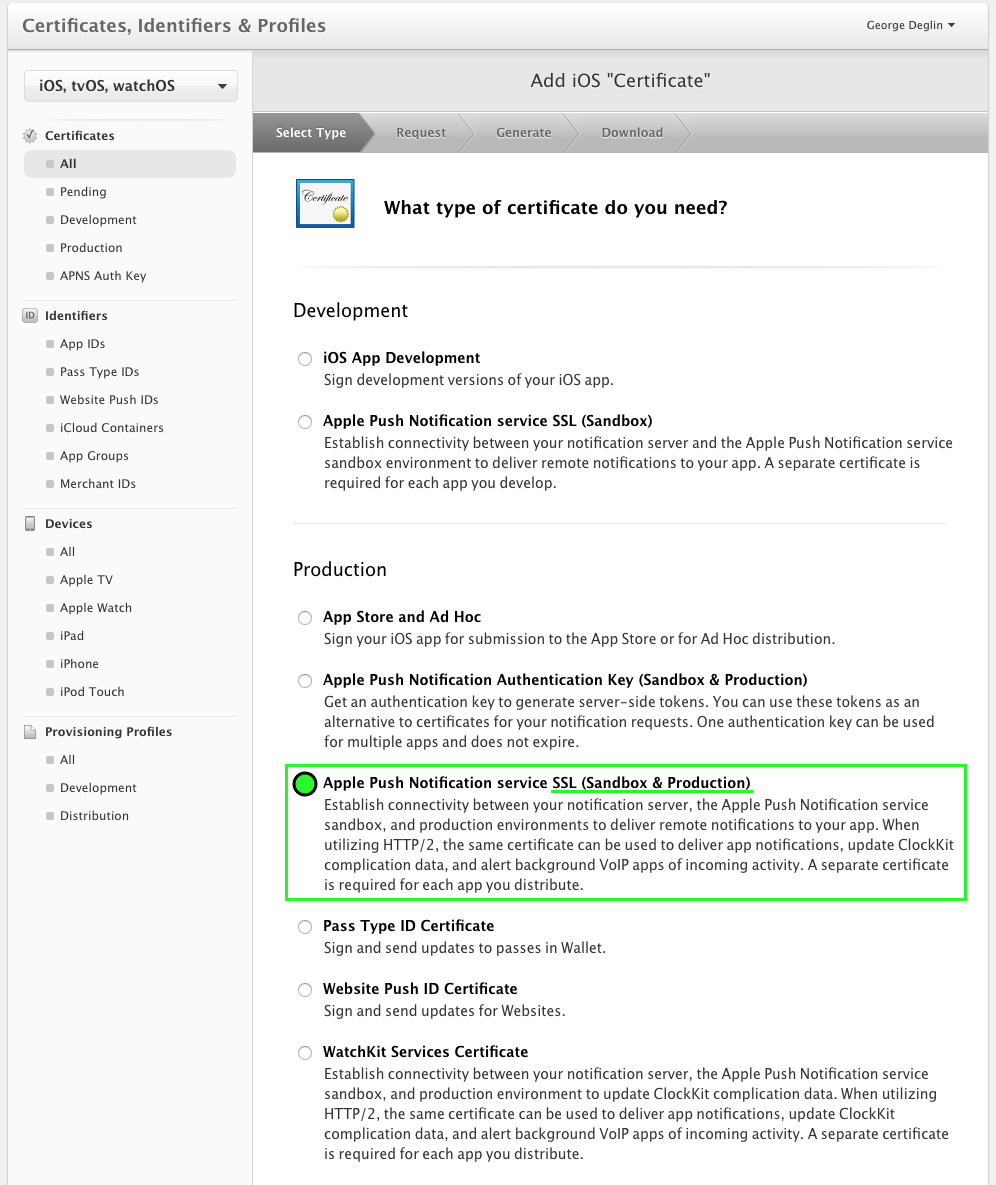
## 2. Enable Push Notifications and apply the Certification Request to generate Certificate

**2.1** Select your app from the [Apple's Developer site](https://developer.apple.com/account/ios/identifier/bundle) and press "Edit"[](https://files.readme.io/NHddNnCsRB2qDbvz4nkK_SC_SelectAppId.png)

**2.2** Scroll down to the bottom and enable Push Notifications. Press Done, but do not configure either Production or Development certificate.

Instead, go to [Add iOS Certificate](https://developer.apple.com/account/ios/certificate/create) and select "**Apple Push Notification service SSL (Sandbox & Production)**" and click Continue.

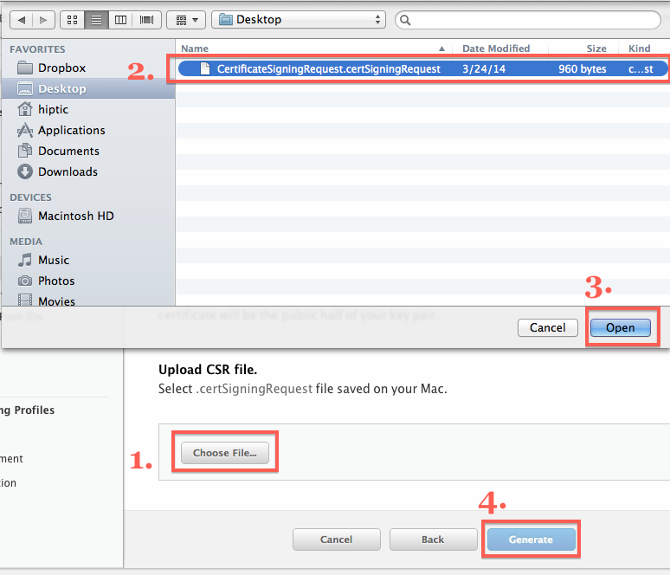
This certificate will be applicable to both Sandbox and Production environments, so you do not need a separate key for each one.

[](https://files.readme.io/5011511-Screenshot_2016-09-22_00.36.35.png)

