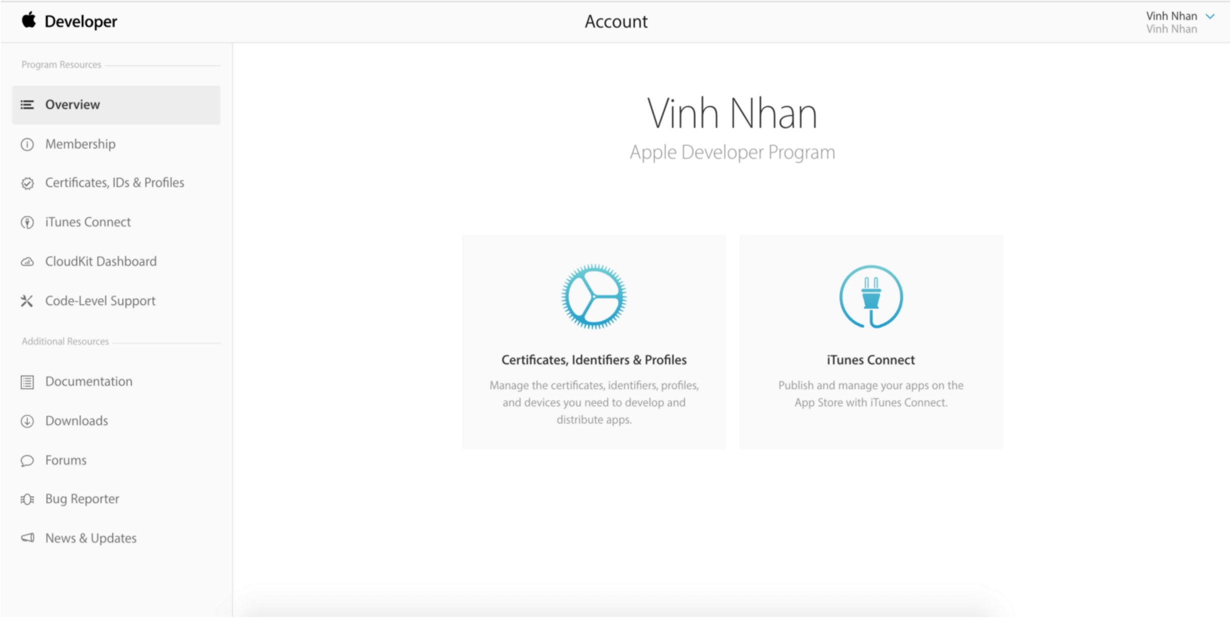
Implementing Push Notifications on iOS with Firebase

[*Created by Kesiyamol Varkey*]

[http://www.appcoda.com/firebase-push-notifications/]

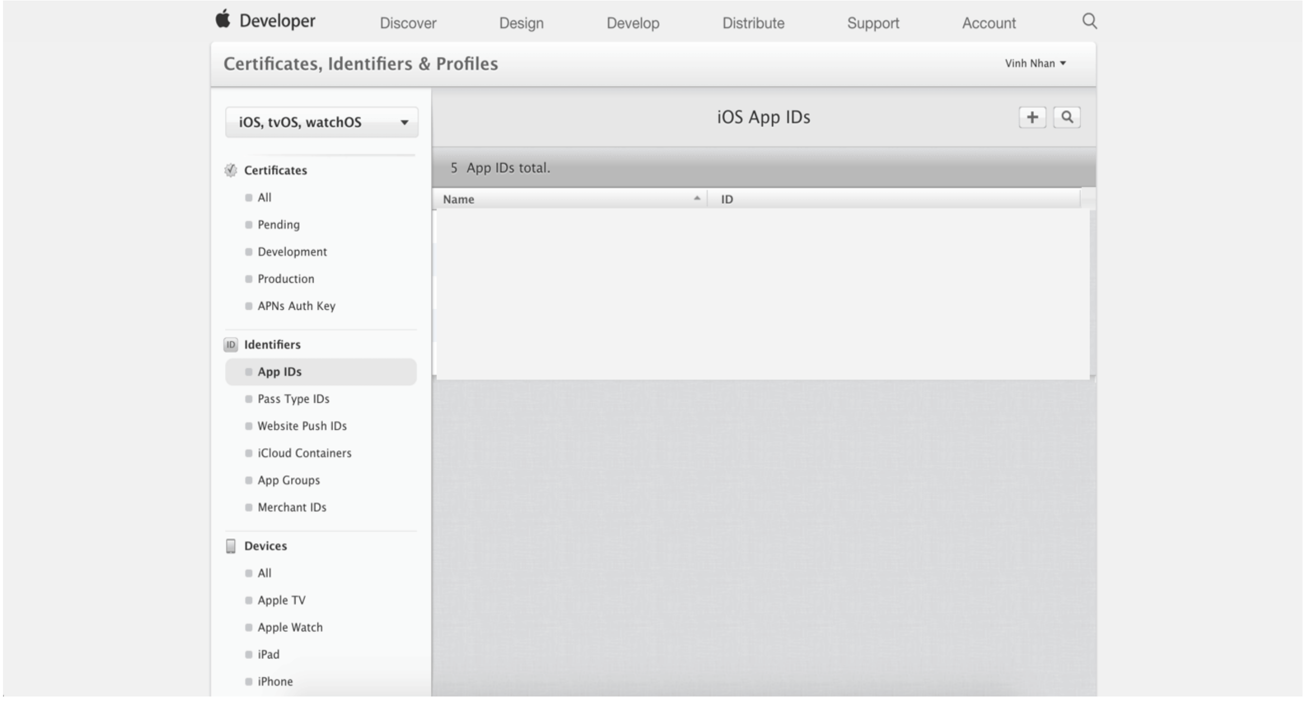
login to your Apple Developer account. Once logged in, you will be re-directed to the Apple Developer homepage. From there, you should see "Account" at the top navigation bar. Click that option.



Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

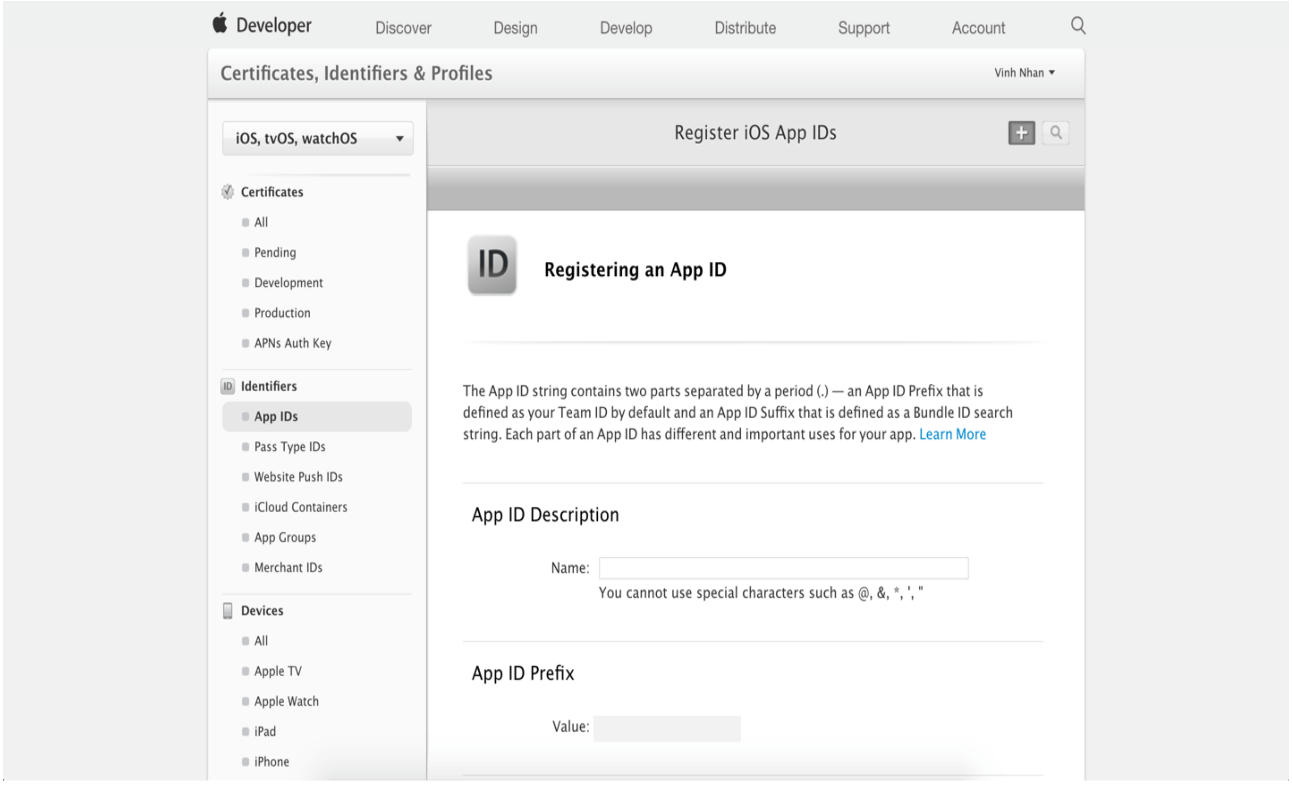
Publish and manage your apps on the App Store with iTunes Connect.

Select "Certificates, IDs & Profiles." Look to left side bar and there should be a section called "Identifiers." Under that section, there is a link that says "App IDs." Press that.You should see all your iOS App IDs.



5 App IDs total.

Now at the top right, you should see a + button.



Register iOS App IDs

Registering an App ID

The App ID string contains two parts separated by a period (.) — an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. Learn More

App ID Description

Name:

You cannot use special characters such as @, &, *, ', "

App ID Prefix

Value:

We now need to fill out the followings:

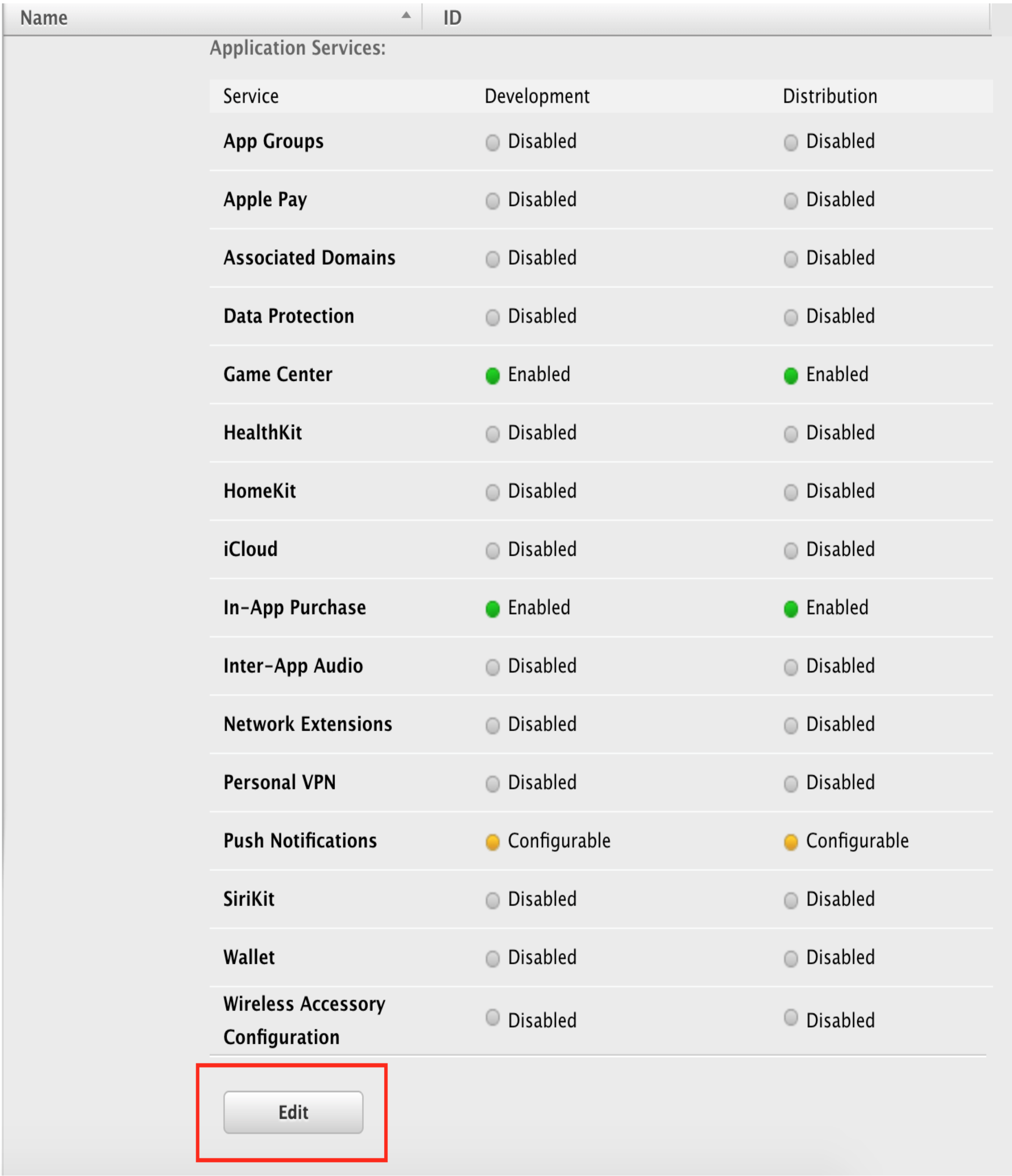
- **App ID Description—Name**. Here, you should put your app's name (e.g. Firebase Notification Demo)
- **App ID Suffix—Explicit App ID—Bundle ID**. Here, you need to select a unique bundle identifier for your app (e.g. com.appcoda.firebasenotificationsdemo). Please make sure you use your own bundle ID instead of using mine.

