

One Signal Unity Integration

(Vivek Kumar Yadav)

Reference: -

<https://www.youtube.com/channel/UCKPUy0X6r8IN9vXSbXgEWJQ>

<https://www.youtube.com/watch?v=4H0tic78DuY>

(Thanks to DreamScape YouTube Channel)

LINK RELATED: -

<https://documentation.onesignal.com/docs/unity-sdk-setup>

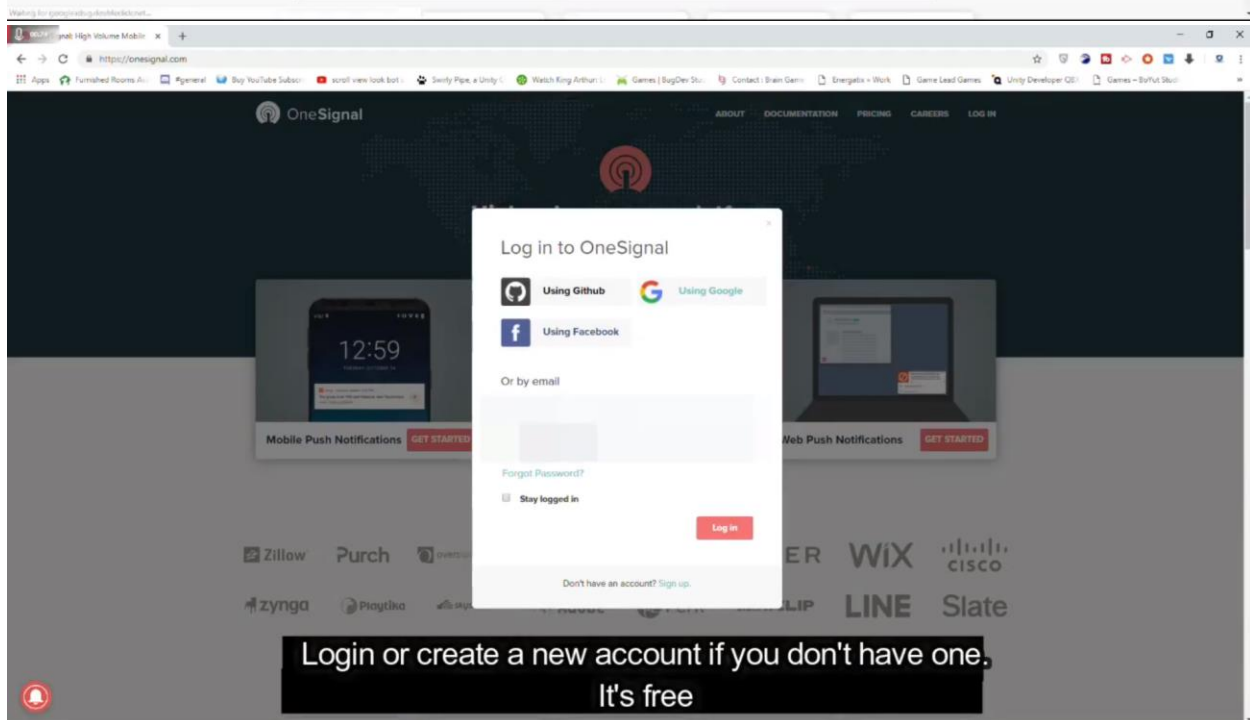
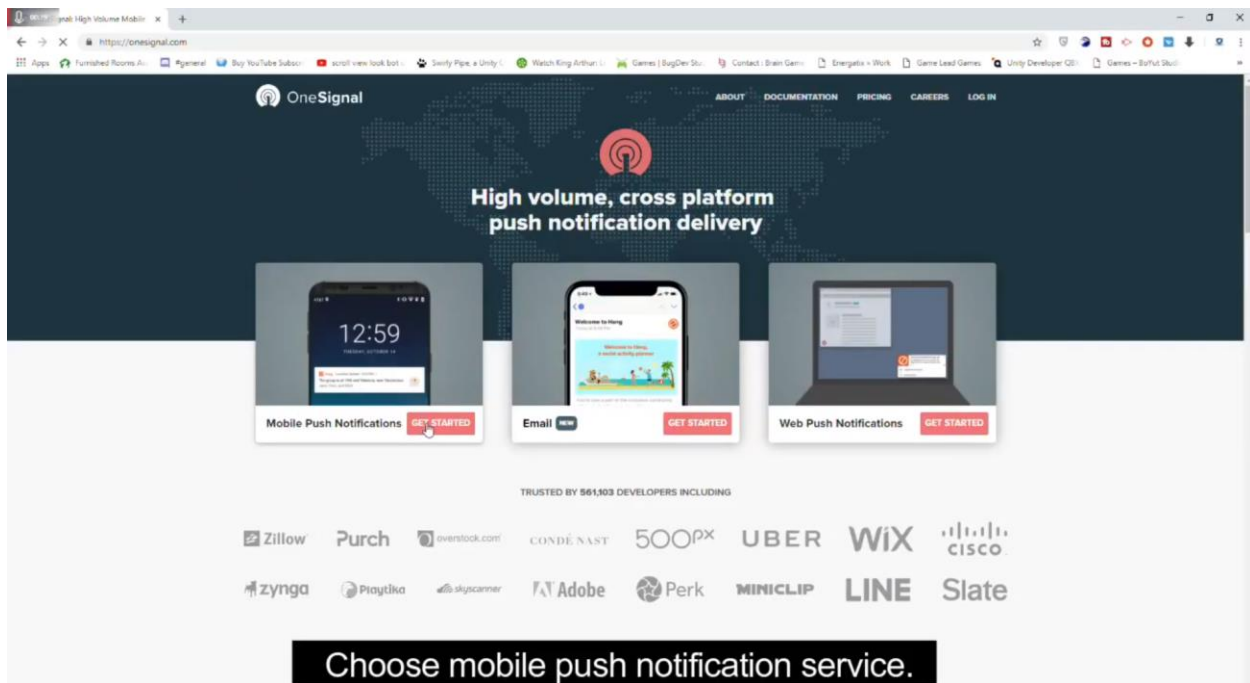
<https://documentation.onesignal.com/docs/troubleshooting-unity>

