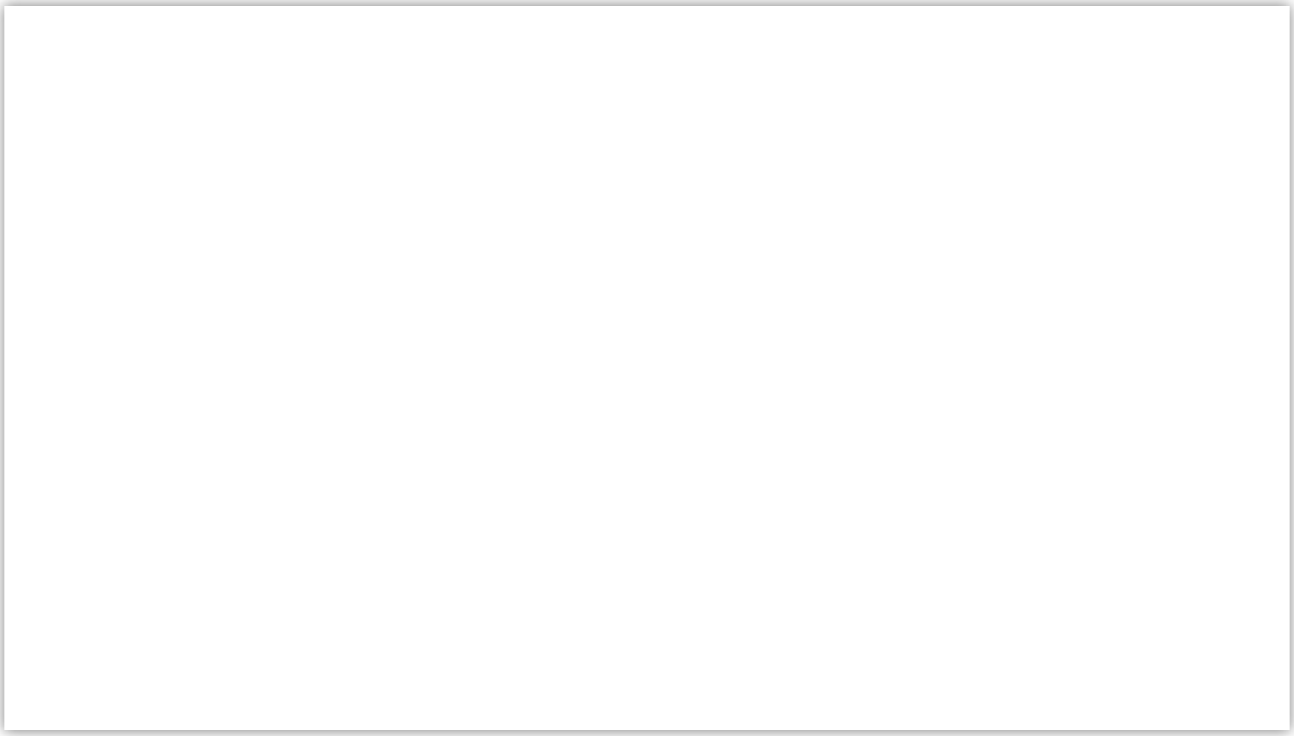
PlayTube Andorid

Getting Started

Before building your app, better take a look

Do you need a host for our items?