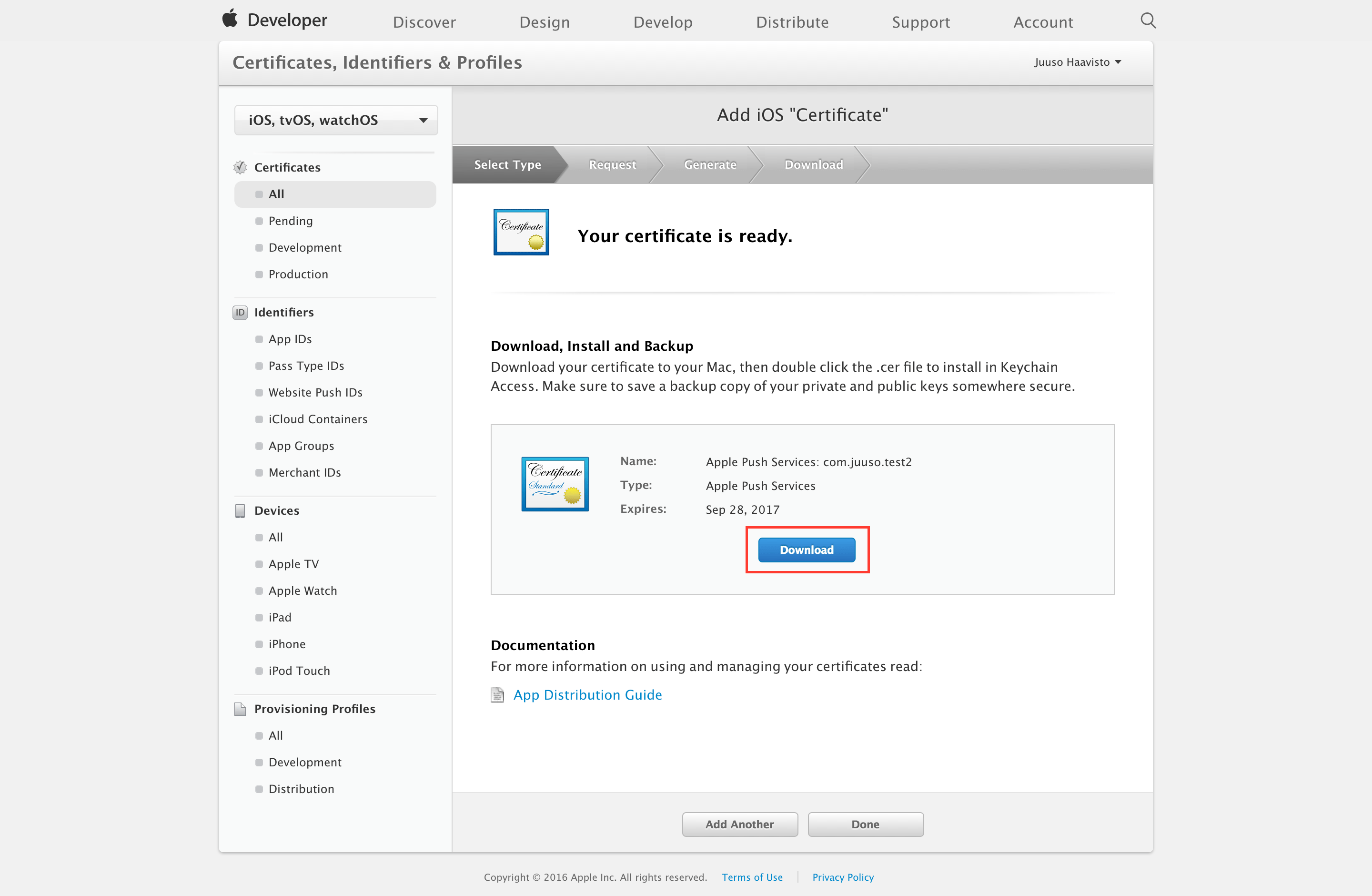
**2.3** Choose an App ID from the App ID pop-up menu, and click Continue.

**2.4** Press Continue

**2.5** Press "Choose File..", select the "certSigningRequest" file you saved in step 1, press open, and then press "Generate".

[](https://files.readme.io/s8GTOvNdTz6RxlzgTOA1_SC_PressChooseFile.png)

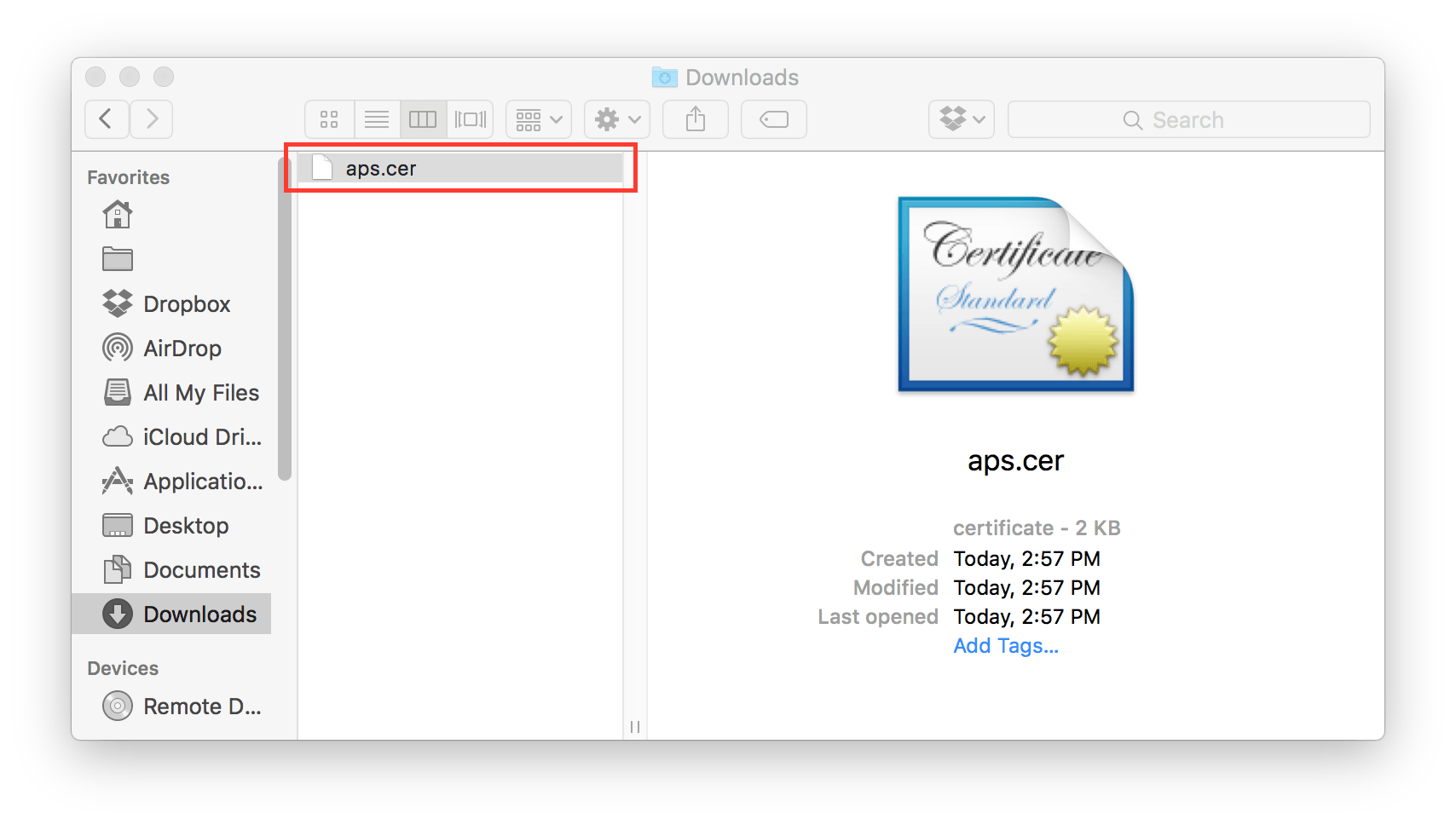
**2.6** Press Download to save your certificate