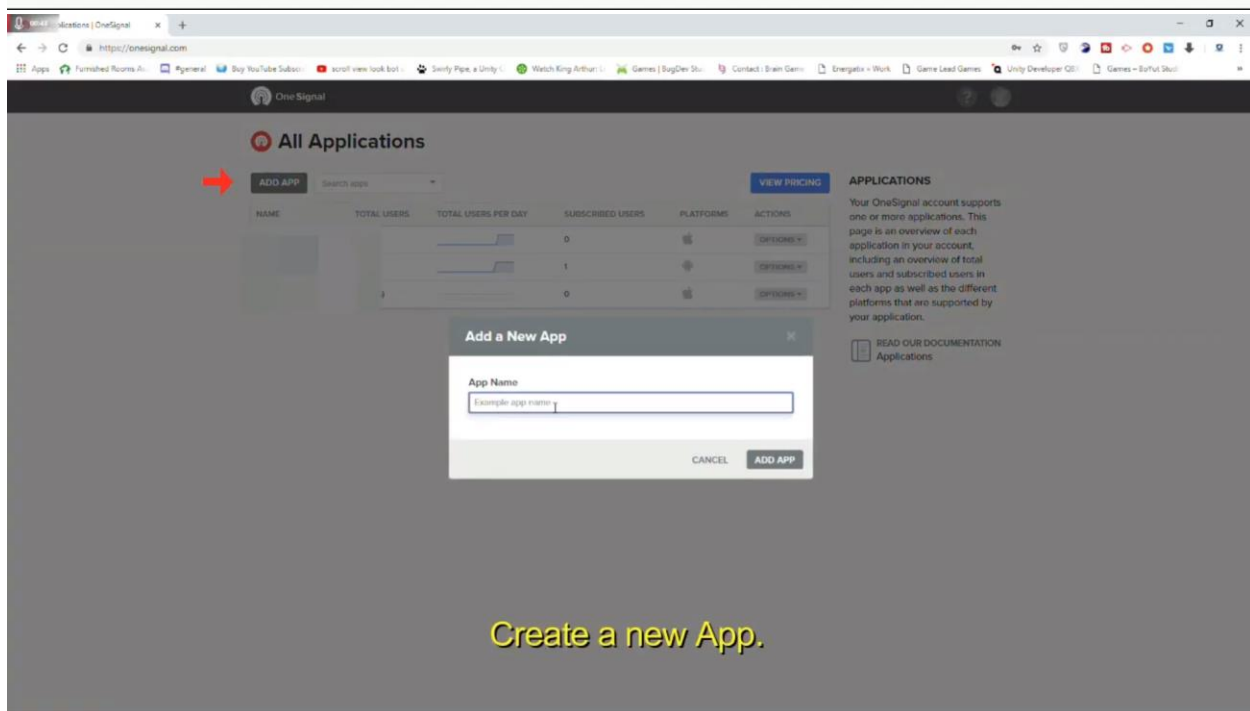
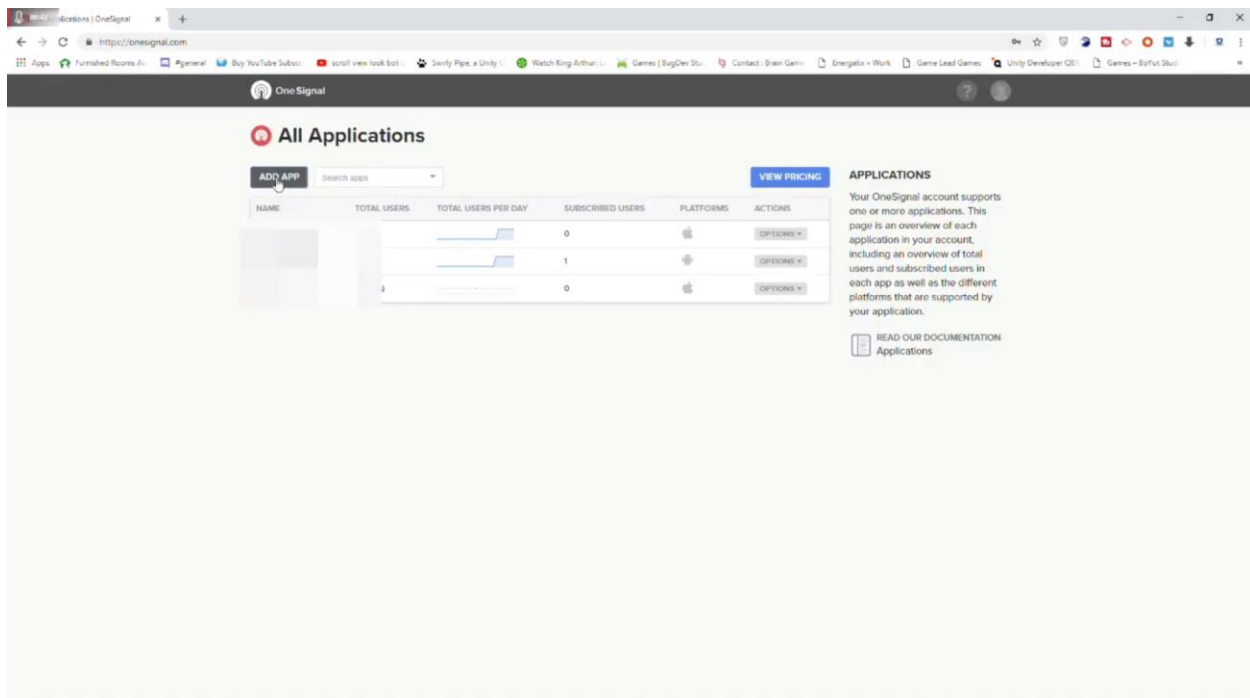
-----Pre-requisite-----

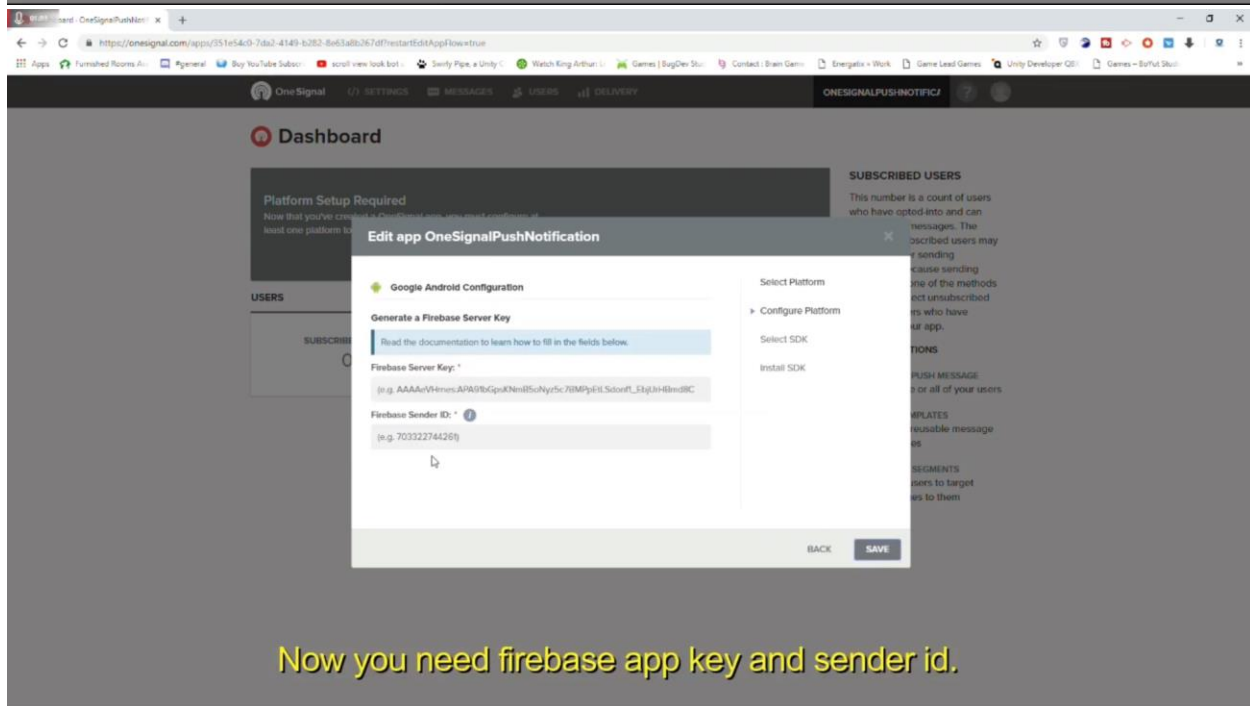
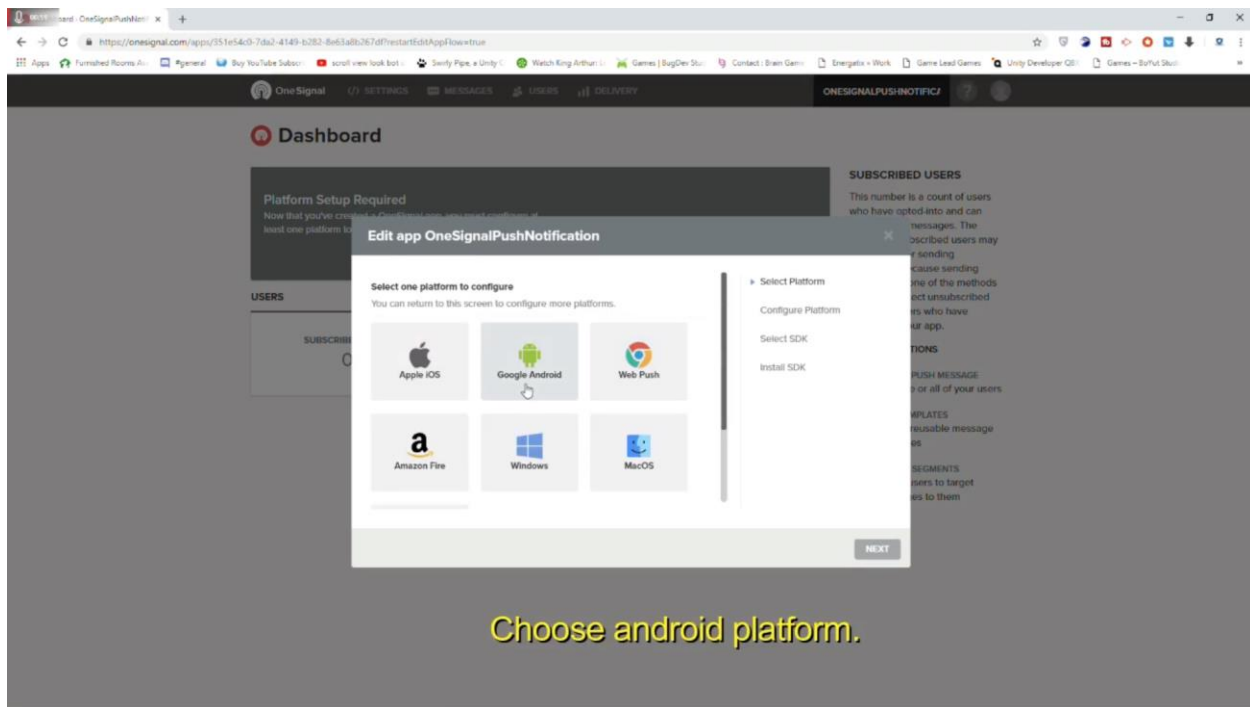
Go to the YouTube link and learn how to get your OneSignal app ID then continue with this documentation.