Best hosting 100% for your script compatible with our mobile app speed (Guaranteed).
You can get started using **UltaHost**

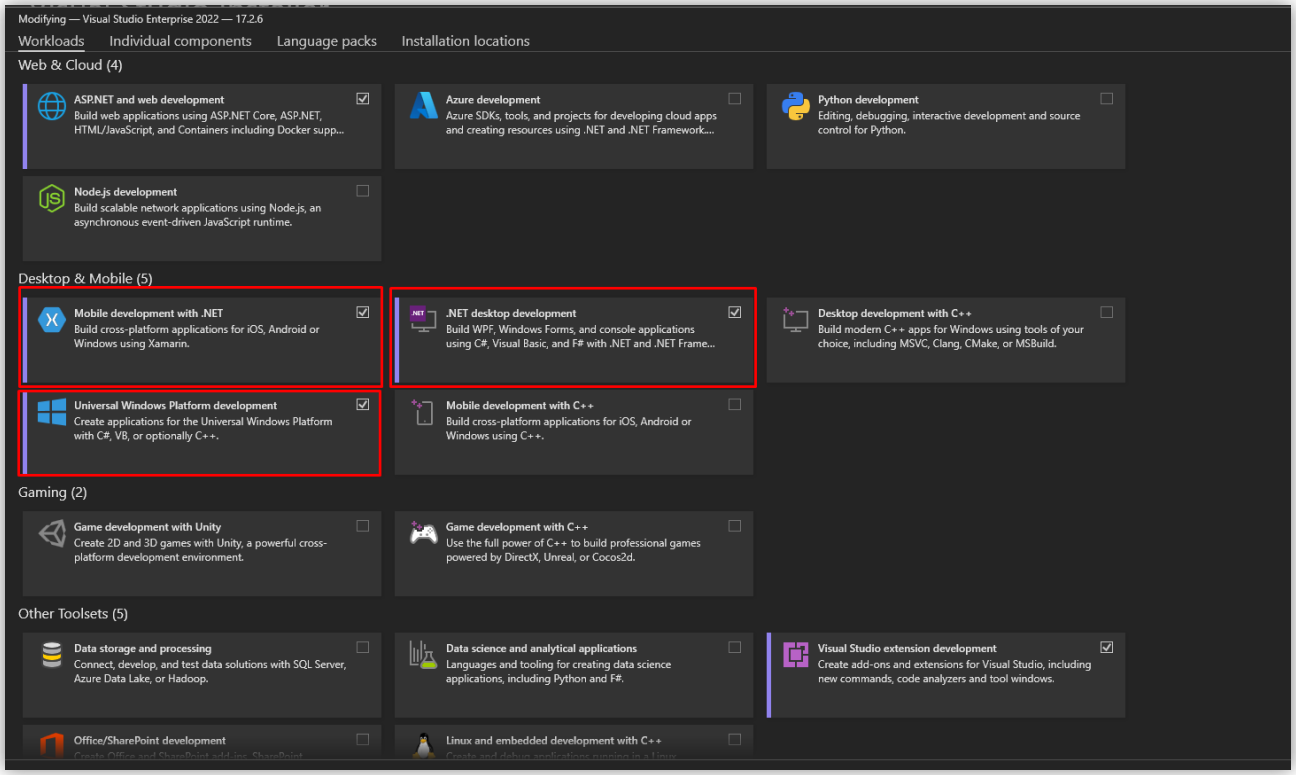


Articles

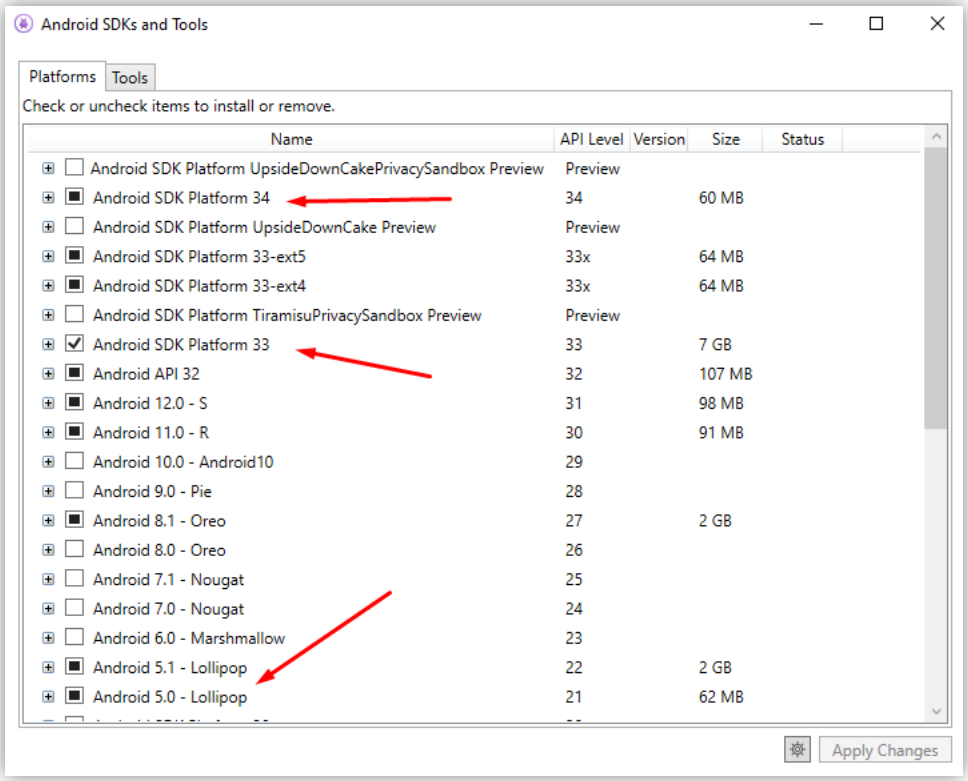
Installation

Follow the steps below to setup your app :