Then under App Services, tick "Push Notifications." Press continue.

After that, you will be redirected to a "Confirm your App ID" page. Press register.

Now we are back to our "iOS App IDs" page. Look for the App ID you just created. Press on it and you should see a a drop down of Application Services.

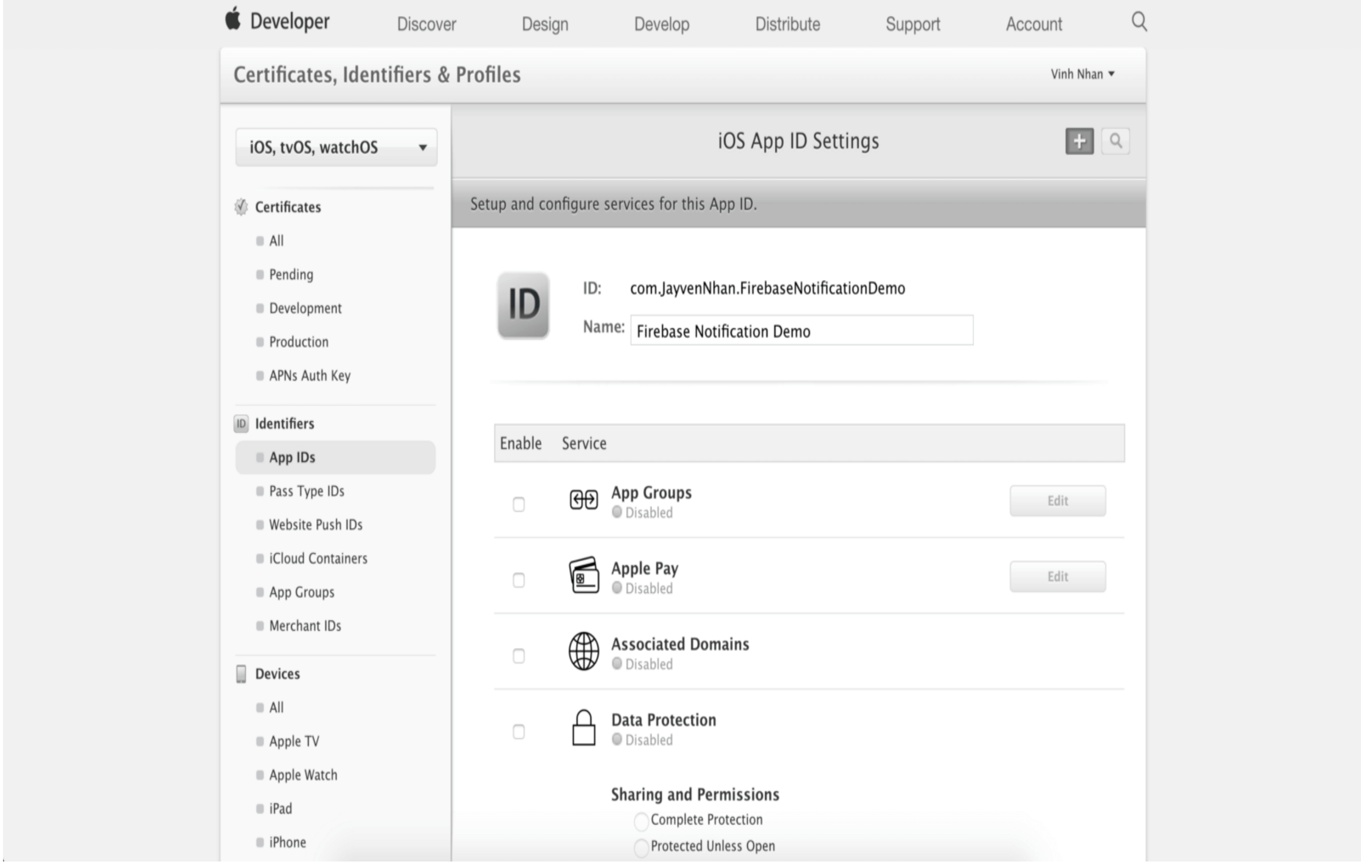
Scroll down until you reach the end of the drop down and you should see an "Edit" button. Press that.



Application Services:

Service	Development	Distribution
App Groups	○ Disabled	○ Disabled
Apple Pay	○ Disabled	○ Disabled
Associated Domains	○ Disabled	○ Disabled
Data Protection	○ Disabled	○ Disabled
Game Center	🟢 Enabled	🟢 Enabled
HealthKit	○ Disabled	○ Disabled
HomeKit	○ Disabled	○ Disabled
iCloud	○ Disabled	○ Disabled
In-App Purchase	🟢 Enabled	🟢 Enabled
Inter-App Audio	○ Disabled	○ Disabled
Network Extensions	○ Disabled	○ Disabled
Personal VPN	○ Disabled	○ Disabled
Push Notifications	🟠 Configurable	🟠 Configurable
SiriKit	○ Disabled	○ Disabled
Wallet	○ Disabled	○ Disabled
Wireless Accessory Configuration	○ Disabled	○ Disabled

The “iOS App ID Settings” page will show up.



iOS App ID Settings

Setup and configure services for this App ID.

ID: com.JayvenNhan.FirebaseNotificationDemo

Name: Firebase Notification Demo

Enable	Service
□	App Groups Disabled
□	Apple Pay Disabled
□	Associated Domains Disabled
□	Data Protection Disabled

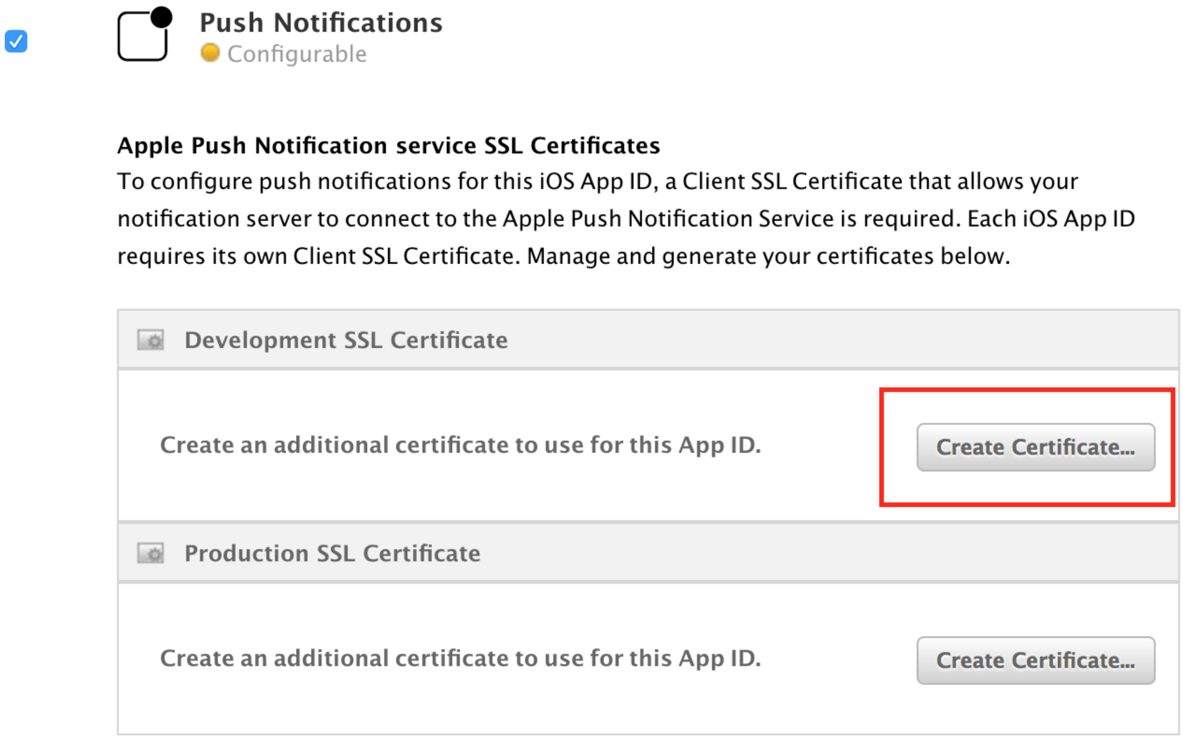
Sharing and Permissions

○ Complete Protection

○ Protected Unless Open

Scroll all the way down until you see “Push Notifications.”

It is time for us to create a “Client SSL Certificate.” This will allow our notification server (Firebase) to connect to the Apple Push Notification Service. Under Development SSL Certificate, press on the “Create Certificate…” button.

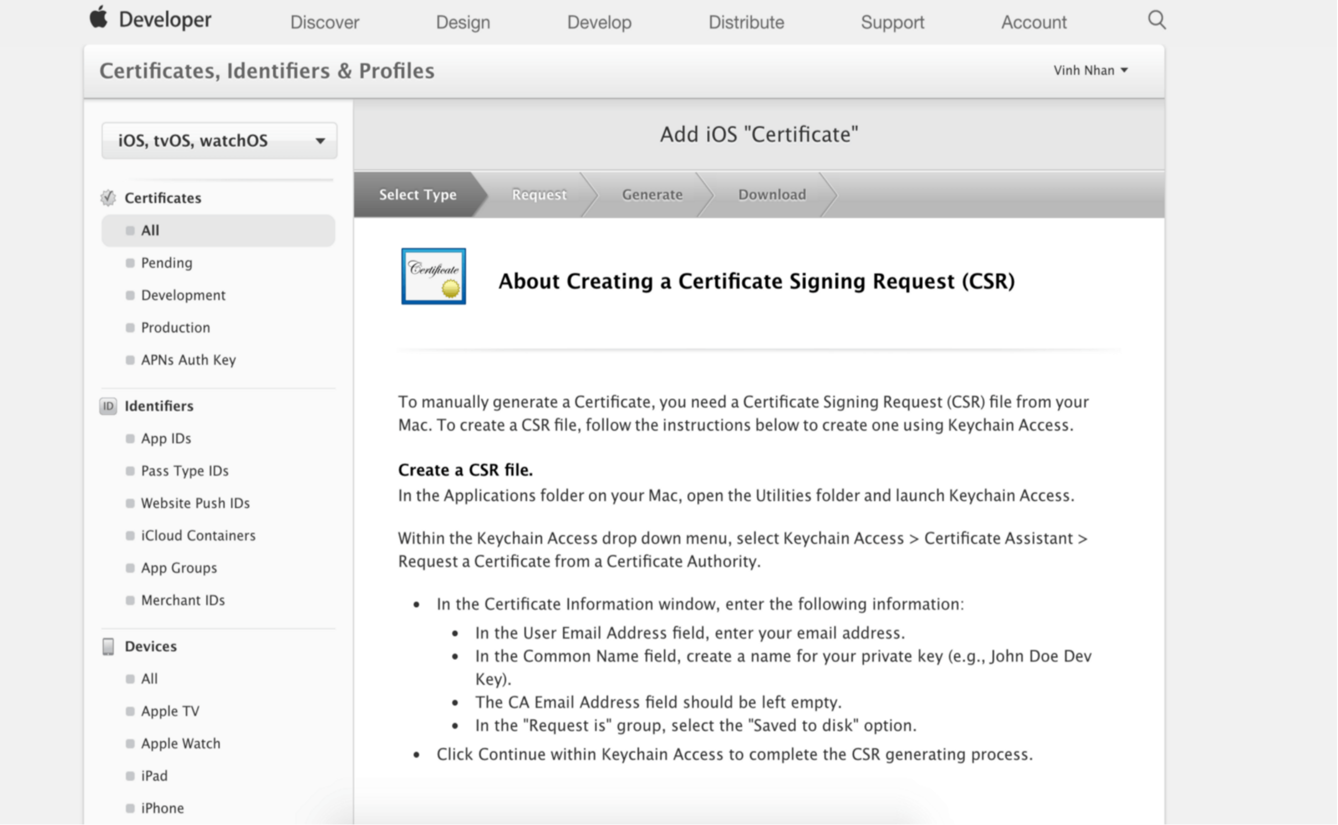


Apple Push Notification service SSL Certificates
To configure push notifications for this iOS App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each iOS App ID requires its own Client SSL Certificate. Manage and generate your certificates below.