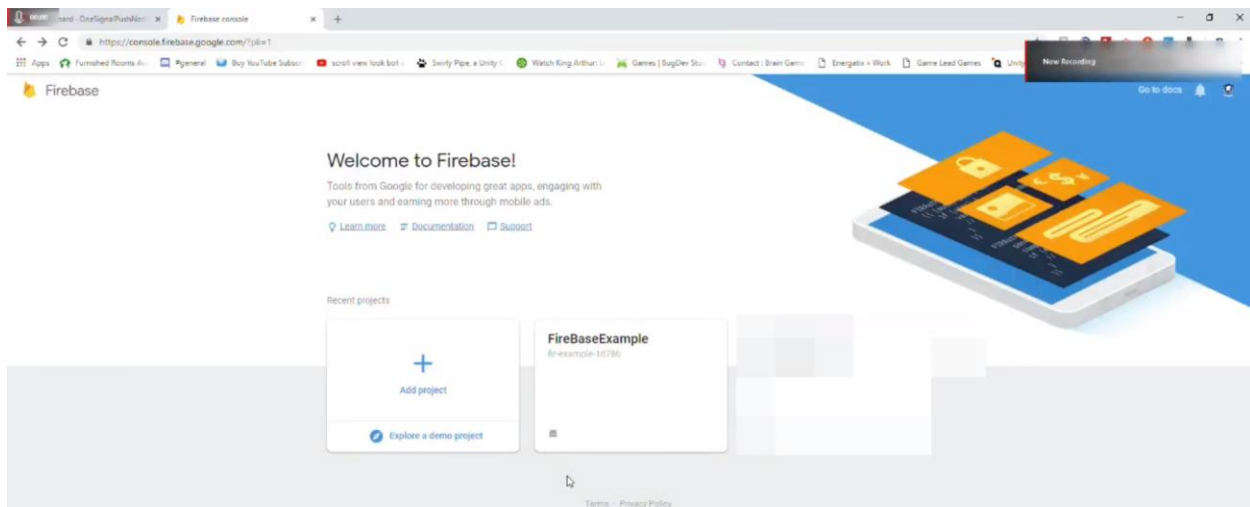
<https://www.youtube.com/watch?v=4H0tic78DuY>

OR JUST FOLLOW

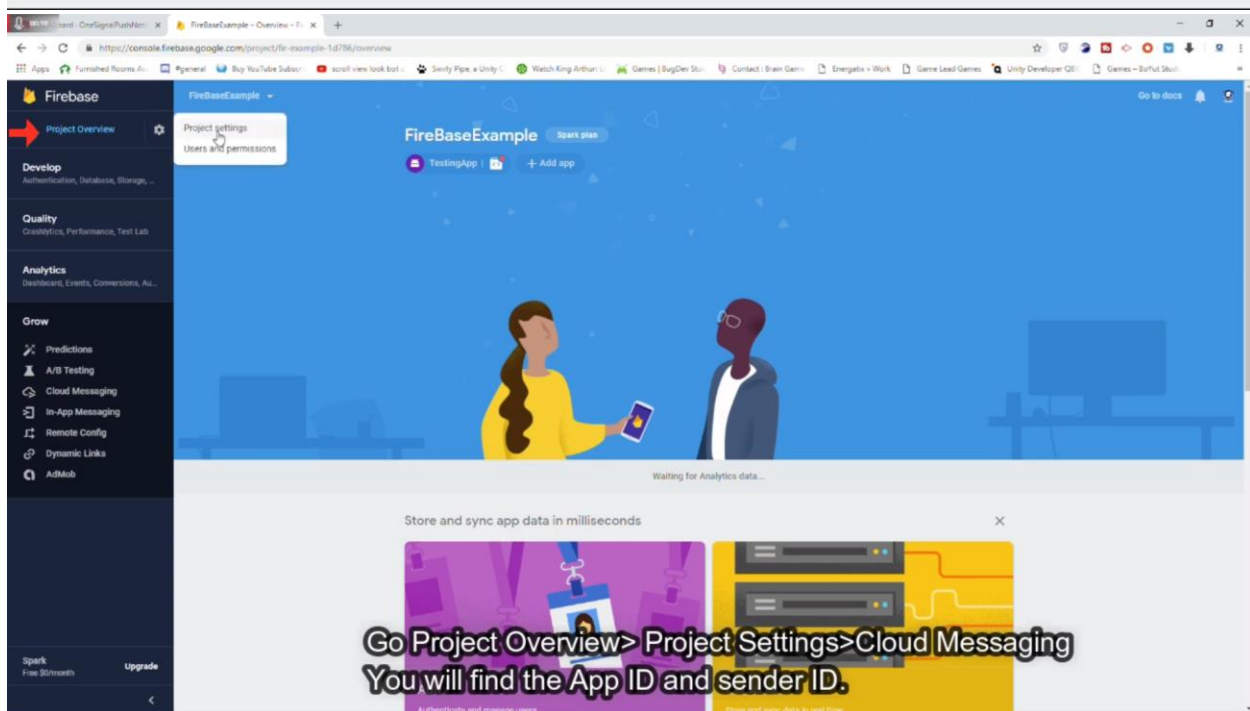








Go firebase.google.com. login and create an app.
Here i have already created one.