[](https://files.readme.io/4f5ccb7-Screen_Shot_2016-08-29_at_1.44.36_PM-fs8.png)

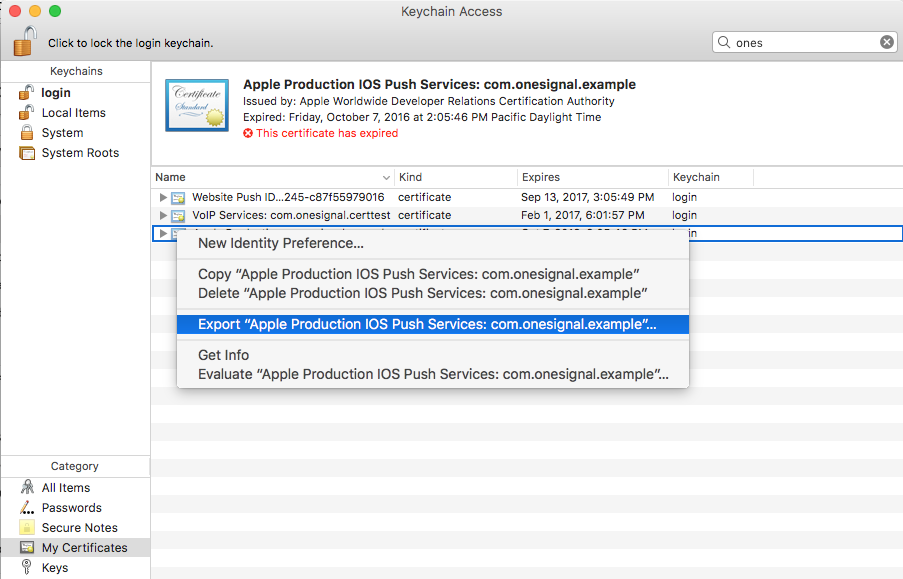
## 

## 3. Creating a Private Key

**3.1** Open the .cer file you downloaded in the last step by double clicking on it in Finder.

[](https://files.readme.io/0e27ab2-cert-fs8.png)

**3.2** After a few seconds the "Keychain Access" program should pop up. Select Login > My Certificates then right click on your key in the list and select "Export"

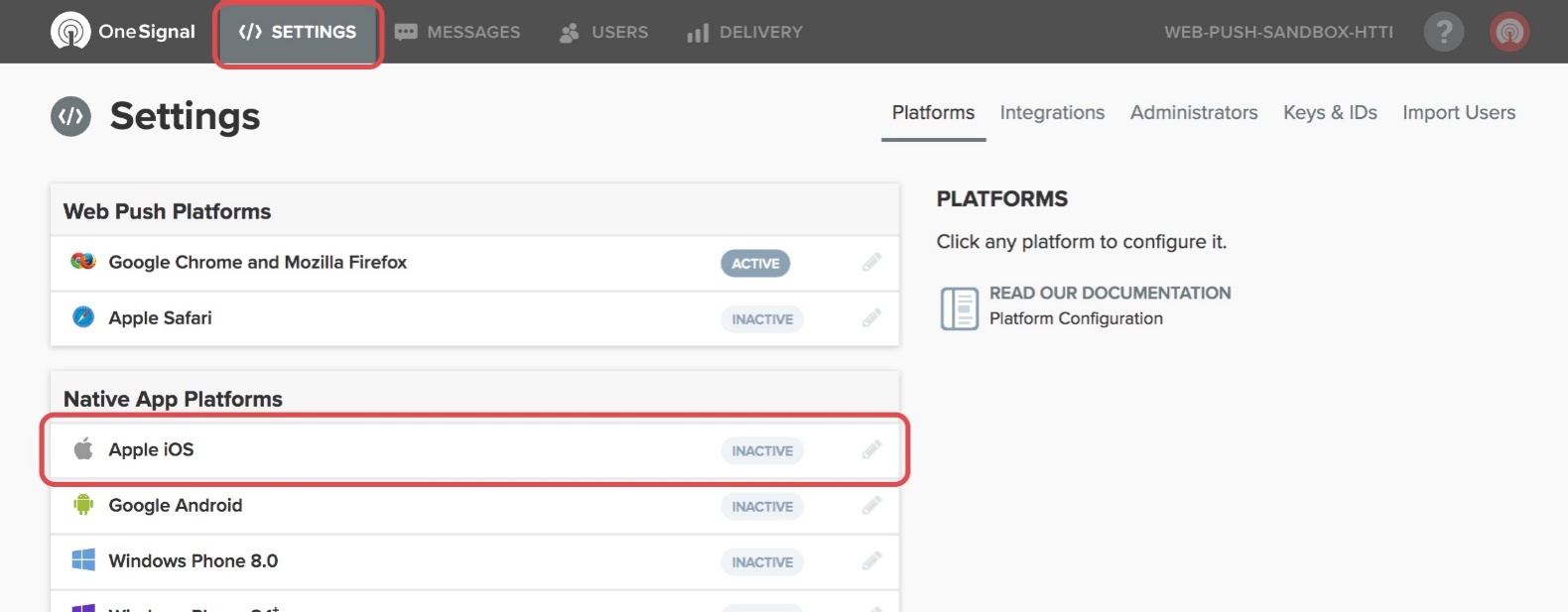
[](https://files.readme.io/7d484fe-Screenshot_2016-12-27_14.08.25.png)

**3.3** Give the file a unique name using the .p12 extension, and press save. You will have an option to protect the file with a password. If you add a password, you need to enter this same password on OneSignal.