Create an additional certificate to use for this App ID.

Create an additional certificate to use for this App ID.

Now



About Creating a Certificate Signing Request (CSR)

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

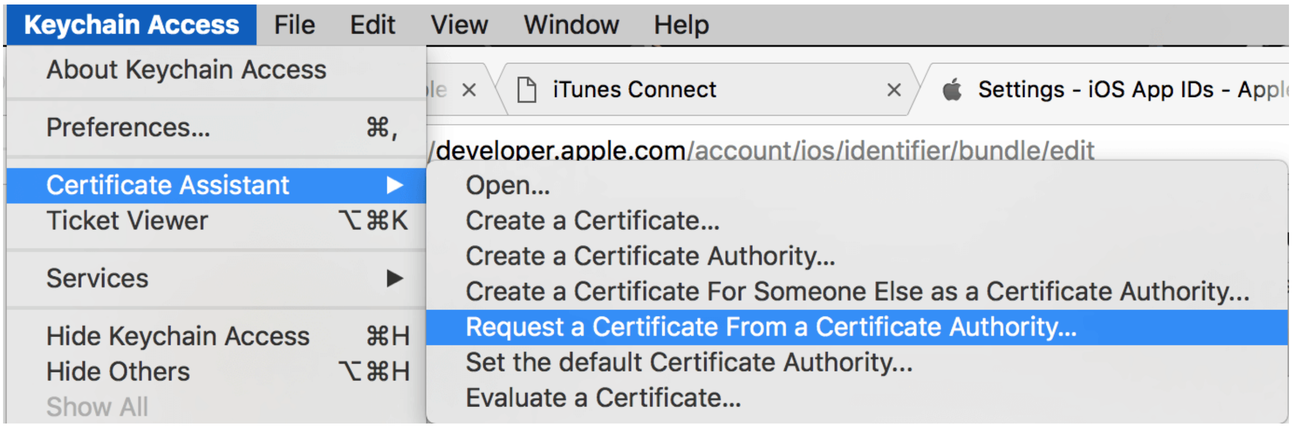
Create a CSR file.
In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

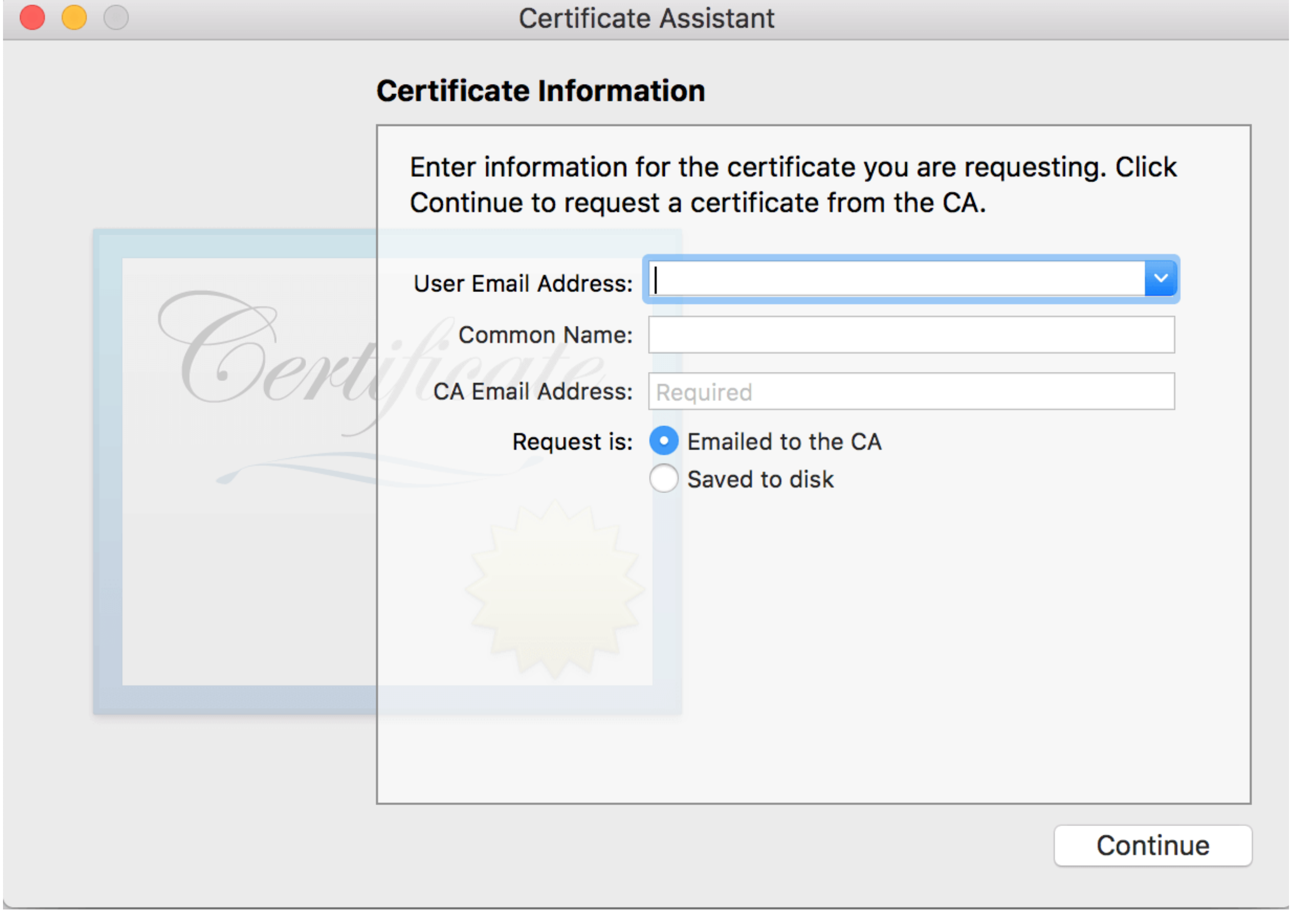
- In the Certificate Information window, enter the following information:
  - In the User Email Address field, enter your email address.
  - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
  - The CA Email Address field should be left empty.
  - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.

To generate a CSR file, press cmd + space and do a spotlight search for “Keychain Access.” Open

Keychain Access, and go up to the menu to select *Keychain Access > Certificate Assistant > Request a Certificate From a Certificate Authority…*


Keychain Access
File
Edit
View
Window
Help
About Keychain Access
Preferences... ⌘,
Certificate Assistant ▶
Ticket Viewer ⌥⌘K
Services ▶
Hide Keychain Access ⌘H
Hide Others ⌥⌘H
Show All
iTunes Connect
Settings - iOS App IDs - Appl
/developer.apple.com/account/ios/identifier/bundle/edit
Open...
Create a Certificate...
Create a Certificate Authority...
Create a Certificate For Someone Else as a Certificate Authority...
Request a Certificate From a Certificate Authority...
Set the default Certificate Authority...
Evaluate a Certificate...

A “Certificate Assistant” pop up should appear.ill in your email address and name. Choose “Saved to disk” and press Continue. Then save your CSR somewhere on your hard drive.


Certificate Assistant

Certificate Information

Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.

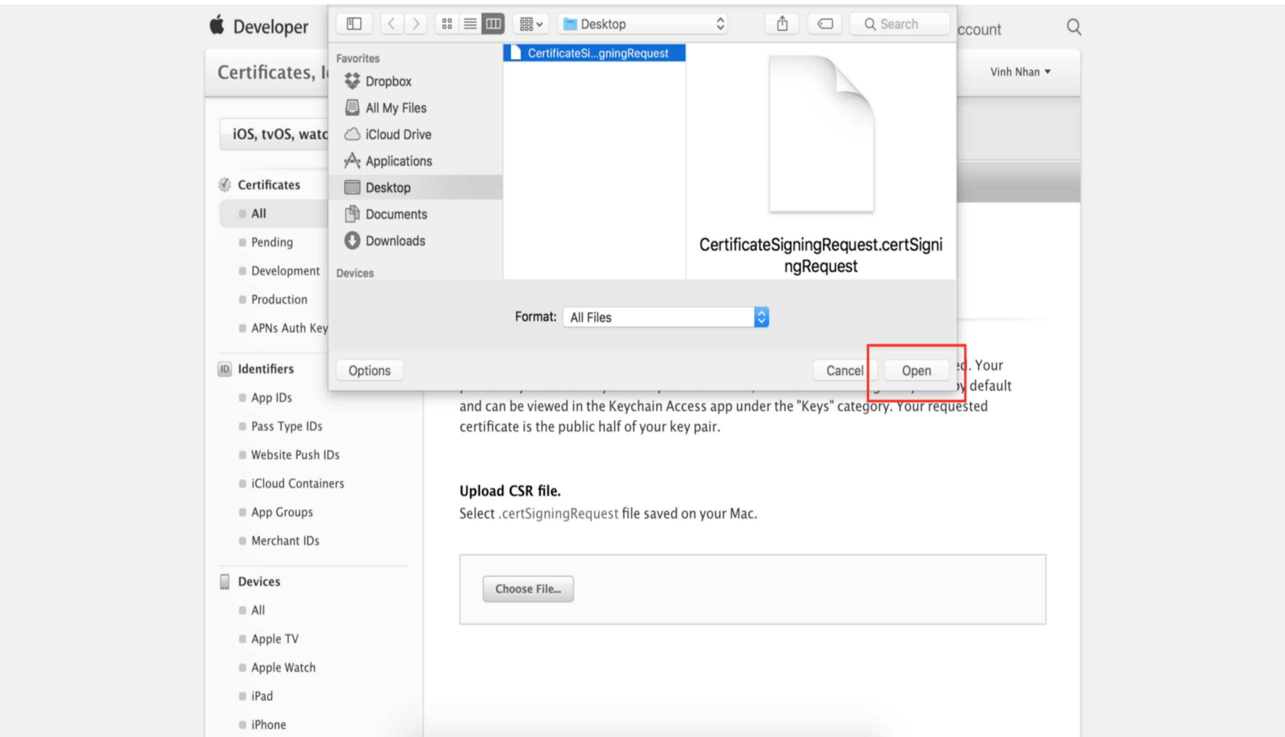
User Email Address:

Common Name:

CA Email Address: Required

Request is: ● Emailed to the CA
○ Saved to disk

Now that we have our CSR generated, it is ready to go back to the “Add iOS Certificate” page.Scroll down. Press continue, and then click “Choose file…” Select the CSR file you just saved on your hard drive.