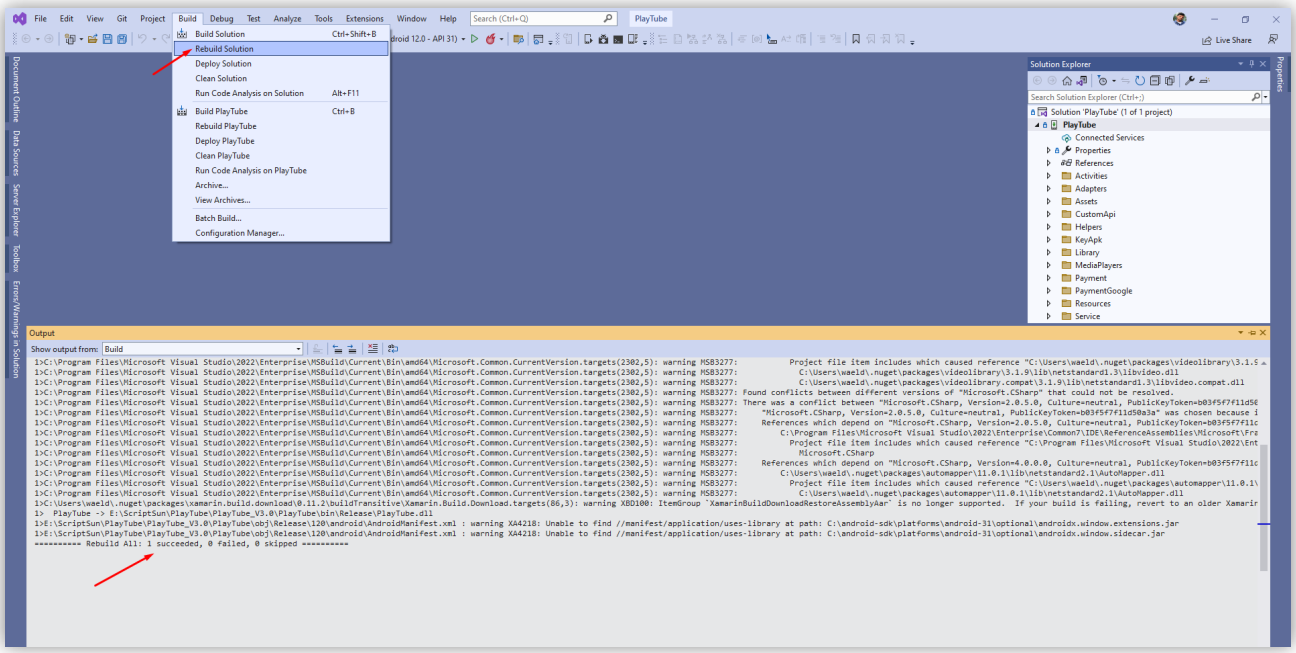
- PlayTube Script, you can get it from Here
- Download Visual Studio 2022 Here. Please remember, always use the latest code, latest visual studio, and other latest product versions of products.