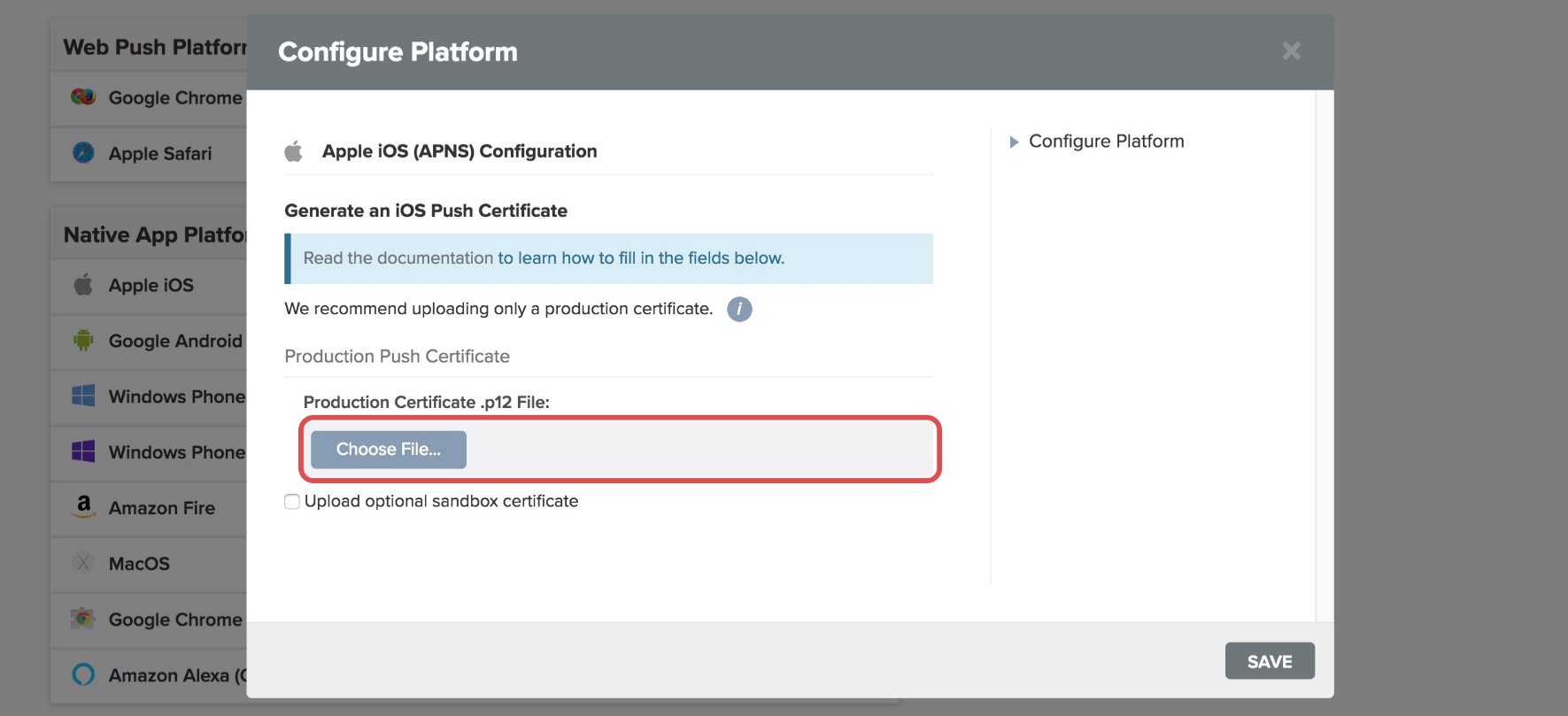
## 4. Upload Your Push Certificate to OneSignal

If you haven't already, you should [set up your OneSignal account](https://onesignal.com/).

**4.1** Select your app from the All Apps page in OneSignal, then go to "App Settings" and press Configure to the right of the Apple iOS Settings.

[](https://files.readme.io/65f91ec-settings-platforms-ios.jpg)

**4.2** Select the .p12 you exported along with a password if you added one and press Save.

[](https://files.readme.io/5c1cd65-settings-platforms-ios-configure.jpg)

## 5. Install Tools and Deploy app

Deploy Mobile app you need install some Tools

* Download & Install NodeJS (8.9.0): https://nodejs.org/en/download/ After installing Node JS, install Cordova and Ionic using the following command:



npm install -g ionic@3.2.0

#

then install cordova



npm install -g cordova@3.2.0

#

Application is build and tested using the

* Cordova CLI: 7.1.0
* Ionic Framework Version: 3.9.2
* Ionic CLI Version: 3.19.0
* Node Version: v8.9.0

**It is recommending to use the environment with above Specifications.**

Once you already installed, prepare your project

Open your project directory and ‘**right click**’ on white space while holding ‘Shift’ button and select option  “**Open command window here**”.

For example we have project name “**ionicshop\_laravel\_demo1**“. We will open this folder and **right click** on white space while holding **shift key.**Now select**“Open command window here”.**

Now you have cmd opened with your project directory path loaded.

Hence the environment is change it is recommended to rebuild the sass. Type in cmd “**npm rebuild node-sass**”

## 5.2. Build Android App for Testing On Windows

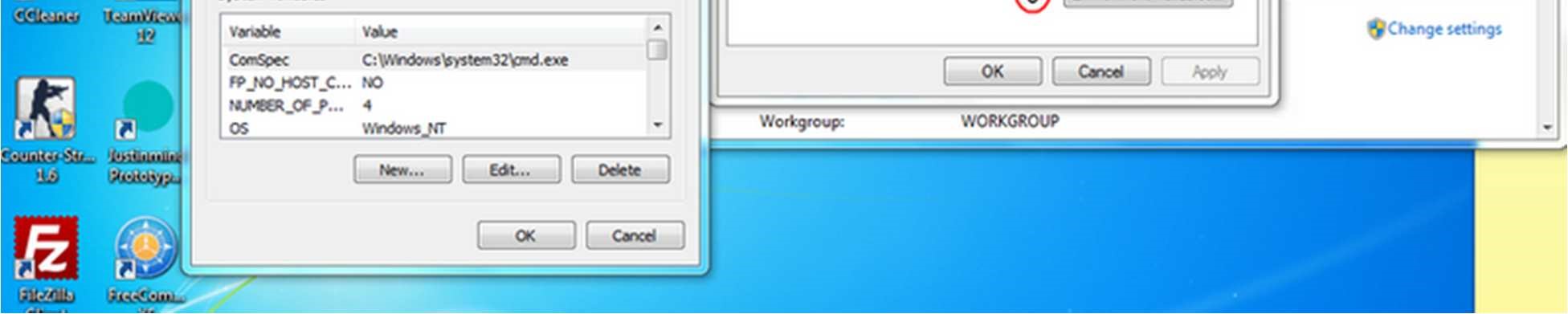
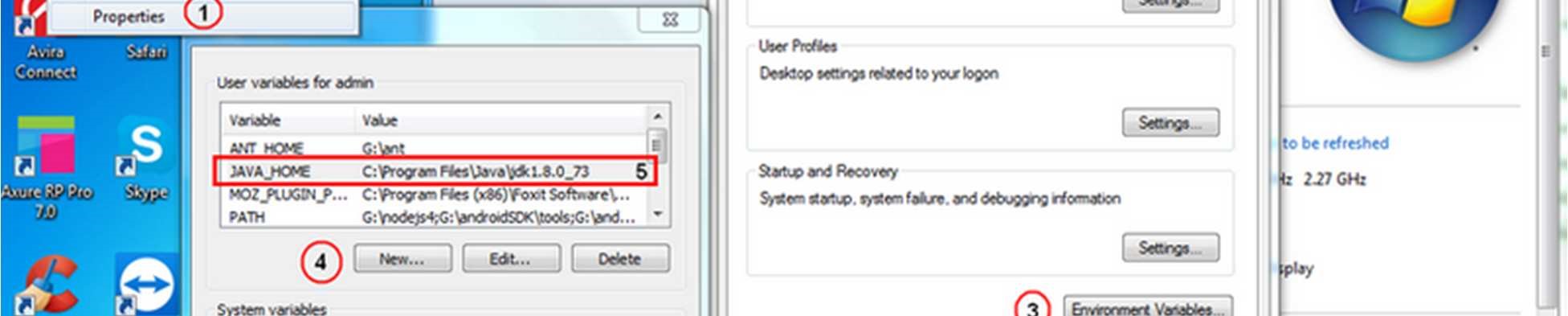
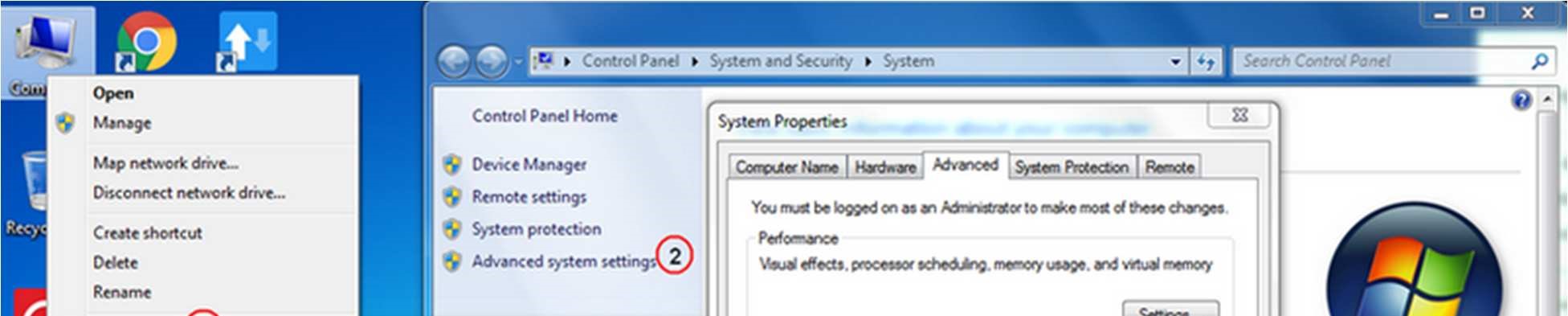
##### 5.2.1. Install Java Development Kit (JDK)