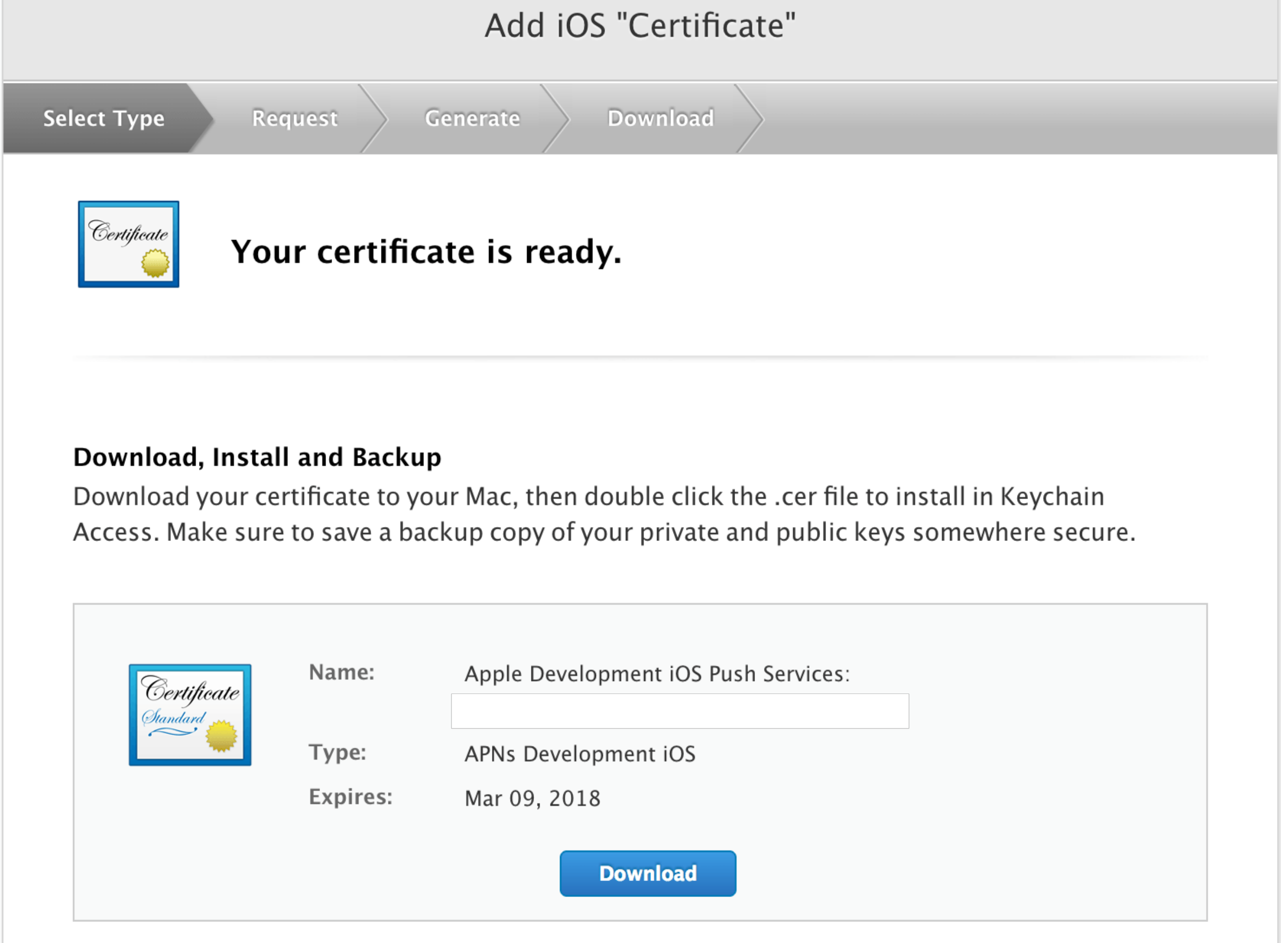
d. Your
y default
and can be viewed in the Keychain Access app under the "Keys" category. Your requested certificate is the public half of your key pair.

Upload CSR file.

Select .certSigningRequest file saved on your Mac.

Choose File...

Next, click continue again. Then the web page should say “Your certificate is ready.”



Add iOS "Certificate"

Your certificate is ready.

Download, Install and Backup

Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.

Name: Apple Development iOS Push Services:

Type: APNs Development iOS

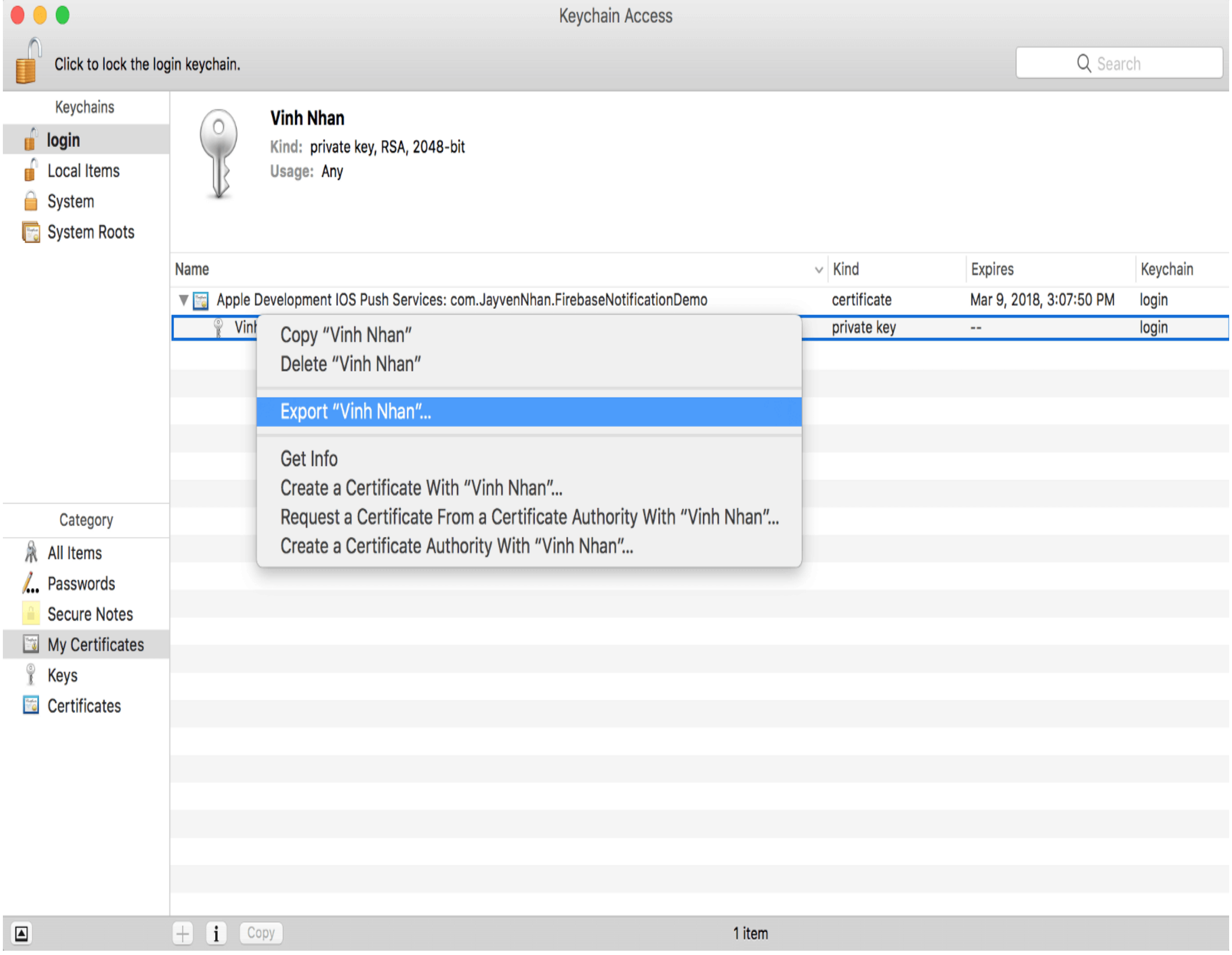
Expires: Mar 09, 2018

Now you can go ahead and click on the blue download button to download your certificate.Now that you have created the iOS certificate, we will then prepare the APNs (short for Apple Push

Notifications) certificate, which will be used later in the Firebase configuration. Open up *Finder* and locate the certificate you downloaded.
Double click the certificate file (e.g. aps_development.cer) to add the certificate into Keychain Access.

Now open up *Keychain Access*. Under the “My Certificates” category, you should see the certificate you just added.

Click the expand arrow to the left of the certificate’s name to reveal the private key option. Right click on the private key and press export.



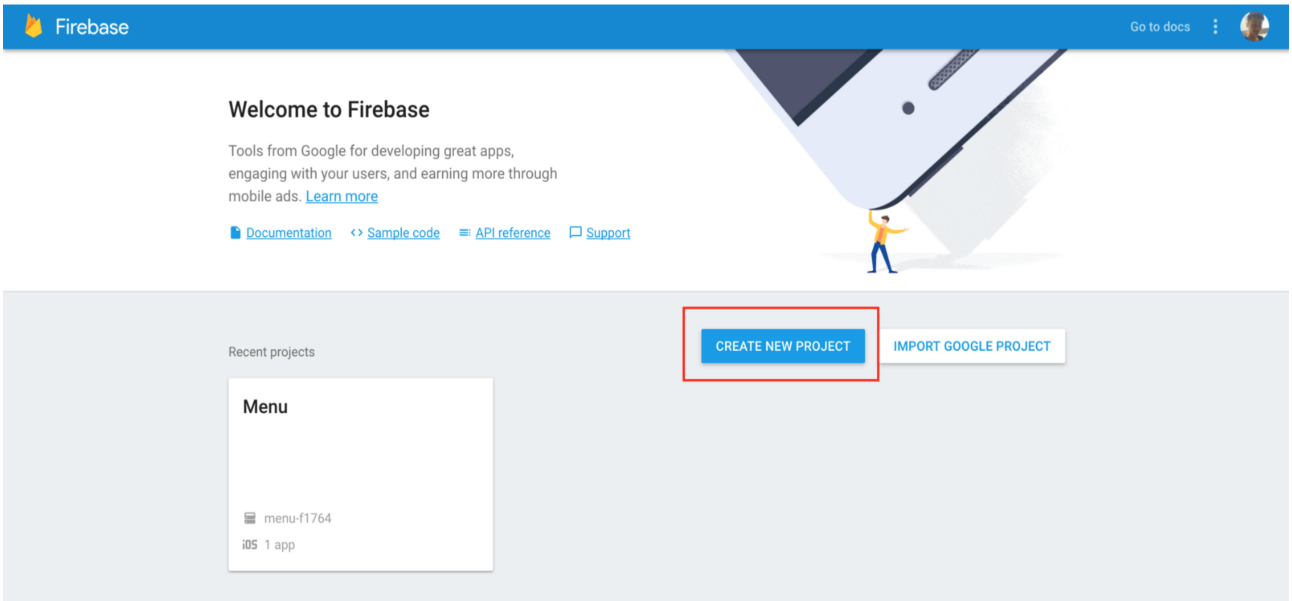
Vinh Nhan
Kind: private key, RSA, 2048-bit
Usage: Any

Name	Kind	Expires	Keychain
Apple Development IOS Push Services: com.JayvenNhan.FirebaseNotificationDemo	certificate	Mar 9, 2018, 3:07:50 PM	login
Vinh	private key	--	login

Once clicking Export, a pop up will prompt you to save your private key as a `.p12` file. Go ahead and click save.Then enter a password if you want to protect your exported certificate. Click OK to confirm.

Configuring Firebase for Push Notifications

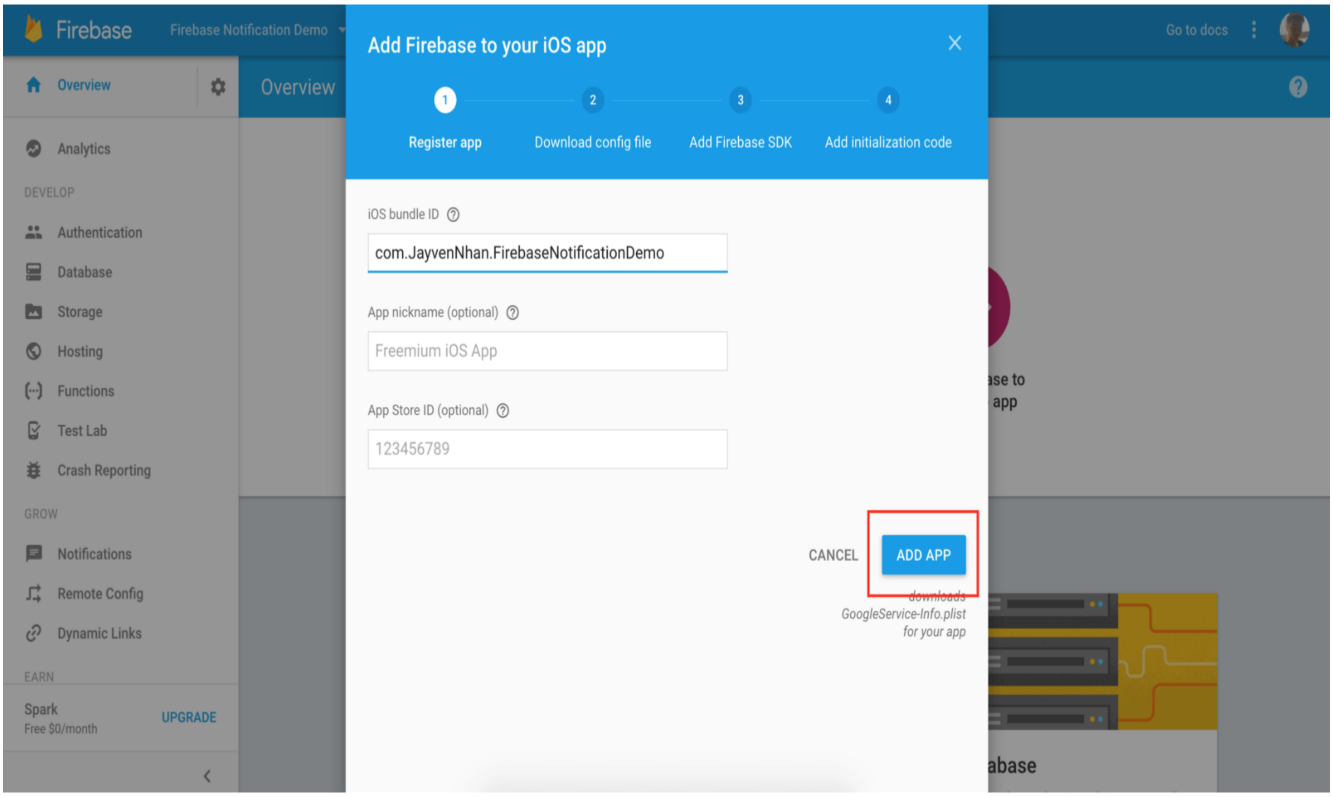
First, head over to Firebase console. Sign in with your Google account to enter the console. If you don’t know how to switch to the console, there is a button at the very top right that says “Go to console.” Go ahead and click on that.Once you are at the console, click on the “CREATE A NEW PROJECT” button.