- Unzip the **PlayTube** archive, extract it to a new folder, and then open the folder.
- In the main folder, you will find the solution (Name: **PlayTube.sln** Type: Microsoft Visual Studio Solution) double click on it and wait till everything is loaded.
- Install all Android SDKs in your system, From your Visual studio go to `tools menu >> Android >> SDK Manager` Select the SDK 14.0 and version 5.0



- In the Visual Studio menu bar go to Build > Build Solution, click it and wait till it finish building your project.

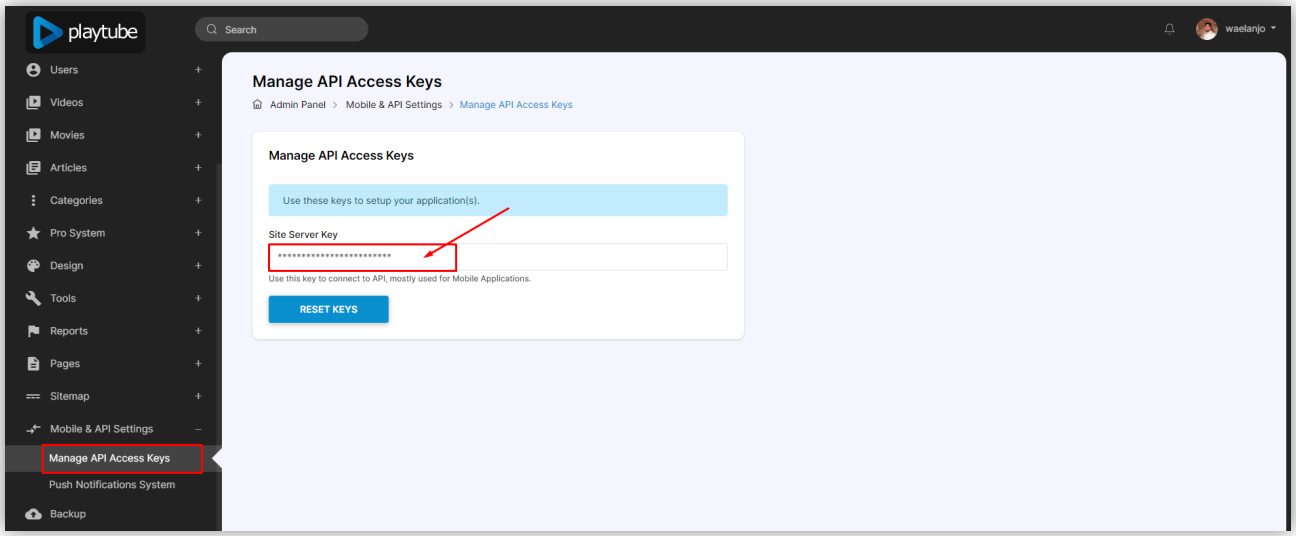


Note

If you have an error when making a build "***NDK compiler exited with an error exit code 0***" you should update to the latest version of NDK on your PC.