Go to Project Overview > Project Settings > Cloud Messaging
You will find the App ID and sender ID.

Go to Project Overview > Project Settings > Cloud Messaging
You will find the App ID and sender ID.

FireBase

Project Overview

Develop

Quality

Analytics

Grow

Predictions

A/B Testing

Cloud Messaging

In-App Messaging

Remote Config

Dynamic Links

AdMob

Spark

Free 30-month

Upgrade

Settings

General

Cloud Messaging

Integrations

Service accounts

Data privacy

Users and permissions

Project credentials

Add server key

Key	Token
Server key	
Legacy server key	
Sender ID	

IOS app configuration

You don't have an IOS app

Web configuration

Web Push certificates

Web Push certificates

Generate key pair

Copy the key ID and paste it in onesignal account.

OneSignal

Dashboard

Platform Setup Required

Now that you've created a OneSignal account, you need to configure your app to use OneSignal.

USERS

SUBSCRIBED

SUBSCRIBED USERS

This number is a count of users who have opted into and can receive messages. The subscribed users may be sending messages. The subscribed users may be sending messages. The subscribed users may be sending messages.

Edit app OneSignalPushNotification

Google Android Configuration

Generate a Firebase Server Key

Read the documentation to learn how to fill in the fields below.

Firebase Server Key: *

Firebase Sender ID: *

Select Platform

Configure Platform

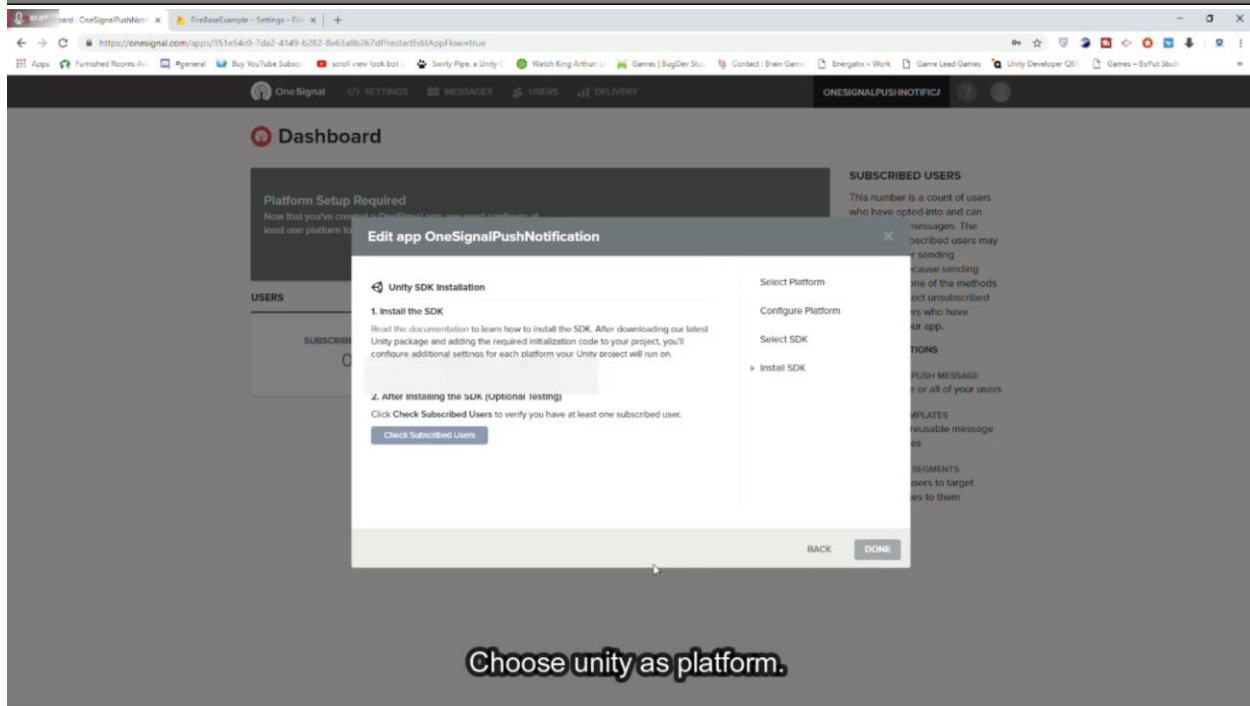
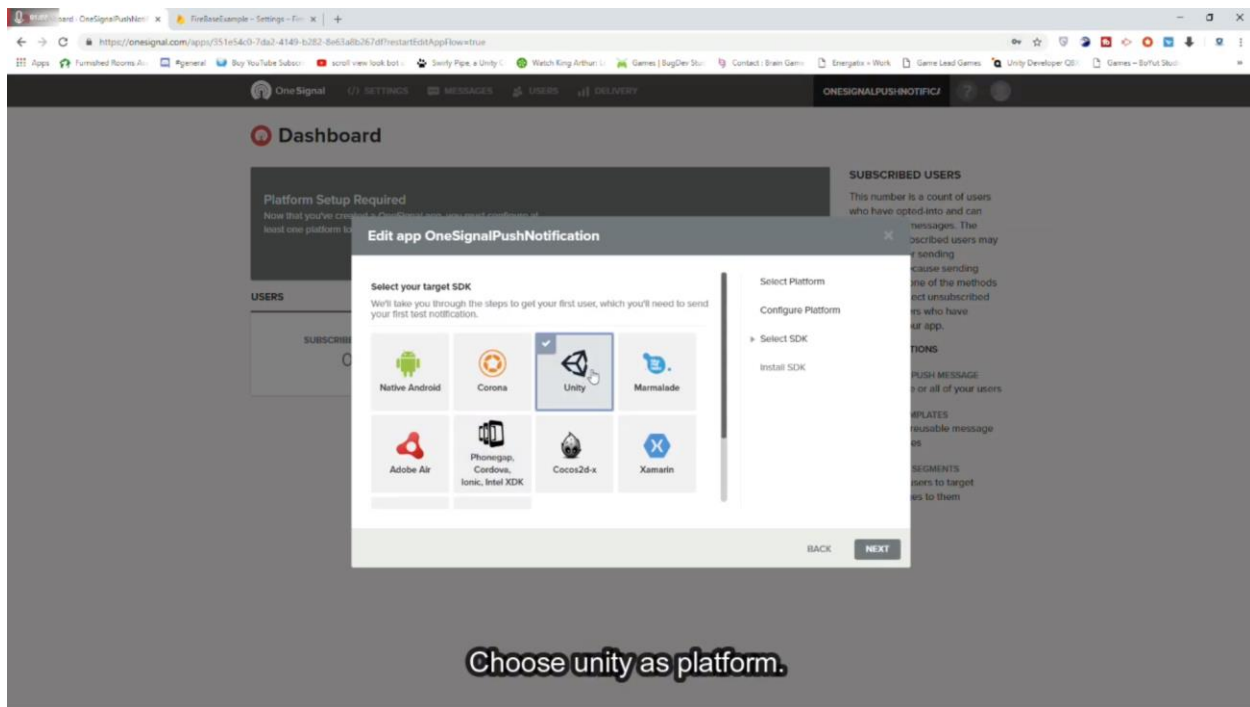
Select SDK

Install SDK

BACK

SAVE

Copy the key ID and paste it in onesignal account.