Welcome to Firebase

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads. Learn more

Enter Project Name & Country .Click CREATE PROJECT.After that, you are redirected to the project overview page. Now click the “Add Firebase to your iOS” button. Enter your iOS bundle ID. Then click the “ADD APP” button.

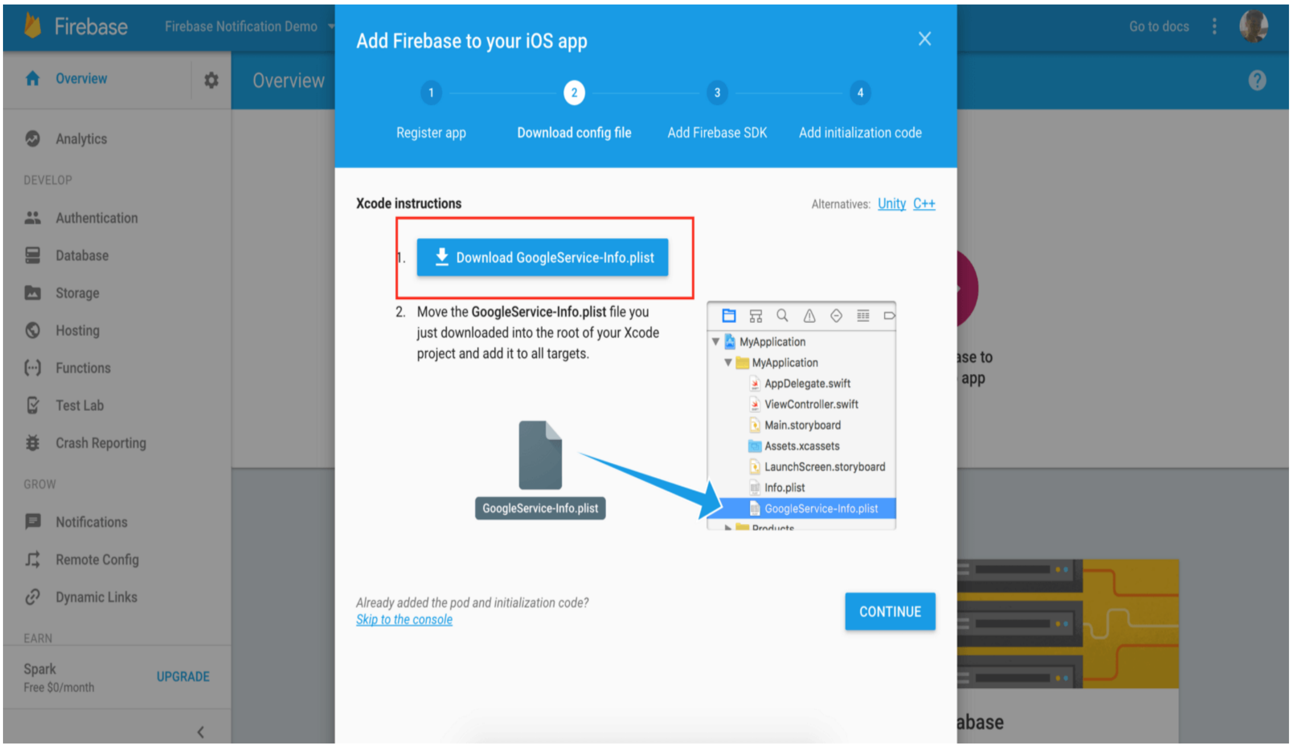


iOS bundle ID: com.JayvenNhan.FirebaseNotificationDemo

App nickname (optional): Freemium iOS App

App Store ID (optional): 123456789

Follow the on-screen instruction to download the `GoogleInfo.plist` file. We are going to need this file later.Click “continue” to go to the next step. You will see instructions showing you how to add the Firebase SDK into our project. I will walk you through how to configure the SDK later. For now, just ignore and click “Continue” to proceed.



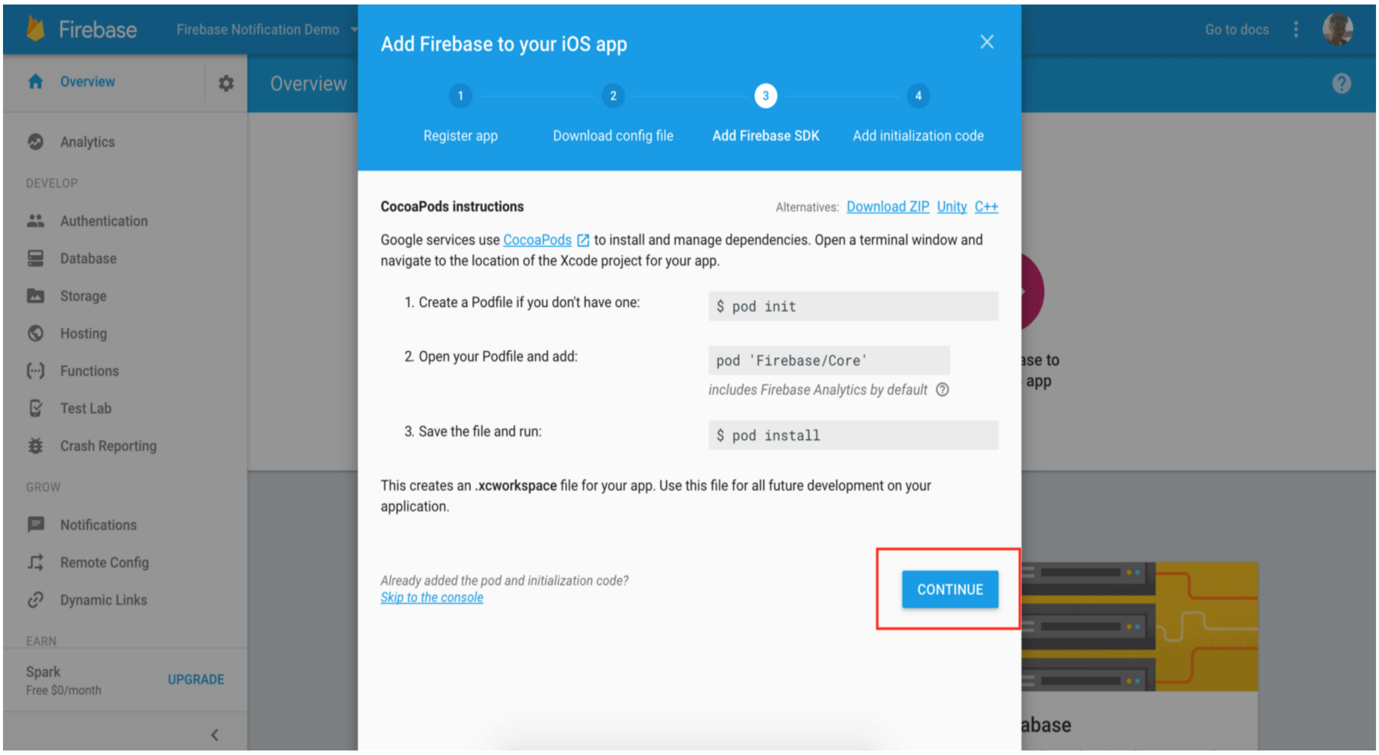
Add Firebase to your iOS app

Xcode instructions

Alternatives: Unity C++

1. Download GoogleService-Info.plist
2. Move the **GoogleService-Info.plist** file you just downloaded into the root of your Xcode project and add it to all targets.

Already added the pod and initialization code?
Skip to the console



Add Firebase to your iOS app

CocoaPods instructions

Alternatives: Download ZIP Unity C++

Google services use CocoaPods to install and manage dependencies. Open a terminal window and navigate to the location of the Xcode project for your app.

1. Create a Podfile if you don't have one:

```
$ pod init
```

2. Open your Podfile and add:

```
pod 'Firebase/Core'
```

includes Firebase Analytics by default

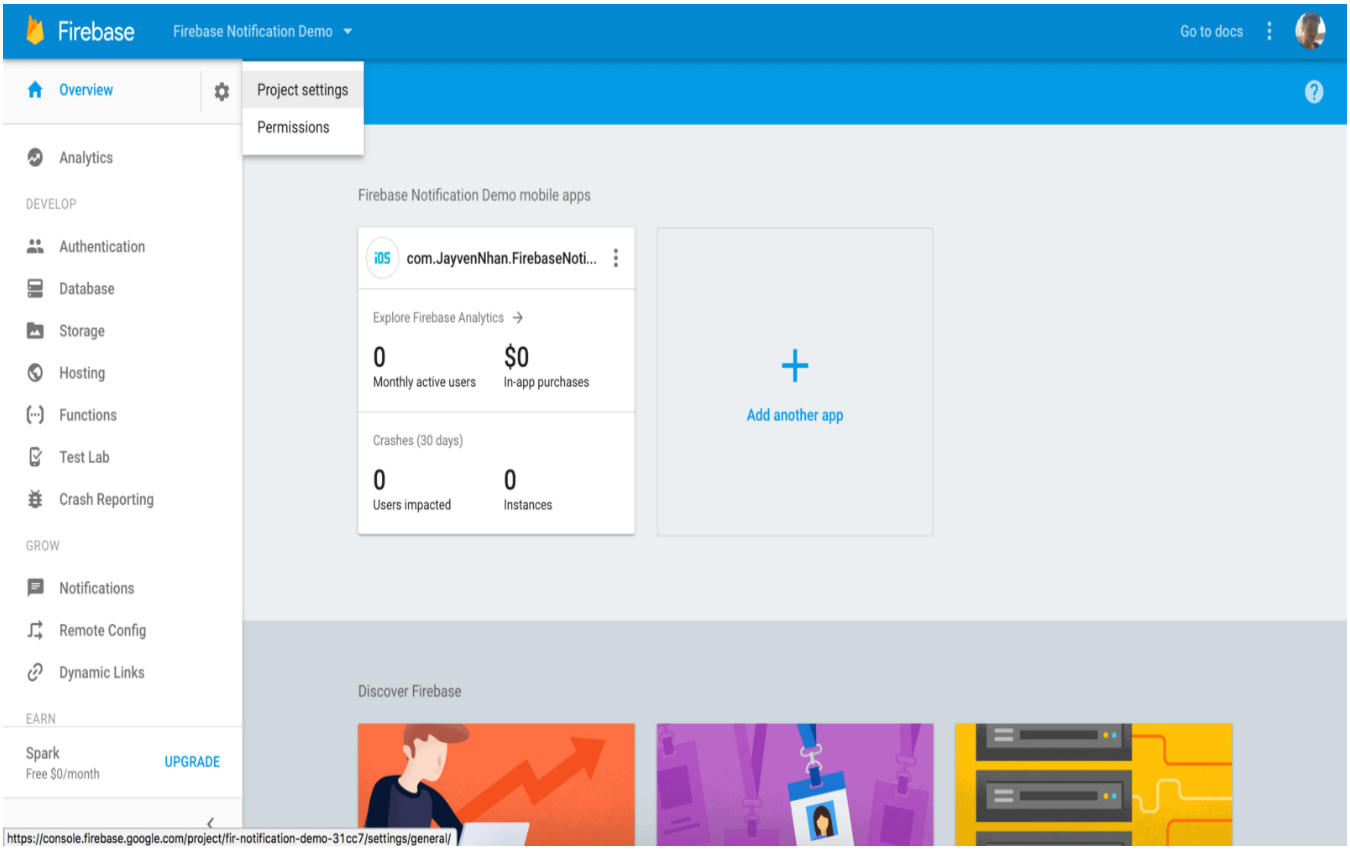
3. Save the file and run:

```
$ pod install
```

This creates an **.xcworkspace** file for your app. Use this file for all future development on your application.