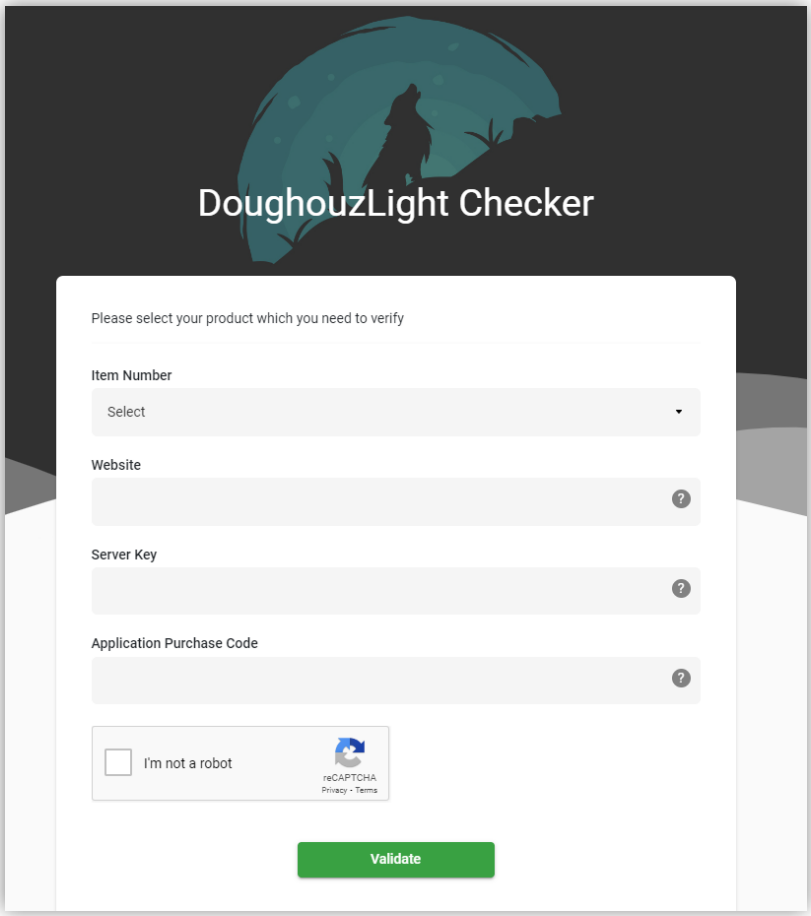
Verify Application

- You can get a purchase code from **Envato**.
- Get Your *API* ***Server_Key*** which is located **here** copy it.