* **Step 1**: Download & Install Java Development Kit

(JDK) latest: http://www.oracle.com/technetwork/java/javase/downloads/jdk8downloads-2133151.html

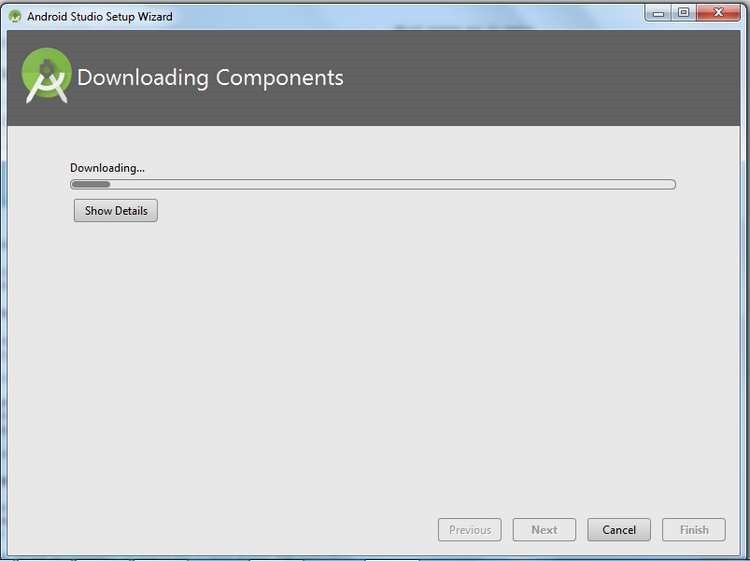
* **Step 2**: Set the JAVA\_HOME environment variable to the location of your JDK installation

In File Explorer right-click on the **This PC (or Computer) icon**, then click **Properties -> Advanced System Settings -> Environmental Variables**.



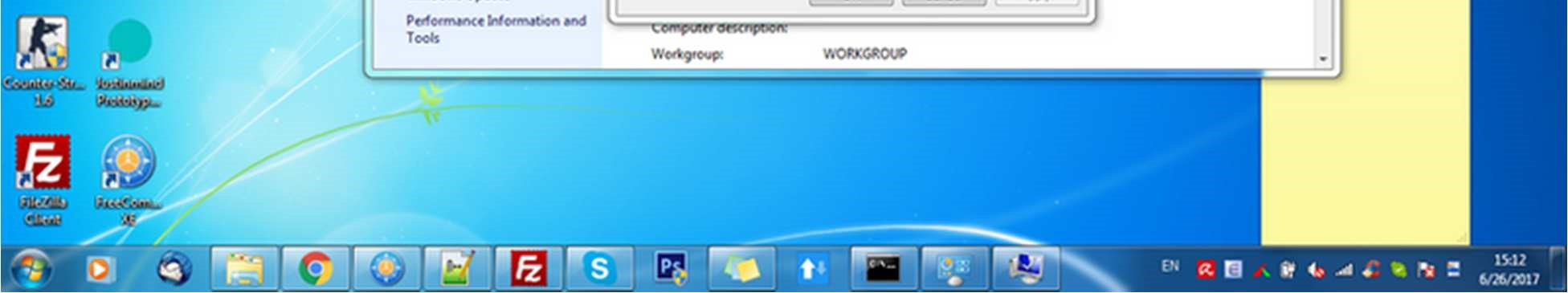
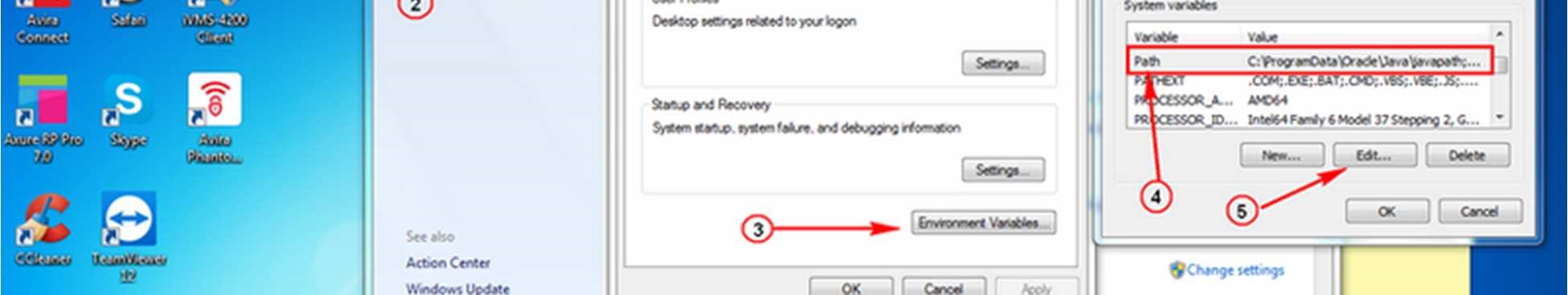
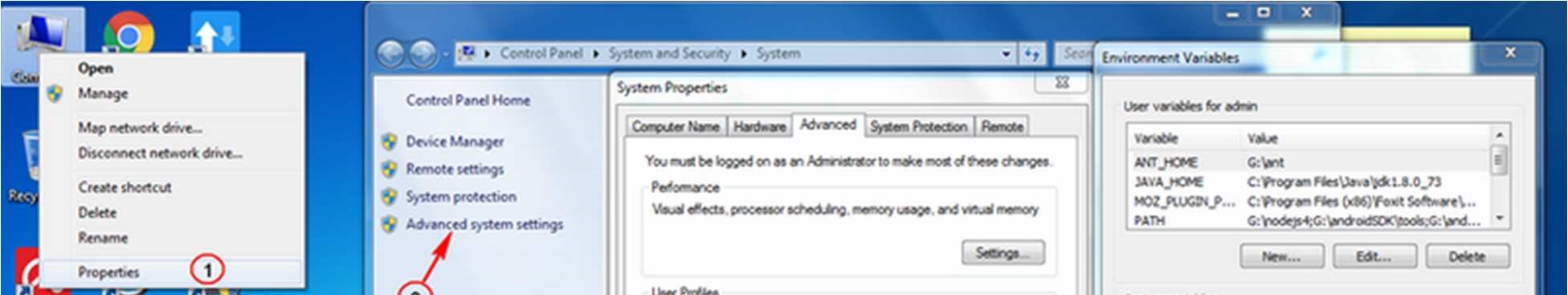
##### 5.2.2. Android Studio & Android SDK