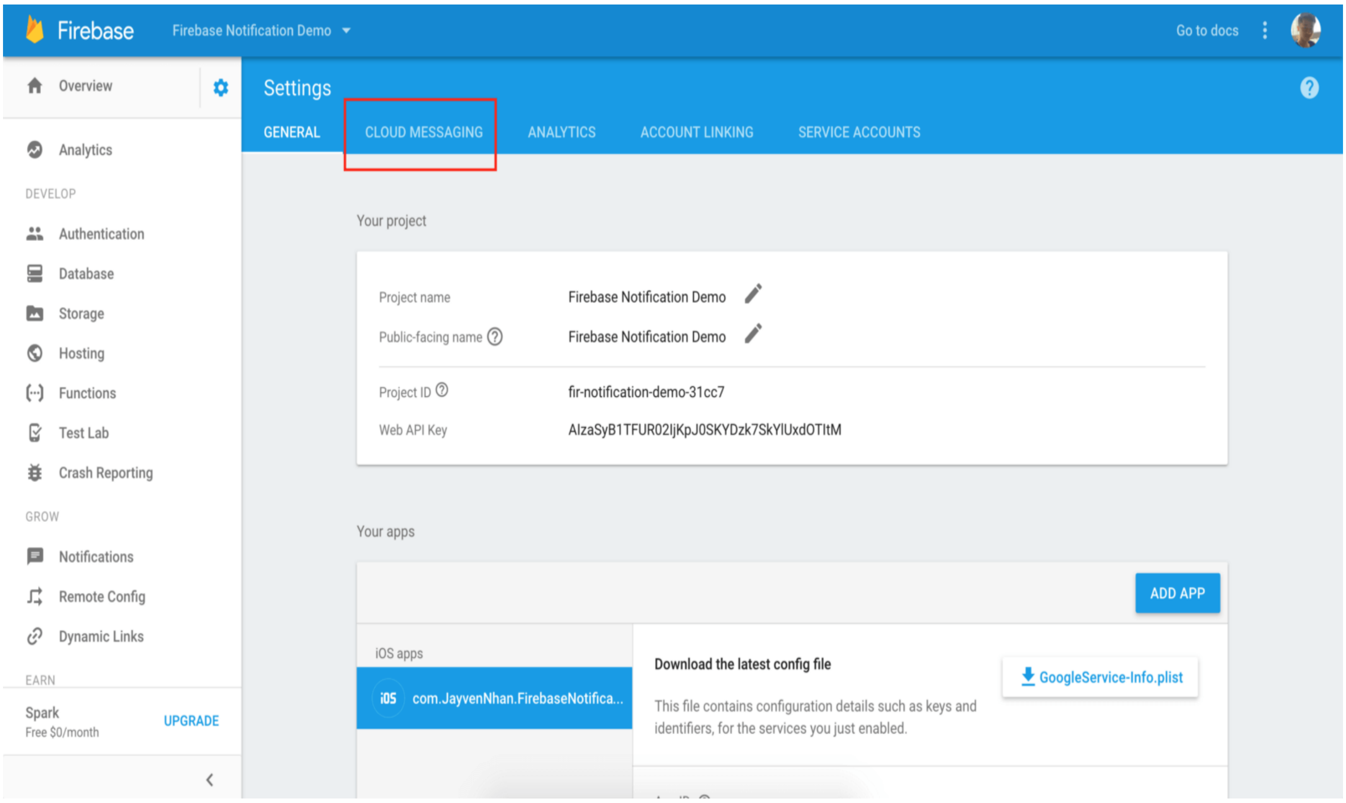
Already added the pod and initialization code?
Skip to the console

Finally click “Finish” to complete the configuration. You should then see your iOS app in the Firebase overview page.Look for the setting icon at the top right. Click the settings icon > Project settings.



https://console.firebase.google.com/project/fir-notification-demo-31cc7/settings/general/

Select the *Cloud Messaging* tab.



Project name Firebase Notification Demo

Public-facing name Firebase Notification Demo

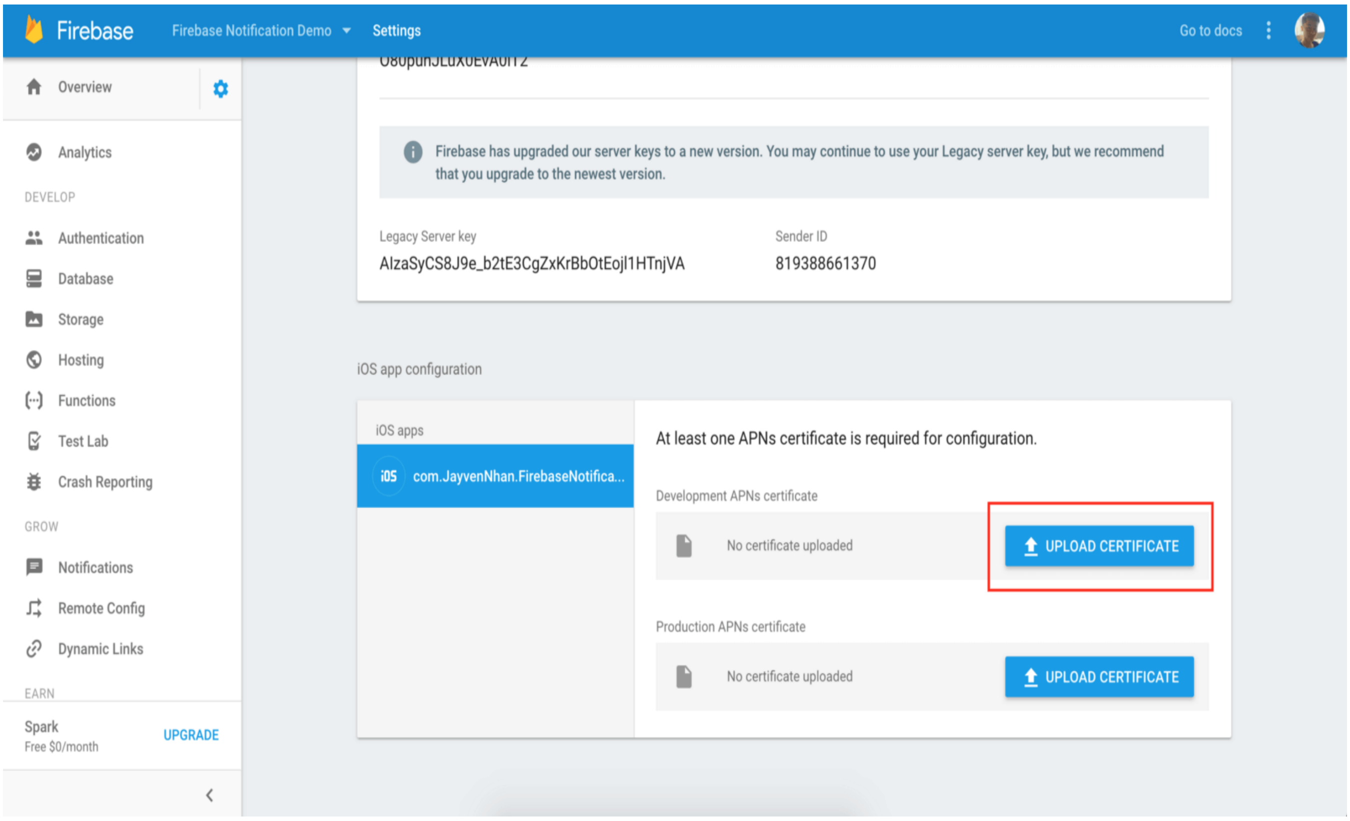
Project ID fir-notification-demo-31cc7

Web API Key AIzaSyB1TFUR02ljKpJ0SKYDzk7SkYlUxdOTItM

Download the latest config file

This file contains configuration details such as keys and identifiers, for the services you just enabled.

Scroll down, and click the “Upload Certificate” button.



U8upunJLuXUEVAUi1Z

Firebase has upgraded our server keys to a new version. You may continue to use your Legacy server key, but we recommend that you upgrade to the newest version.

Legacy Server key
AIzaSyCS8J9e_b2tE3CgZxKrBbOtEojl1HTnjVA

Sender ID
819388661370

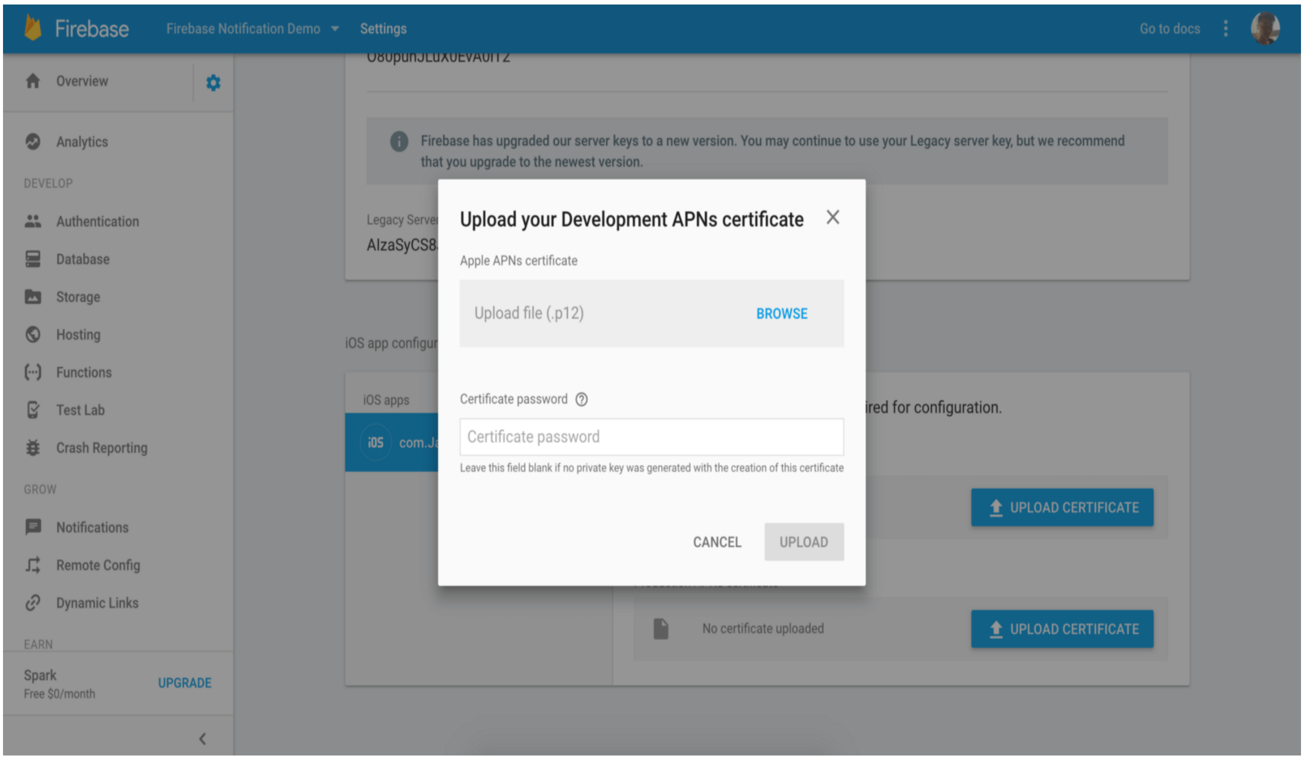
iOS app configuration

At least one APNs certificate is required for configuration.

Development APNs certificate
No certificate uploaded

Production APNs certificate
No certificate uploaded

Then a pop should appear to ask you for your Development APNS certificate.