-----1st part-----

Unity SDK Setup

Instructions for adding the OneSignal Unity SDK to your Unity app for iOS, Windows Phone 8.1, Android, and derivatives like Amazon

Step 1 - Requirements

- A **OneSignal Account** if you do not already have one
- Your OneSignal App ID, available in **Keys & IDs**
- iOS - A Mac with a new version of Xcode (Required)

Step 2 - Generate Credentials

Before setting up the Unity SDK, you must generate the appropriate credentials for the platform(s) you are releasing on:

iOS - **Generate an iOS Push Certificate**

Android - **Generate a Google Firebase Server API Key**

Amazon Fire - **Generate an Amazon API Key**

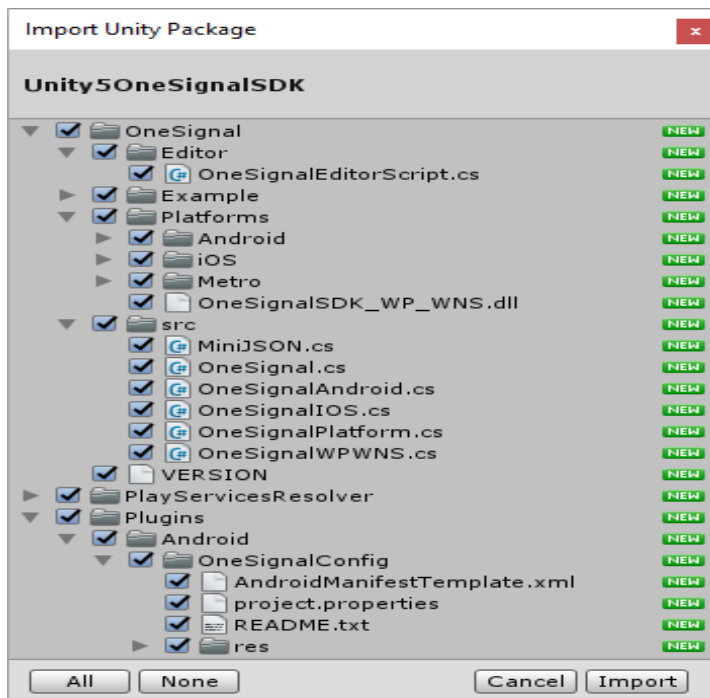
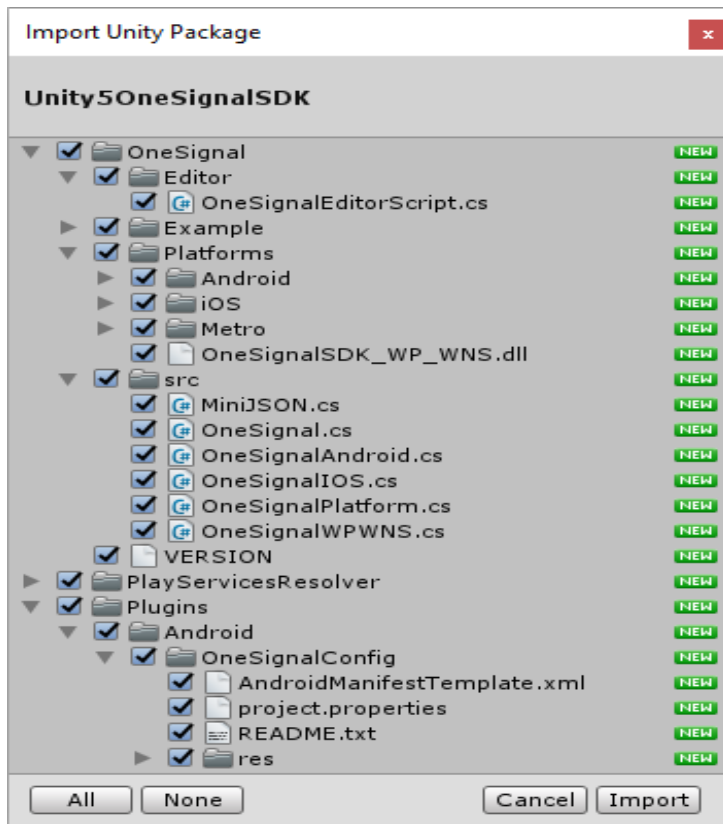
Step 3 - Import the OneSignal Unity Plugin

3.1 Download the **latest OneSignal unitypackage file**

3.2 Open your Unity project, then open the downloaded unitypackage file for the version of Unity you're using.

3.3 The following import package screen will come up. Press import.

3.3 The following import package screen will come up. Press import.



Step 4 - Initialize OneSignal in your Unity scene

Go to game controller script and change the app id with yours

```
C#

using System.Collections.Generic;

void Start () {

    // Uncomment this method to enable OneSignal Debugging log output

    // OneSignal.SetLogLevel(OneSignal.LOG_LEVEL.INFO, OneSignal.LOG_LEVEL.INFO);

    // Replace 'YOUR_ONESIGNAL_APP_ID' with your OneSignal App ID.
    OneSignal.StartInit("YOUR_ONESIGNAL_APP_ID")

        .HandleNotificationOpened(HandleNotificationOpened)

        .Settings(new Dictionary<string, bool>() {

            { OneSignal.kOSSettingsAutoPrompt, false },

            { OneSignal.kOSSettingsInAppLaunchURL, false } })

        .EndInit();

    OneSignal.inFocusDisplayType = OneSignal.OSInFocusDisplayOption.Notification;

    // The promptForPushNotifications function code will show the iOS push notification prompt. We
    // recommend removing the following code and instead using an In-App Message to prompt for
    // notification permission.

    OneSignal.PromptForPushNotificationsWithUserResponse(OneSignal_promptForPushNotifications
    Reponse);

    private void OneSignal_promptForPushNotificationsReponse(bool accepted) {

        Debug.Log("OneSignal_promptForPushNotificationsReponse: " + accepted);

    }
}
```

```
}
```

```
// Gets called when the player opens the notification.
```