* **Step 1**: Download & Install Android studio: https://developer.android.com/studio/install.html?pkg=studio
* **Step 2**: Open Android Studio, Click **Next** to **download all Package for Android SDK**



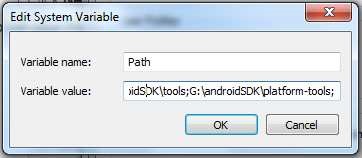
##### 5.2.3. Setting environment variables for Android SDK

* **Step 1**: In File Explorer right-click on the **This PC (or Computer) icon**, then click **Properties -> Advanced System Settings -> Environmental Variables**.



* **Step 2**: Select the **PATH** variable and press **Edit**. Add more path to android-sdk\tools and android-sdk\platform-tools to **PATH**, example:

G:\androidSDK\tools;G:\androidSDK\platform-tools;

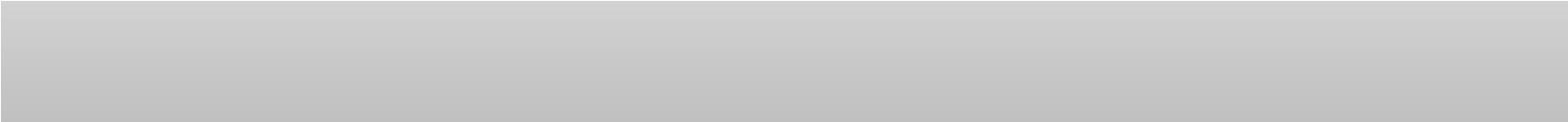


##### 5.2.4. Add Platform & Build to APK

Move to the YOUR\_PROJECT folder in the terminal, run the following command:



Ionic cordova platform add android@6.3.0

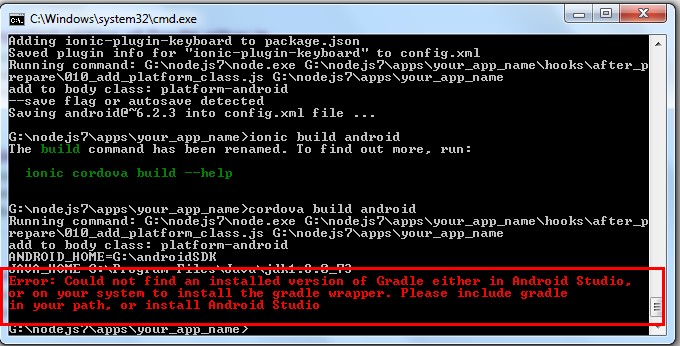


Ionic cordova build android --prod

When success, file apk will in **YOUR\_PROJECT/platforms/android/build/outputs/apk with name "androiddebug.apk".**

##### 5.2.5. Error: Could not find an installed version of gradle either in Android Studio

* if you see this error:



* Please install **gradle** at here: https://gradle.org/install

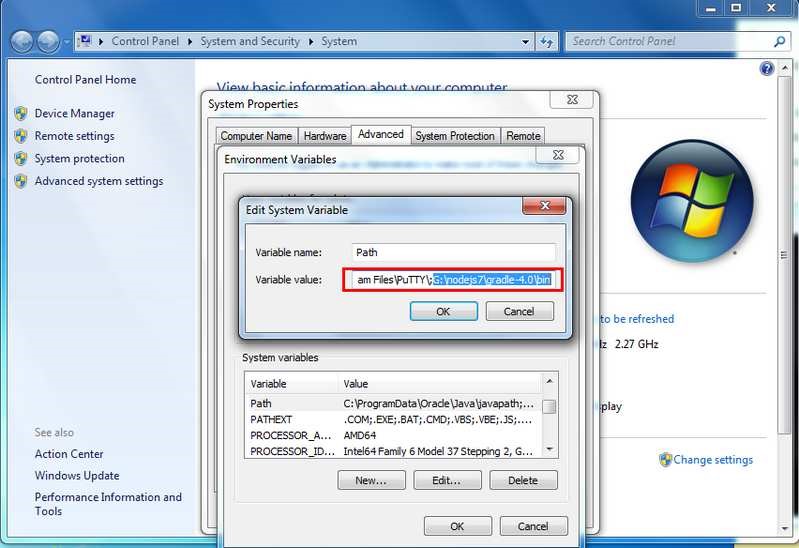
Step 1: Download https://gradle.org/releases

Step 2: unzip to your computer

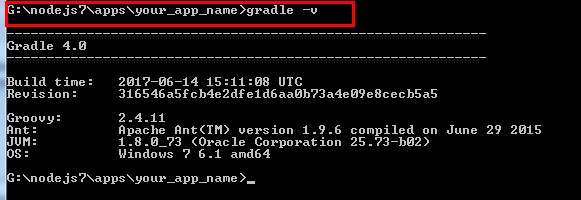
Step 3: In File Explorer right-click on the **This PC (or Computer) icon**, then click **Properties -> Advanced System Settings -> Environmental Variables**.