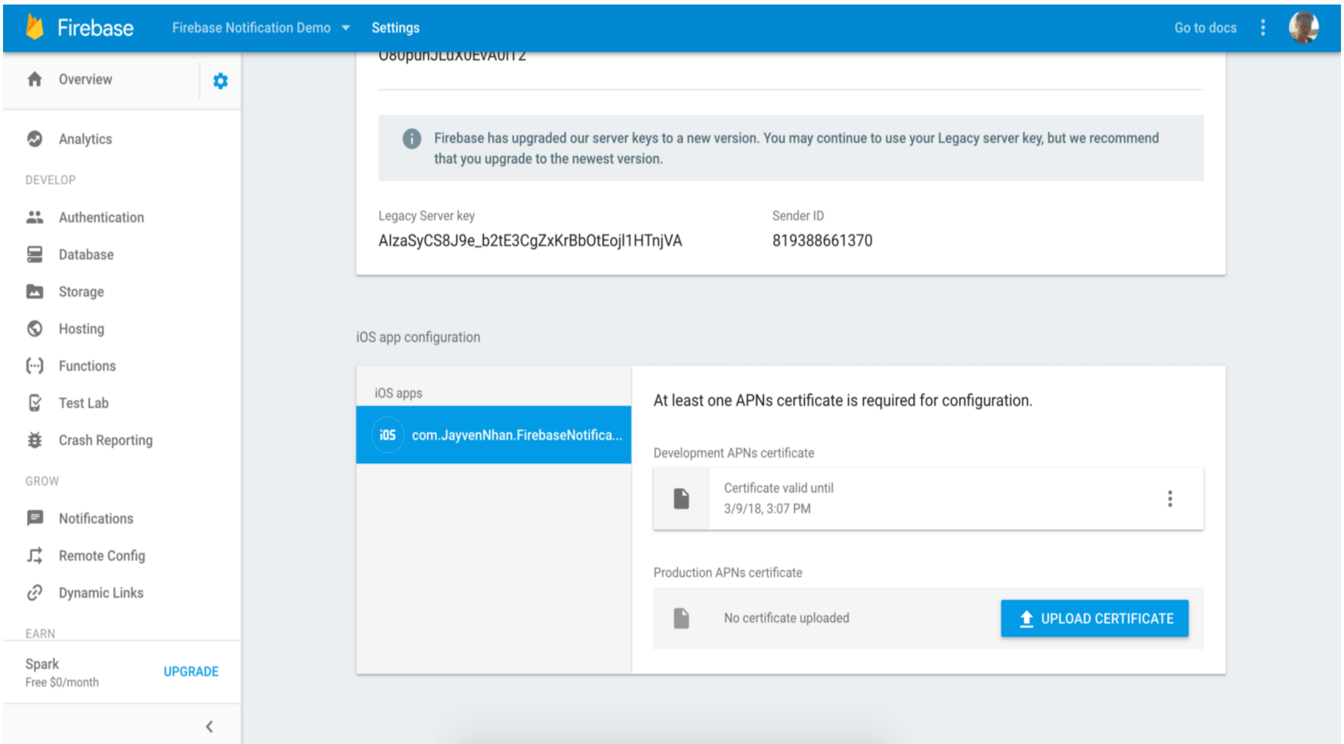


Upload your Development APNs certificate

Apple APNs certificate
Upload file (.p12)

Certificate password
Certificate password
Leave this field blank if no private key was generated with the creation of this certificate

Click browse and choose your APNs certificate (i.e. `.p12` file) that you prepared in the earlier section. If you configured the file with a password, enter certificate password, followed by clicking the Upload button.Now you should see a Development APNs certificate file in the Cloud Messaging setting.



O8upunJLuXUEVAUITZ

Firebase has upgraded our server keys to a new version. You may continue to use your Legacy server key, but we recommend that you upgrade to the newest version.

Legacy Server key: AIzaSyCS8J9e_b2tE3CgZxKrBbOtEojl1HTnjVA

Sender ID: 819388661370

iOS app configuration

iOS apps

iOS com.JayvenNhan.FirebaseNotifica...

At least one APNs certificate is required for configuration.

Development APNs certificate

Certificate valid until 3/9/18, 3:07 PM

Production APNs certificate

No certificate uploaded

Note:If you do decide to put your app onto the App Store, make sure that you have a Production APNs certificate file for Firebase as well.

Installing the Firebase SDK Using CocoaPods

Next, it is time for us to integrate the Firebase SDK into our Xcode project. The simplest way to do that is by using CocoaPods. First, close the Xcode project and then open *Terminal*. Change to the directory of your project and run the following command to initialize your Podfile:

```
pod init
```

Open the Podfile, add the following pods to your Podfile:

```
pod 'Firebase/Core'
pod 'Firebase/Messaging'
```

Save it. Your podfile should look something like this:

```
#Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'
  target 'FirebaseNotificationDemo' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!
  # Pods for FirebaseNotificationDemo
  pod 'Firebase/Core'
  pod 'Firebase/Messaging'
```

end

Go back to your terminal and run the following command to install the pods:
pod install
CocoaPods will then download the SDK for you and create a `.xcworkspace` file for you
With the SDK installed, it's time to add the `GoogleService-Info.plist` file that you have downloaded before. Drag that file from Finder to your project folder.

Enabling Push Notifications

The next step is to enable the Push Notifications capability of the demo project. Select the project in the project navigation, and then choose Target > Capabilities. Flip the switch of *Push Notifications* to `ON` to enable it.
`AppDelegate.swift` and import the following modules:

```
import UserNotifications
import Firebase
import FirebaseInstanceID
import FirebaseMessaging
```

Adopt the following protocols in your App Delegate class:

```
UNUserNotificationCenterDelegate, FIRMessagingDelegate
```

Next, implement the `applicationReceivedRemoteMessage` required method of the `FIRMessagingDelegate` protocol like this:

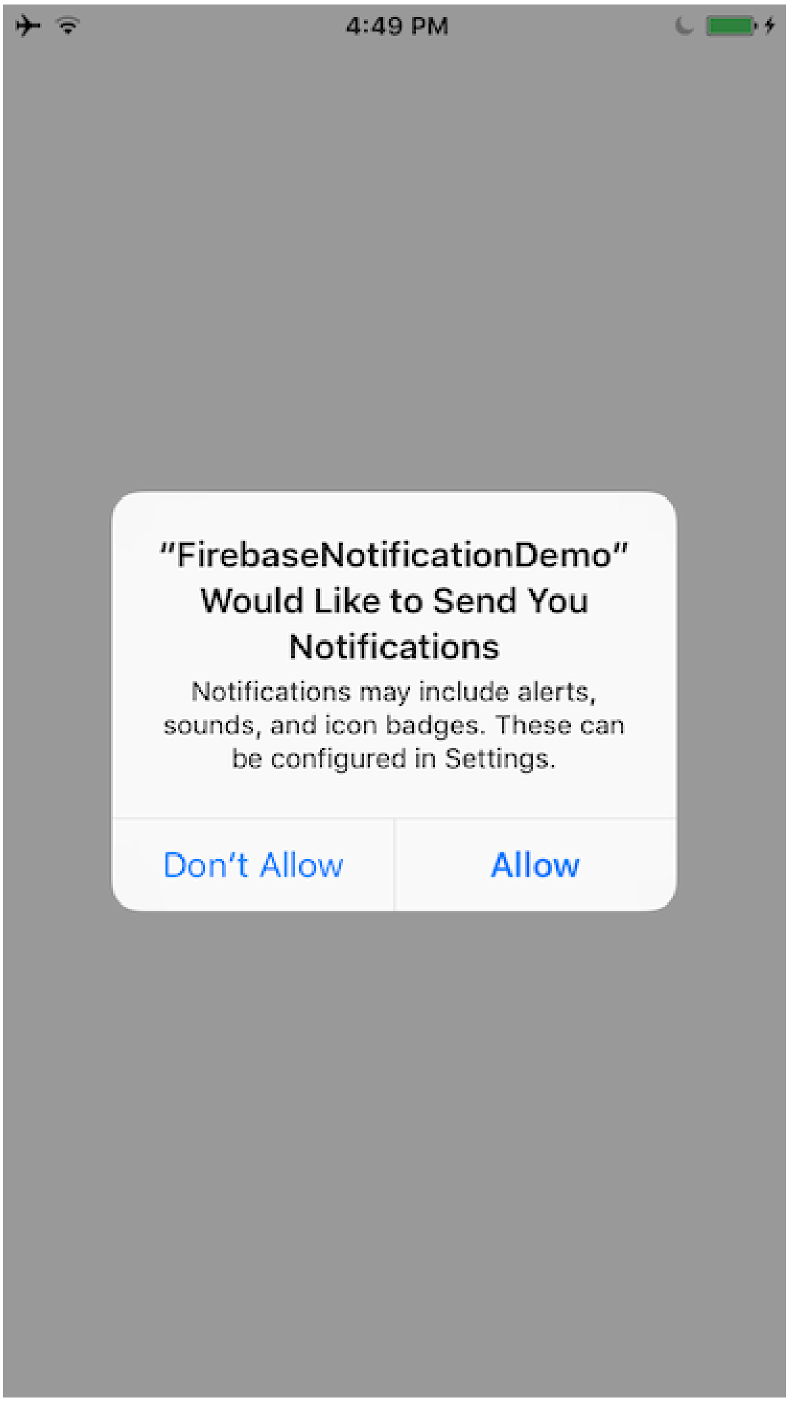
```
 func applicationReceivedRemoteMessage(_ remoteMessage:
FIRMessagingRemoteMessage) {
        print(remoteMessage.appData)
    }
```

Now inside of your `application(_:didFinishLaunchingWithOptions:)` method, add the following code:

```
 if #available(iOS 10.0, *) {
            // For iOS 10 display notification (sent via APNS)
            UNUserNotificationCenter.current().delegate = self
            let authOptions: UNAuthorizationOptions =
[.alert, .badge, .sound]
            UNUserNotificationCenter.current().requestAuthorization(
                options: authOptions,
                completionHandler: {_, _ in })
            // For iOS 10 data message (sent via FCM
            FIRMessaging.messaging().remoteMessageDelegate = self
        } else {
            let settings: UIUserNotificationSettings =
                UIUserNotificationSettings(types:
[.alert, .badge, .sound], categories: nil)
            application.registerUserNotificationSettings(settings)
        }
        application.registerForRemoteNotifications()
        FIRApp.configure()
```

This sets up push notification settings based on your iOS version. The last line `FIRApp.configure()` is to initialize and configure Firebase.

Once you have your App Delegate set up, you can run your Xcode project on your device. Make sure you tap *Allow* when your app asks you for notifications permission.

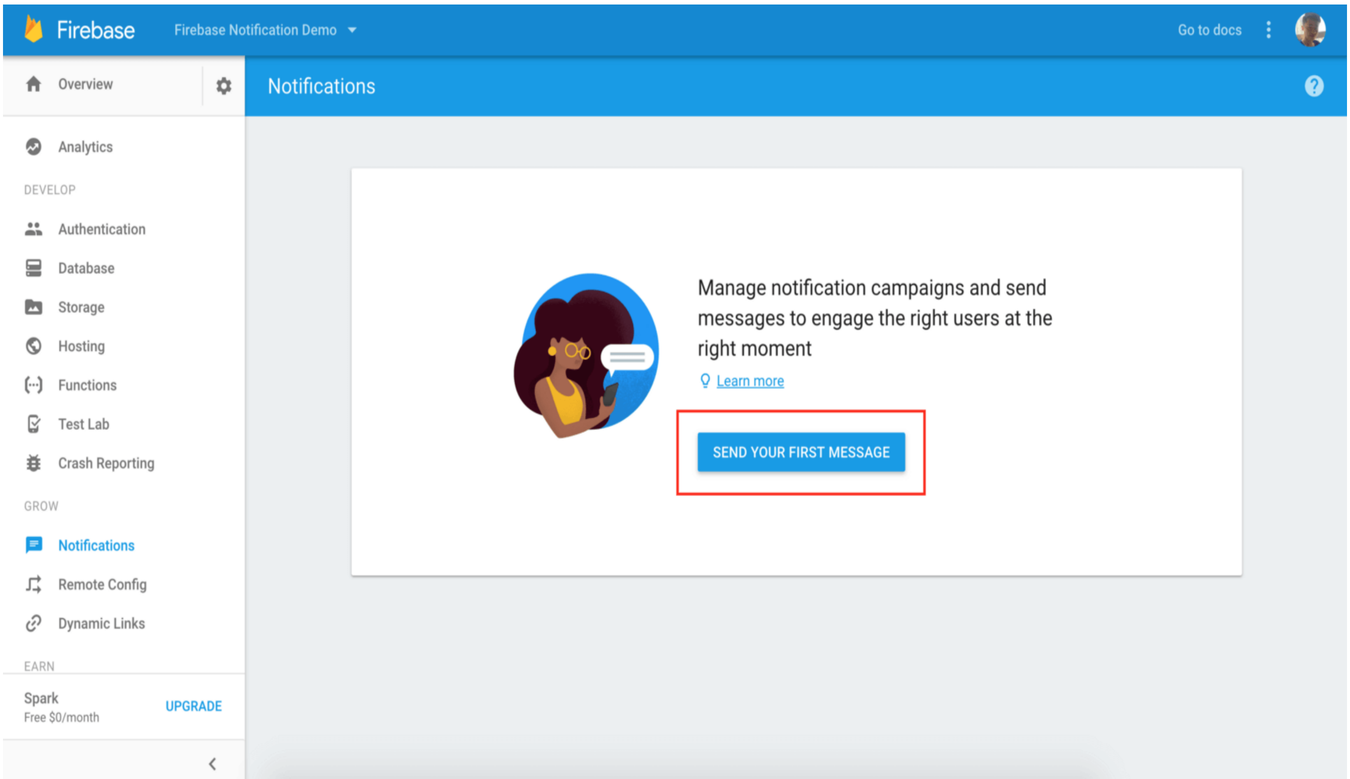


"FirebaseNotificationDemo" Would Like to Send You Notifications

Notifications may include alerts, sounds, and icon badges. These can be configured in Settings.