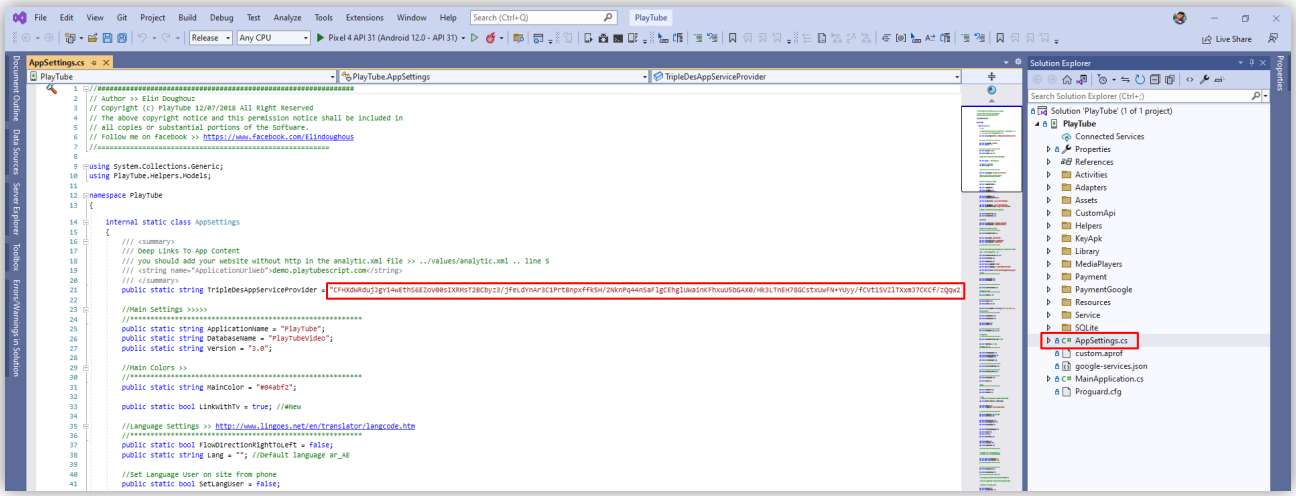


- Create your **Cert** key for your application from **Doughouzlight-License**

  PlayTube Provides Triple **DES algorithm** encryption system + **AES 256-Bit Encryption** in your mobile application to safe your own information and your own server side keys from hackers and crackers, once you are a real buyer you will not fear any cracking or **unpacking APK** actions by eligible black hat people, which may lead to leaking your sensitive server side data to the public.



- Once you have the key you will be able to add the key to your **AppSettings.cs** class.

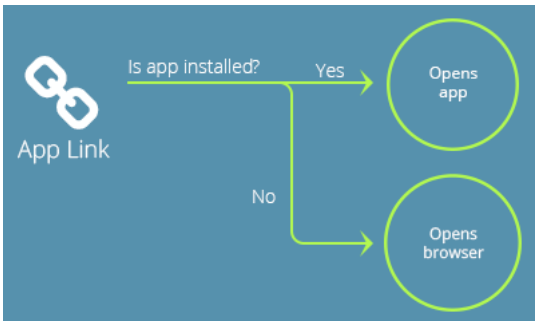


The encryption key includes all your domain information, you don't need to add your website or server key or anything else on the AppSettings class, Be sure you get the key on each new update we release to avoid any issues.

Deep Links To App Content

When a clicked link or programmatic request invokes a web URI intent, the Android system tries each of the following actions, in sequential order, until the request succeeds:

- Open the user's preferred app that can handle the URI, if one is designated.
- Open the only available app that can handle the URI.
- Allow the user to select an app from a dialog.

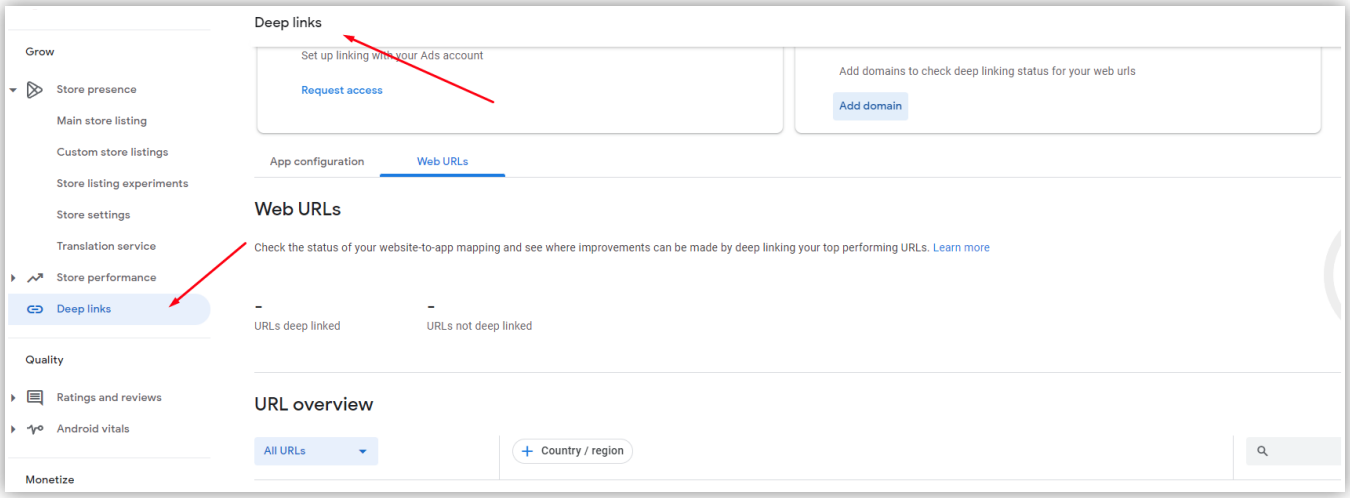


You should add your website without HTTP or HTTPS in the **analytic.xml** file

Also you should add your short URL without HTTP or HTTPS in the **analytic.xml** file