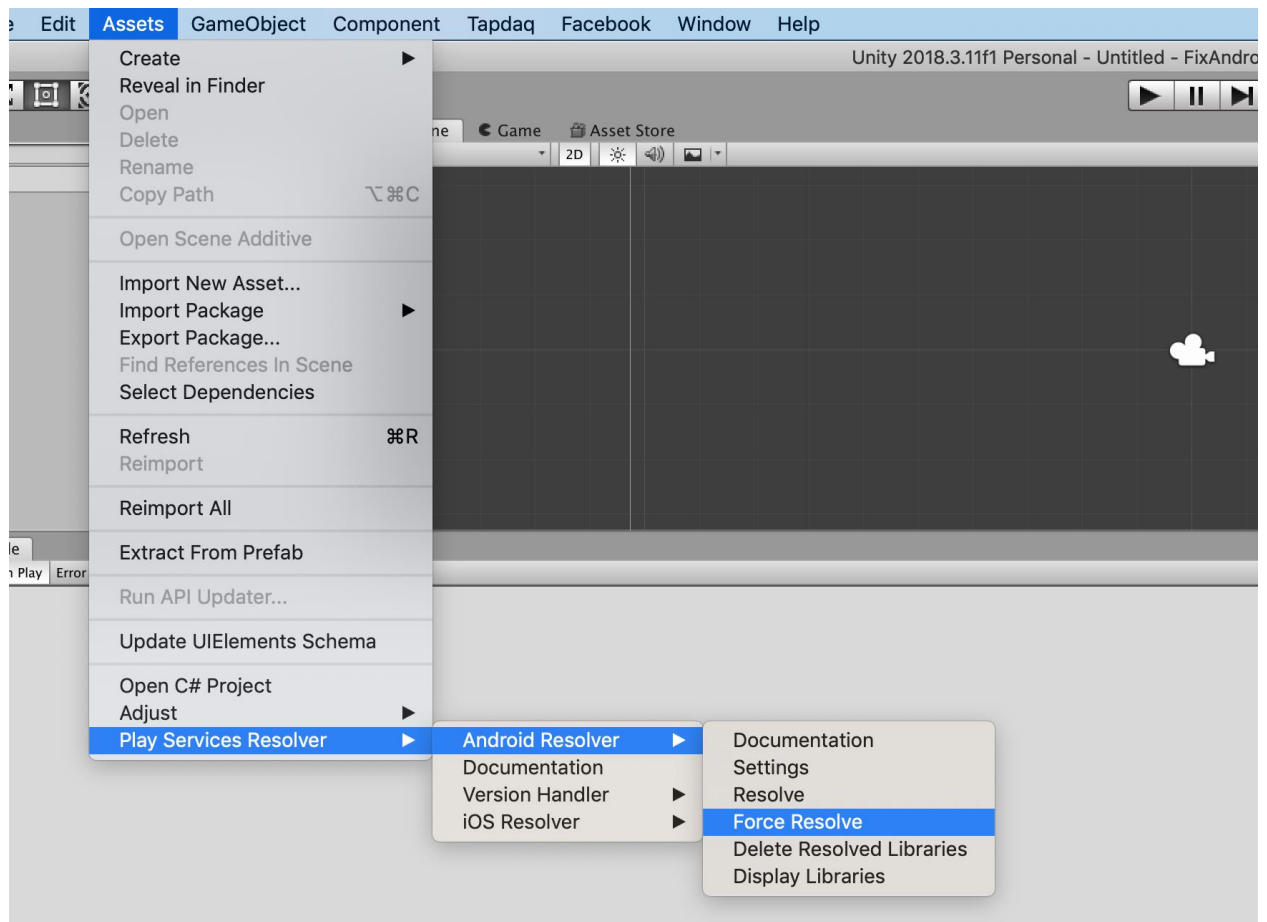
```
private static void HandleNotificationOpened(OSNotificationOpenedResult result) {
```

```
}
```

4.2 Replace "**YOUR_ONESIGNAL_APP_ID**" with your OneSignal app id.

Step 5 - Add Support for Android Notifications (Skip if your app is iOS only)

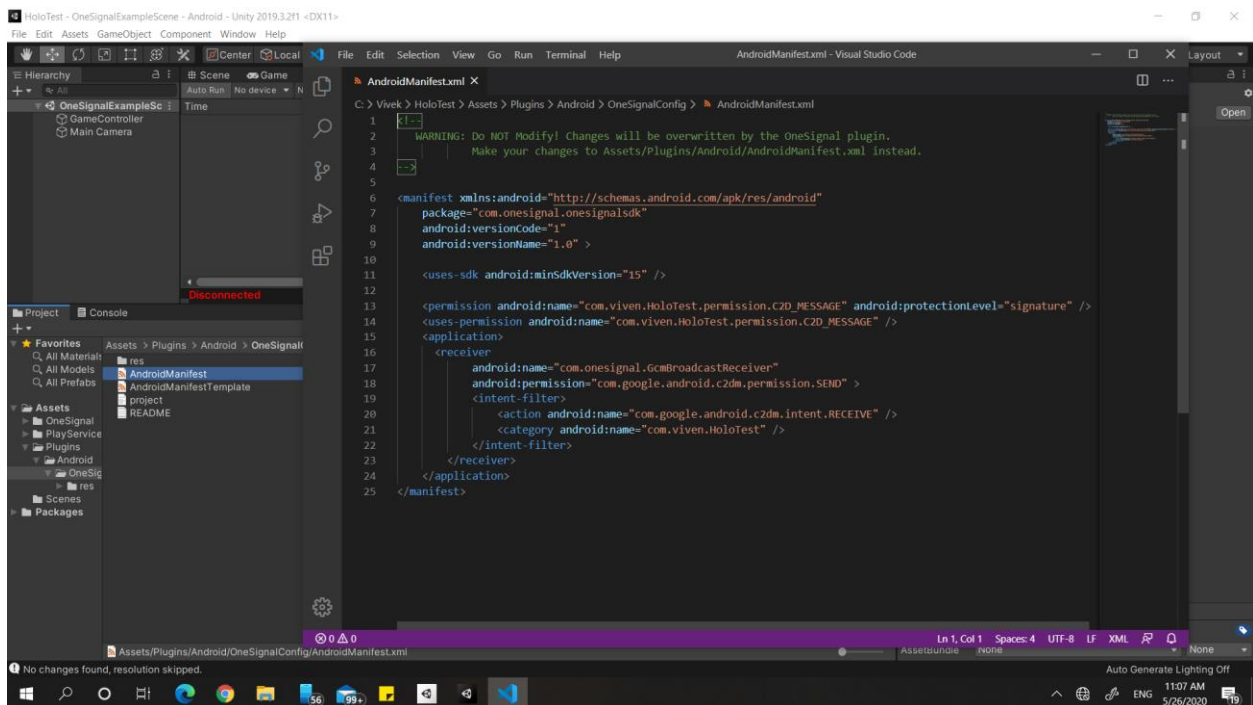
5.1 - Run Assets > Play Services Resolver > Android Resolver > Force Resolve from the menu bar.



2nd part

**Also Do rem after android resolver you will find
androidmanifest.xml in the folder
asset/Plugins/android/onesignalconfig**

And change the playerId or bundleId within.