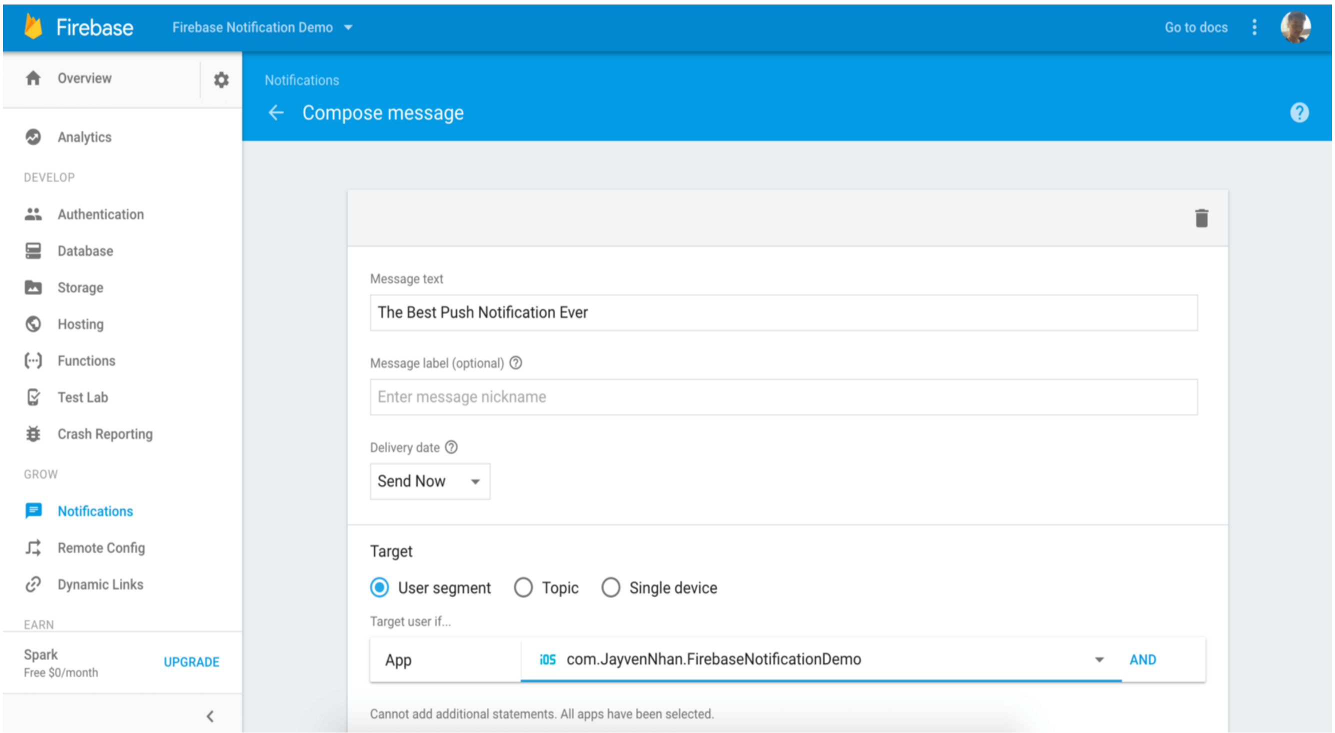
Testing Firebase Notifications

Go back to the Firebase Console, and then select your Firebase project. Look to the left side bar and you should see the Notifications option. Click the button. Once you see the Notification page, click the “SEND YOUR FIRST MESSAGE” button.



Manage notification campaigns and send messages to engage the right users at the right moment

Fill in the “Message text” field with a message you would like your users to see. Then select your app under the Target section.Scroll down and press the “SEND MESSAGE” button.



Message text: The Best Push Notification Ever

Message label (optional): Enter message nickname

Delivery date: Send Now

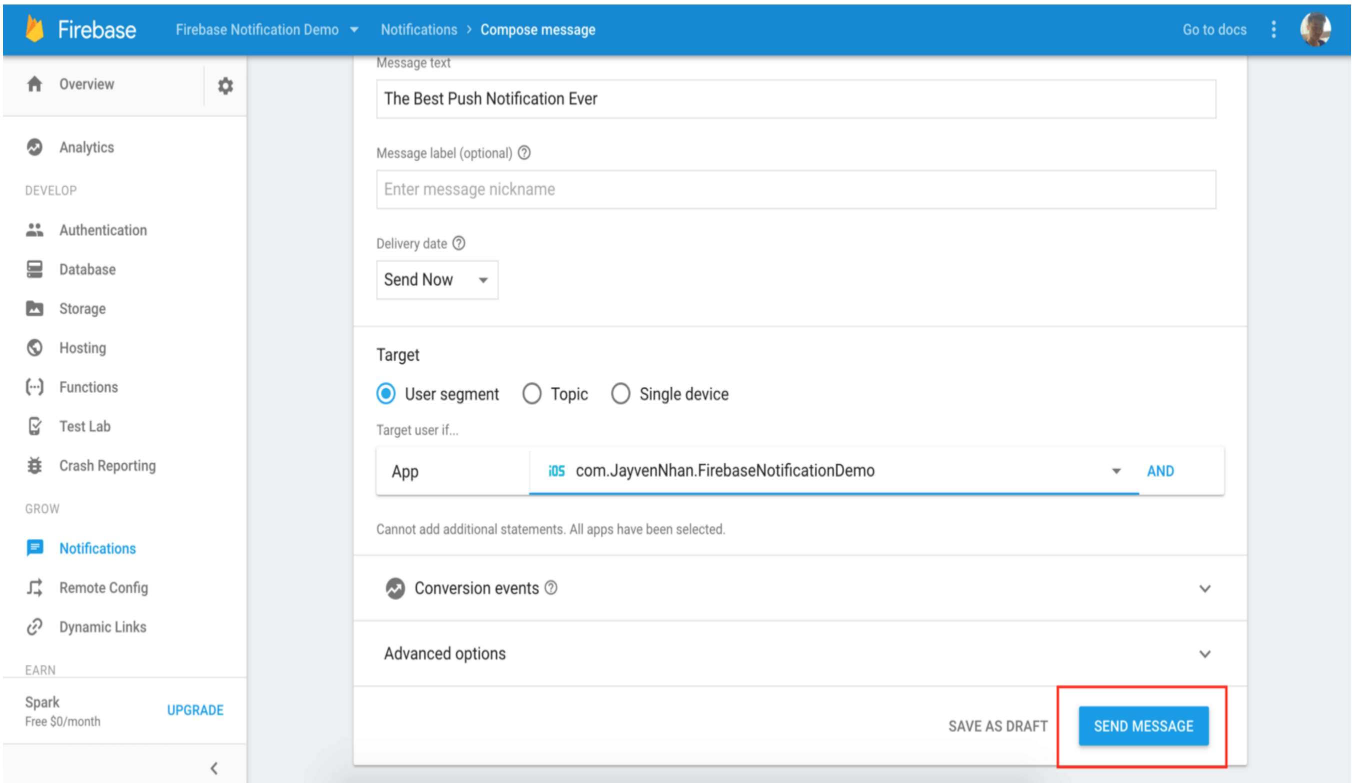
Target

● User segment ○ Topic ○ Single device

Target user if...

App | iOS com.JayvenNhan.FirebaseNotificationDemo | AND

Cannot add additional statements. All apps have been selected.



Message text: The Best Push Notification Ever

Message label (optional): Enter message nickname

Delivery date: Send Now

Target

● User segment ○ Topic ○ Single device

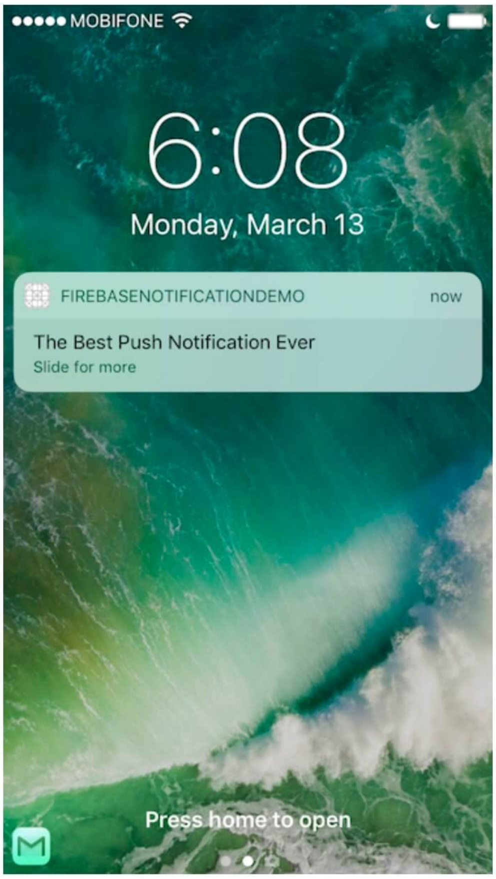
Target user if...

App | iOS com.JayvenNhan.FirebaseNotificationDemo | AND

Cannot add additional statements. All apps have been selected.

A pop up will ask you to review your notification message. Press the *SEND* button to send out your notification.

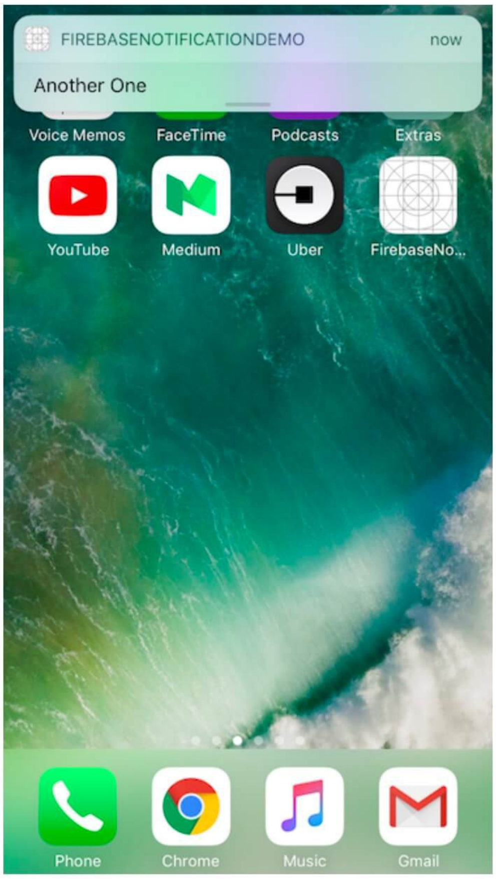
If you lock your device, you should be able to see the notification you pushed on your lock screen. You can then swipe from left to right on the notification to open the app. If you are on your home screen, you will a notification message drop down from the top of your screen. You can tap on the notification to open the app.



FIREBASENOTIFICATIONDEMO now

The Best Push Notification Ever

Slide for more



FIREBASENOTIFICATIONDEMO now

Another One