After creating a new version of your app, you have to add a link from the Google console to support it.

For more info about it, you can read it here.



You can select what is type Share System from file **AppSettings.cs** by the variables with 2 options:

- `ApplicationShortUrl`
- `WebsiteUrl`

Theme Settings

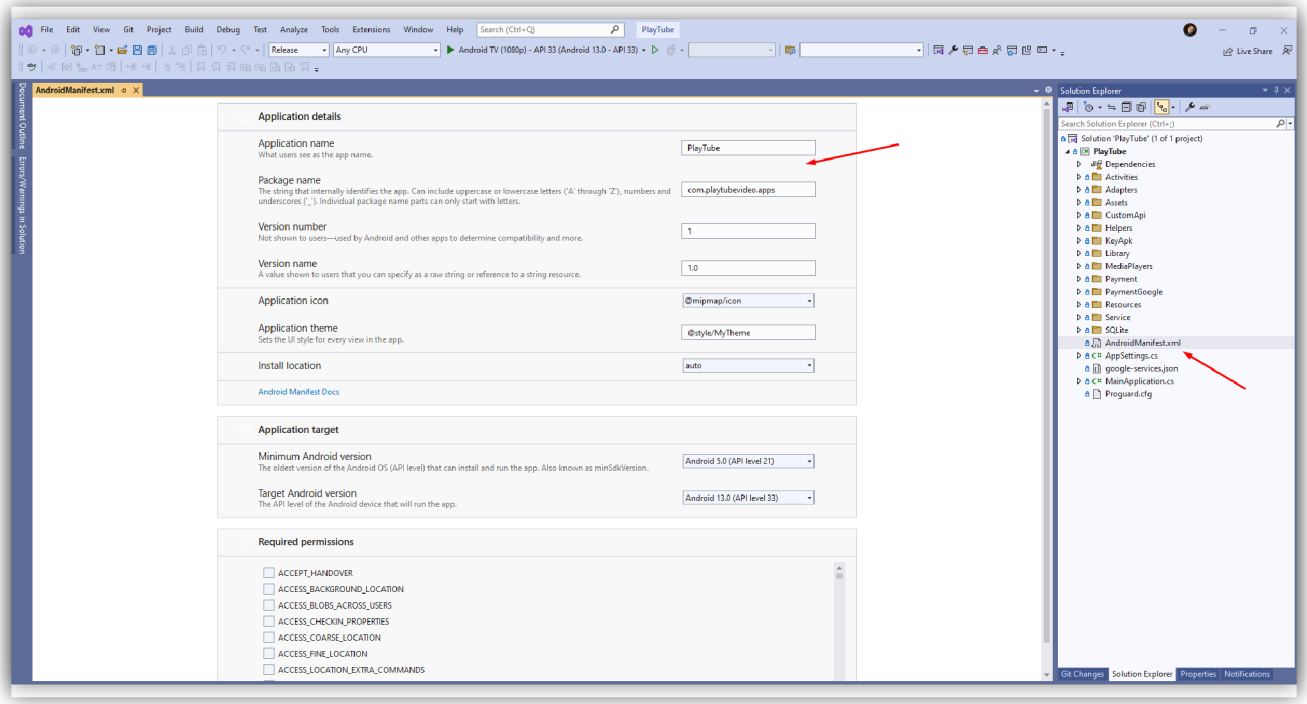
You can control the theme's functionality and customize

Articles

- Application Name
- Colors and Theme
- Package Name
- Language and Translate
- Logos and Icons and SplashScreen
- Font Style

Application Name

- From your main solution right click on the `PlayTube > Android Manifest tab` Change the names as you like and the versions also depend on your google play the last version if you have one.