-----3rd part-----

AndroidX Compatibility



Only follow the below instructions if you're using Unity 2018.4 or newer

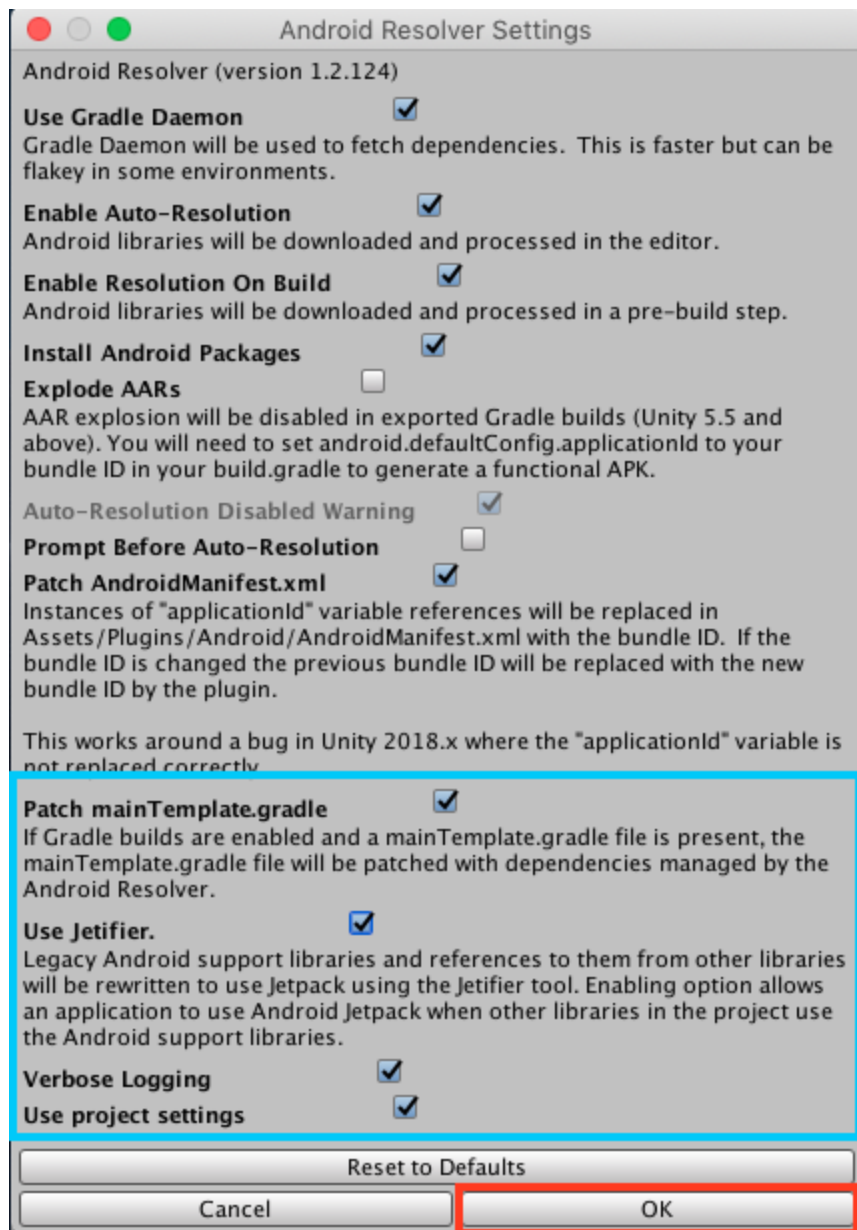
If you have any Plugins in your Unity project that are using AndroidX please follow these steps.

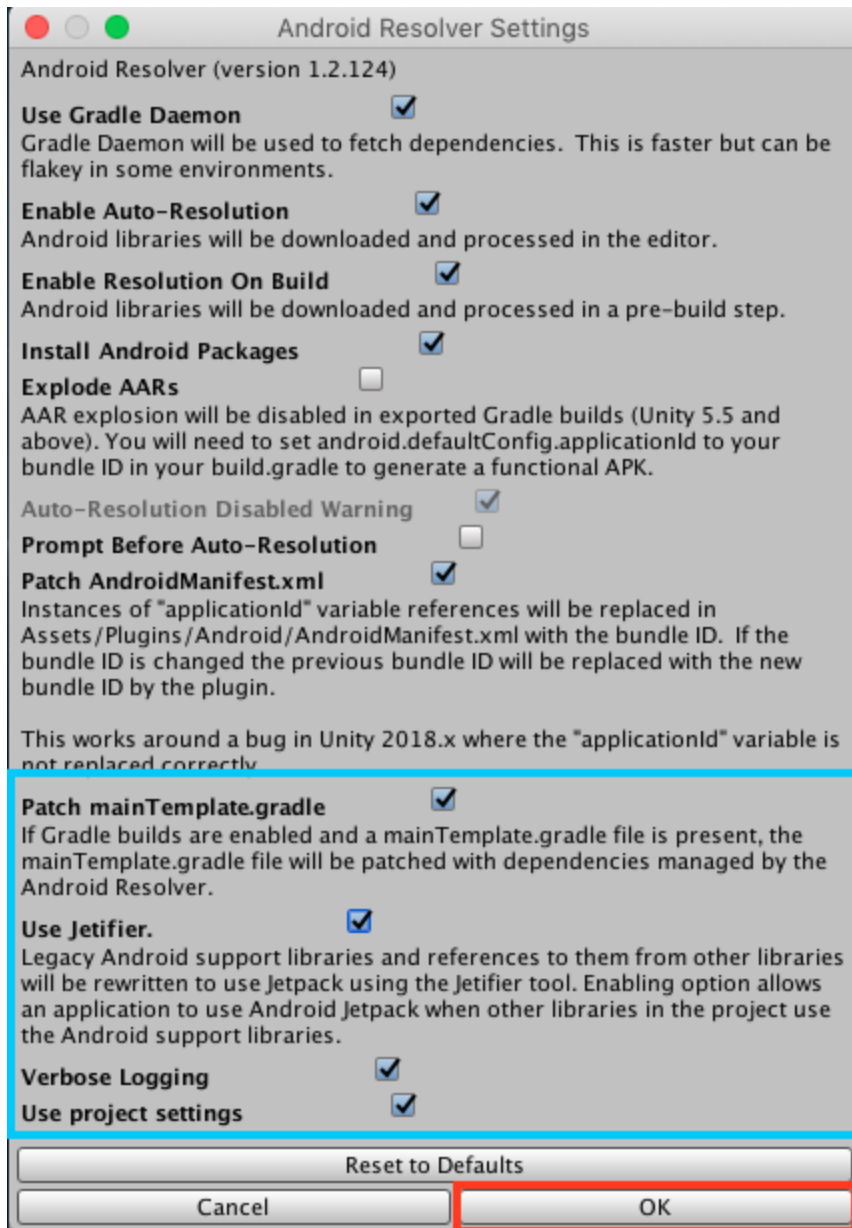
An example of some Plugins that include AndroidX are; GoogleMobileAds 3.17.0+, AdMob 4.12

Please follow these steps if you are seeing errors like this when building;

Execution failed for task ':checkDebugDuplicateClasses'.
java.lang.RuntimeException: java.lang.RuntimeException: Duplicate class
android.support.customtabs.ICustomTabsCallback found in modules classes.jar
(:androidx.browser.browser-1.0.0:) and classes.jar (:com.android.support.customtabs-27.1.1:)

1. Ensure you are using Unity 2018.4 or newer.
2. Go to Assets > Play Services Resolver > Android Resolver > Settings and check the following;
 - Patch main Template.gradle
 - Use jetifier
 - Use project settings





3. Scroll down and press "OK" to save these settings.

4. Depending on your Unity version follow one of the sections below:

Unity 2019.3 Or Newer

5A. Create a new file named Assets/Plugins/Android/settingsTemplate.gradle with the following contents.

Unity 2019.3+

// This file contents should be placed at Assets/Plugins/Android/settingsTemplate.gradle

// This works around Unity's 2019.3 bug where their root build.gradle is placing buildscript under allprojects

// On it's own it doesn't create issues however doing so means including a buildsript block in any sub projects
// such as "unityLibrary" which is generated from the template
Assets/Plugins/Android/mainTemplate.gradle does not work.
// It results in a build error of "Configuration with name 'compileClasspath' not found." on a lint task.
// Normally adding "lintOptions { abortOnError false }" bypasses any lint task errors however
// either due to a bug with the Android Gradle plugin or an order of operations this does seem to be applying in this case.
// Until Unity fixes their root build.gradle we will need to keep using this file to enable any additional Gradle plugins.

```
static void enableJetifier(Project project) {  
    project.ext['android.useAndroidX'] = true  
    project.ext['android.enableJetifier'] = true  
}
```

```
static void addBuildscript(Project project) {  
    project.buildscript {  
        repositories {  
            maven { url 'https://plugins.gradle.org/m2/' } // Gradle Plugin Portal  
        }  
        dependencies {  
            // OneSignal-Gradle-Plugin  
            classpath 'gradle.plugin.com.onesignal:onesignal-gradle-plugin:[0.12.7, 0.99.99]'  
        }  
    }  
}
```

```
static void applyPlugins(Project project) {  
    // Only apply OneSignal-Gradle-Plugin to the :app project. (Unity calls this :launcher)  
    if (project.name != 'launcher')  
        return  
  
    project.afterEvaluate {  
        it.apply plugin: 'com.onesignal.androidsdk.onesignal-gradle-plugin'  
    }  
}
```

```
gradle.rootProject {  
    it.afterEvaluate {  
        it.allprojects {  
            // Since Unity 2019.3 enabling Jetifier via mainTemplate.gradle is no longer working  
            // Enabling it for all gradle projects here.  
            enableJetifier(it)  
  
            addBuildscript(it)  
            applyPlugins(it)  
        }  
    }  
}
```