Under System Variables select **Path**, then click **Edit**. Add an entry for ***G:/nodejs7/gradle-***

***4.0/bin***. Click OK to save.



Step 4: **Close** Window Command Line and **open** it again Step 5: test with command: gradle -v if it is showing result for any version of gradle, it is okay.



Step 6: do **"**Build to APK" again.

Read More: <http://cordova.apache.org/docs/en/6.x/guide/platforms/android/index.html#page-tocsource>

## 5.2. Setting up you Mac for testing app and Deploy App

Please follow below link for setup

https://ionicframework.com/docs/developer-resources/platform-setup/mac-setup.html

# Build iOS app for Testing

## iOS Devices

Unlike Android, iOS developers need to generate a provisioning profile to code sign their apps for testing. The good news is that, as of iOS9, you can develop and test your apps on your iOS device without a paid Apple Developer account. This is particularly great for developers who want to try out mobile development with Ionic, since it saves the cost but still provides a lot of the features of having a full Apple Developer account. For a full breakdown of the features included, check out [Apple’s docs](https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/SupportedCapabilities/SupportedCapabilities.html#//apple_ref/doc/uid/TP40012582-CH38-SW1).

### Requirements

* Xcode 7 or higher
* iOS 9
* A free [Apple ID](https://appleid.apple.com/) or paid Apple Developer account

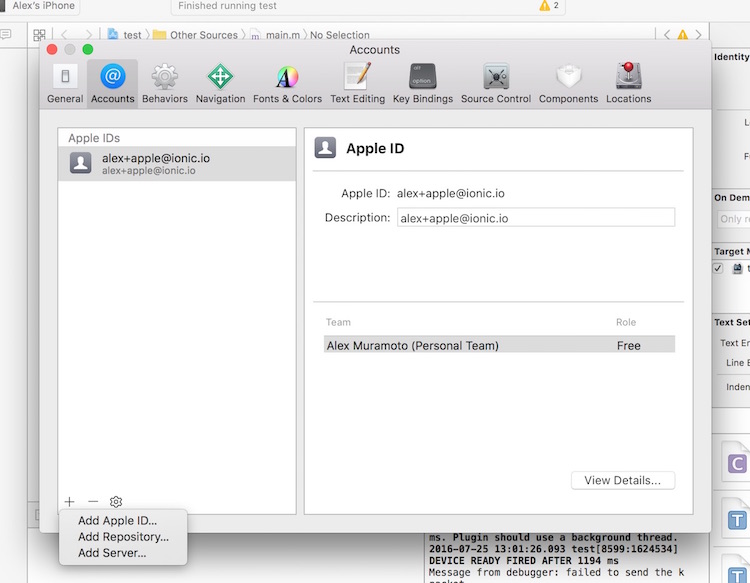
### Creating a Provisioning Profile

To start, you’ll need to set up a provisioning profile to code sign your apps.

#### Using an Apple ID

1. Open Xcode preferences (Xcode > Preferences…)
2. Click the ‘Accounts’ tab
3. Login with your Apple ID (+ > Add Apple ID…)

Once you’ve successfully logged in, a new ‘Personal Team’ with the role ‘Free’ will appear beneath your Apple ID.



#### Using an Apple Developer Account

Creating a provisioning profile with a paid Apple Developer account is a little bit more involved. For full instructions, check out [Launching Your App on Devices](https://developer.apple.com/library/content/documentation/IDEs/Conceptual/AppDistributionGuide/LaunchingYourApponDevices/LaunchingYourApponDevices.html) in the Apple Developer docs.

### Running Your App

1. Run a production build of your app with ionic cordova build ios --prod
2. Open the .xcodeproj file in platforms/ios/ in Xcode
3. Connect your phone via USB and select it as the run target
4. Click the play button in Xcode to try to run your app

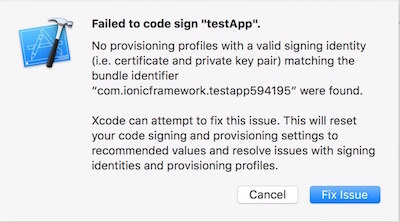
Oops, code signing error! No problem.

### Code Signing Your App

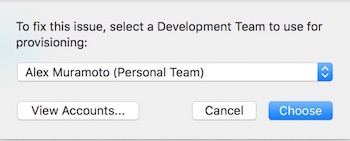
Next, you’ll need to code sign your app. How you do this will depend on if you are running Xcode 8 or an earlier version.

#### Xcode 7 and Earlier

If you are running Xcode 7 or earlier, you’ll get a code signing error that looks like this when you try to run the app:

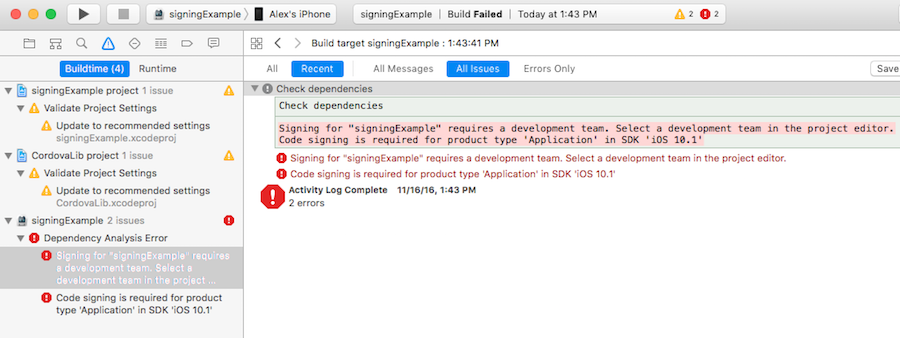


Click the ‘Fix Issue’ button, then select your ‘Personal Team’ profile.