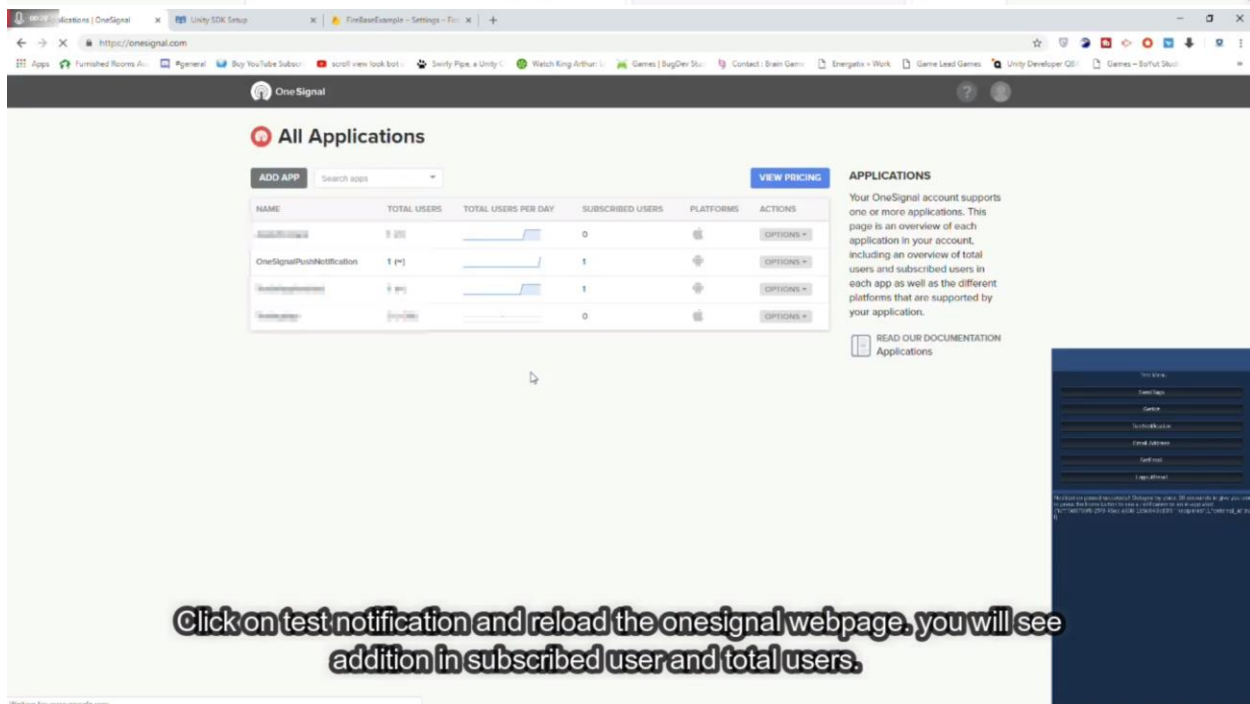
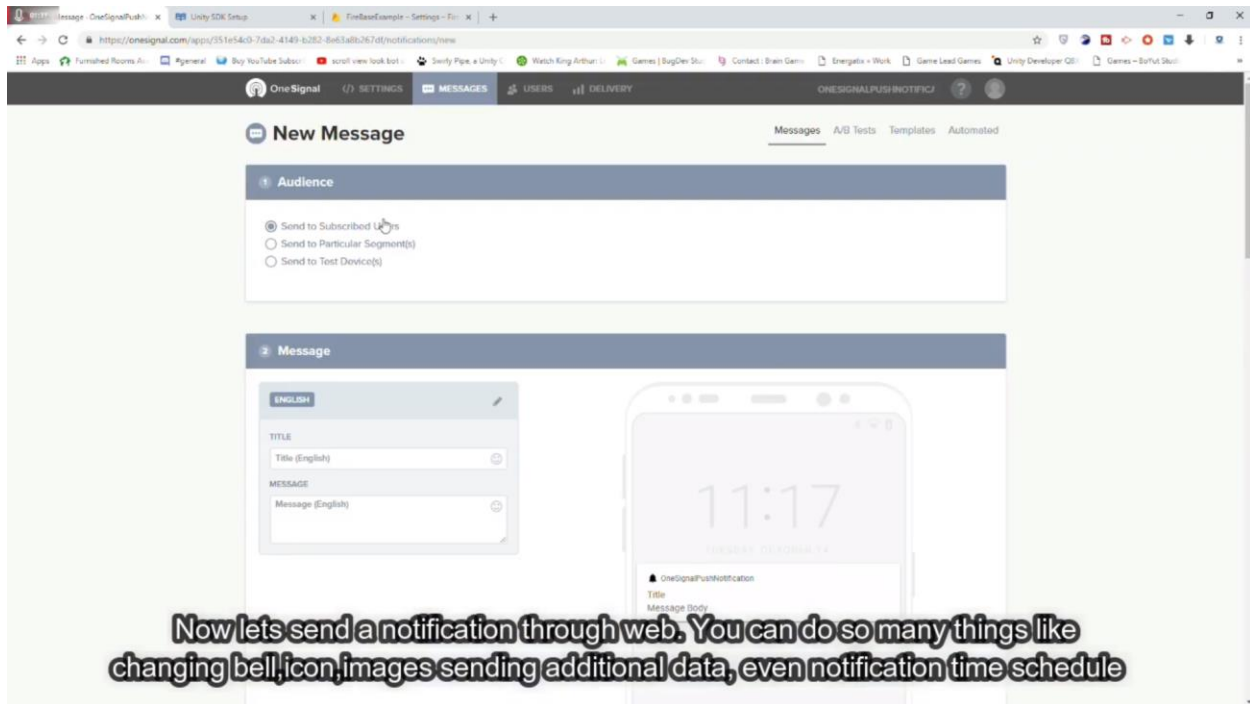
```
}  
}  
}
```

```
// Per Unity's docs /**INCLUDES/** should be at the bottom.  
// https://docs.unity3d.com/Manual/android-gradle-overview.html
```

```
// However it seem to have left out this include line  
include ':launcher', ':unityLibrary'
```

```
**INCLUDES**
```

-----4th part-----



OneSignal Push Notifications Setup

TIME TO LIVE: Default (3 days)

ACTION BUTTONS: BUTTON 1, ACTION ID, TEXT, ICON, ADD ANOTHER

Schedule

DELIVERY: Begin sending, Begin sending later

PER-USER OPTIMIZATION: Send immediately, Intelligent delivery, Optimize for battery

Confirm Before Sending

Included Segments	
Recipients	1
Title	TestNotification
Contents	My message for testing the push notification 🍌
Platforms	Android
Scheduled For	Right away

CANCEL SEND MESSAGE

Now lets send a notification through web. You can do so many things like changing bell,icon,images sending additional data, even notification time schedule

THANK
YOU!!!!!!!!!!!!!!!!!!!!!!

VIVEK KUMAR YADAV

<https://www.linkedin.com/in/vivek-kumar-yadav-02848a145/>