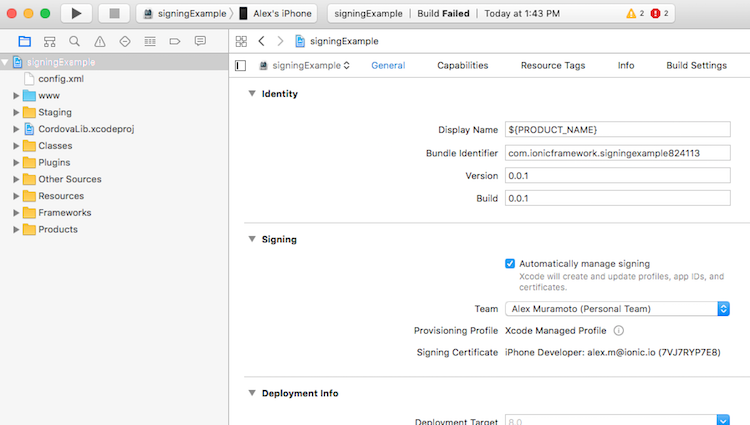
#### Xcode 8

If you are running Xcode 8, the code signing error will appear as a buildtime error, rather than as a pop-up:



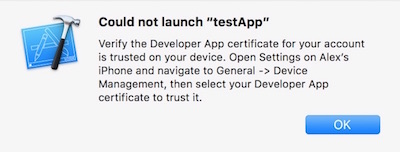
To select the certificate to sign your app with, do the following:

1. Go to the ‘Project Editor’ by clicking the name or your project in the ‘Project Navigator’
2. Select the ‘General’ section
3. Select the team associate with your signing certificate from the ‘Team’ dropdown in the ‘Signing’ section



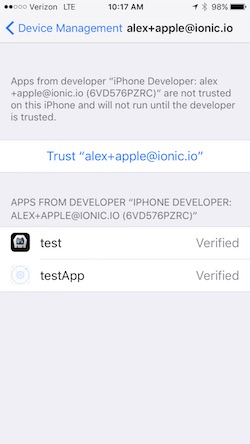
### Trusting the Certificate

Once you’ve code signed your app, you should get a launch error that looks like this. On Xcode 7 and below you’ll see this automatically. On Xcode 8 it will appear the next time you try to run the app:



To get past this, we have to tell our iOS device to trust the certificate we code signed our app with:

1. Open the ‘Settings’ app on your iOS device
2. Go to ‘General > Device Management’. You’ll see the email address associated with the Apple ID or Apple Developer account you used to code sign your app.
3. Tap the email address
4. Tap ‘Trust <your\_email>’:



Now, go back to Xcode and hit that play button or run ionic cordova run ios --device from the command line to install and launch your app on your iOS device.

Read more this:

https://cordova.apache.org/docs/en/latest/guide/platforms/ios/index.html

**Deploy Android app to store**

Deploying to an Android device is a fairly straightforward process. If you have a working Android development environment, you’re ready to go.

### Requirements

* [Java JDK](http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html)
* [Android Studio](https://developer.android.com/studio/index.html)
* Updated Android SDK tools, platform and component dependencies. Available through Android Studio’s [SDK Manager](https://developer.android.com/studio/intro/update.html)

### Running Your App

To run your app, all you have to do is enable USB debugging and Developer Mode on your Android device, then run ionic cordova run android --device from the command line.

This will produce a debug build of your app, both in terms of Android and Ionic’s code

Enabling USB debugging and Developer Mode can vary between devices, but is easy to look up with a Google search. You can also check out [Enabling On-device Developer Options](https://developer.android.com/studio/run/device.html#developer-device-options) in the Android docs.

### Production Builds

To run or build your app for production, run

ionic cordova run android --prod --release

*# or*

ionic cordova build android --prod --release

This will minify your app’s code as Ionic’s source and also remove any debugging capabilities from the APK. This is generally used when deploying an app to the Google Play Store.

### Sign Android APK

If you want to release your app in the Google Play Store, you have to sign your APK file. To do this, you have to create a new certificate/keystore.

Let’s generate your private key using the keytool command that comes with the JDK:

keytool-genkey-v-keystoremy-release-key.jks-keyalgRSA-keysize 2048 -validity 10000 -aliasmy-alias

You’ll first be prompted to create a password for the keystore. Then, answer the rest of the nice tools’s questions and when it’s all done, you should have a file called my-release-key.jks created in the current directory.

**Note**: Make sure to save this file somewhere safe, if you lose it you won’t be able to submit updates to your app!

To sign the unsigned APK, run the jarsigner tool which is also included in the JDK:

jarsigner-verbose-sigalgSHA1withRSA-digestalgSHA1-keystoremy-release-key.jksandroid-release-unsigned.apkmy-alias

This signs the APK in place. Finally, we need to run the zip align tool to optimize the APK. The zipalign tool can be found in /path/to/Android/sdk/build-tools/VERSION/zipalign. For example, on OS X with Android Studio installed, zipalign is in ~/Library/Android/sdk/build-tools/VERSION/zipalign:

zipalign-v 4 android-release-unsigned.apkHelloWorld.apk

To verify that your apk is signed run apksigner. The apksigner can be also found in the same path as the zipalign tool:

apksignerverifyHelloWorld.apk

Now we have our final release binary called HelloWorld.apk and we can release this on the Google Play Store for all the world to enjoy!

All steps can also be found here: [Android SDK docs](https://developer.android.com/studio/publish/app-signing.html#signing-manually)

#### 6. Upload to Google Play Store

Now that we have our release APK ready for the Google Play Store, we can create a Play Store listing and upload our APK.

To start, you’ll need to visit the Google Play Store Developer Console:

https://play.google.com/apps/publish/ and create a new developer account. Unfortunately, this is not free. However, the cost is only $25 compared to Apple’s $99.

Once you have a developer account, you can go ahead and click “Publish an Android App on Google Play”

Please follow below link to upload your app to store

https://www.youtube.com/watch?v=vIusIlp6YGE

#### 7. Upload to App Store

#### To upload app to app store please Follow below Link.

https://www.youtube.com/watch?v=f3Y1amVx\_zM

**8) FAQ’S**

**Q1 : How to change assets inside application ?**

**Ans** **:** Assets are very easy to change open **app/src/assets** folder contain all the icons images and other assets used in app you can change them as you like.

**Q2 : How to Change icon and splash screen ?**

**Ans**  **:**To change the application icon and splash screen please visit.

<https://ionicframework.com/docs/cli/cordova/resources/>

**Q3 : How to change intro slides of application ?**

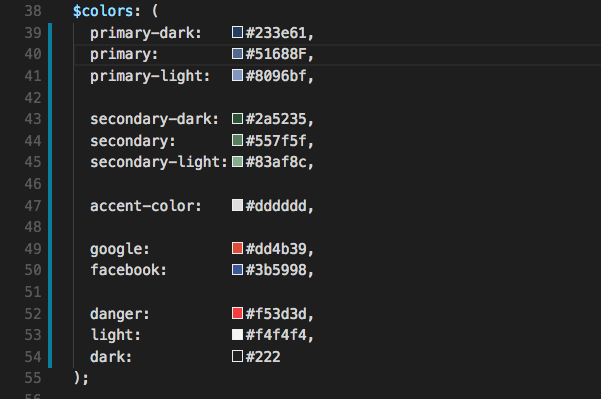
**Ans**  **:**Intro slides are present inside **app/src/assets/intro** folder contain the images of intro slides change them according to your requirement.

**Q4 : How to set Home page and Category Page?**

**Ans:** Home Page and Category page can be changed from the**wordpress**easily.

**Q5 : How to change the Theme and Colors Using sass?**

**Ans:**

   
**$positive**is used to change top bars  **$dark**is used on side panel color while  **$accent-color, $accent-color-light, $accent-color-dark**is used on buttons ,text high light badges.