- Go to **AppSettings.cs** class and change the Application Name below to your own name

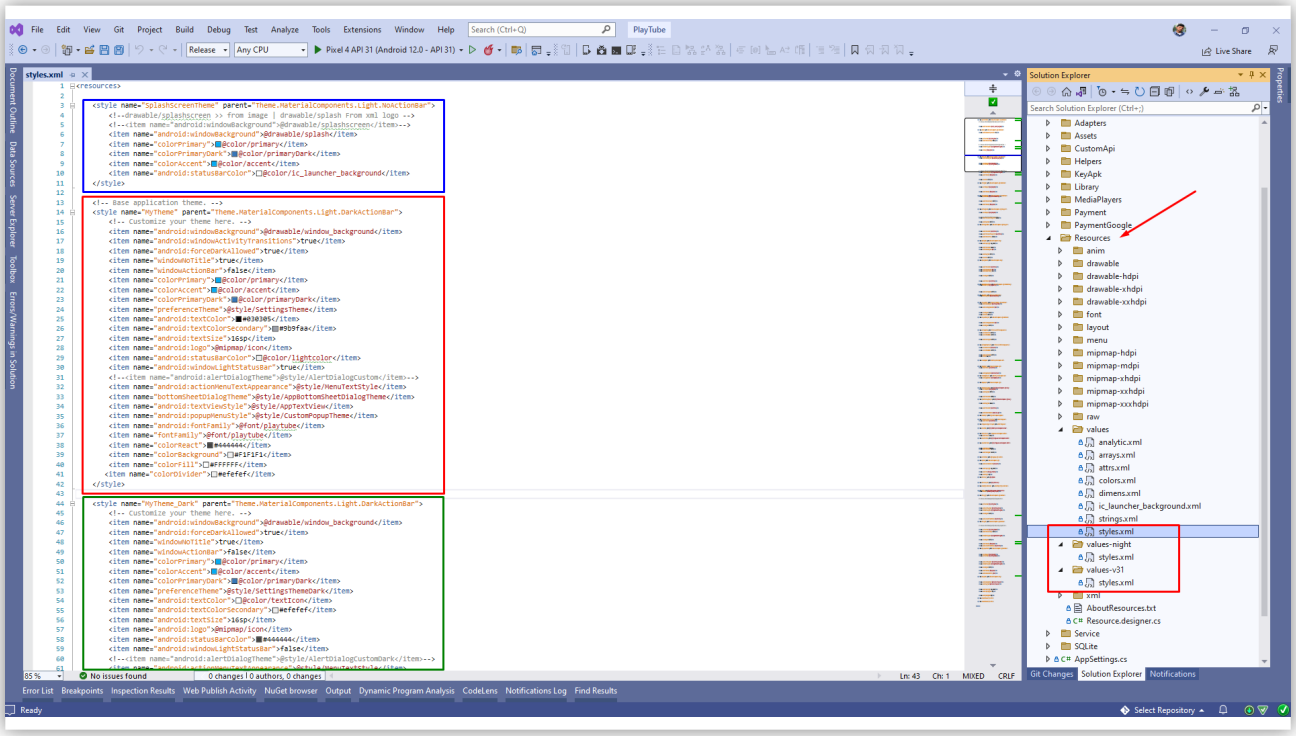
Colors and Theme

To customize and change the main color of the application follow the steps below.

- Go to **AppSettings.cs** class and change the colors below to your own colors.

- Go to the **Values folder** open the **Colors.xml** file and you will see all the XML file which is responsible for the main color and the theme of the main application.

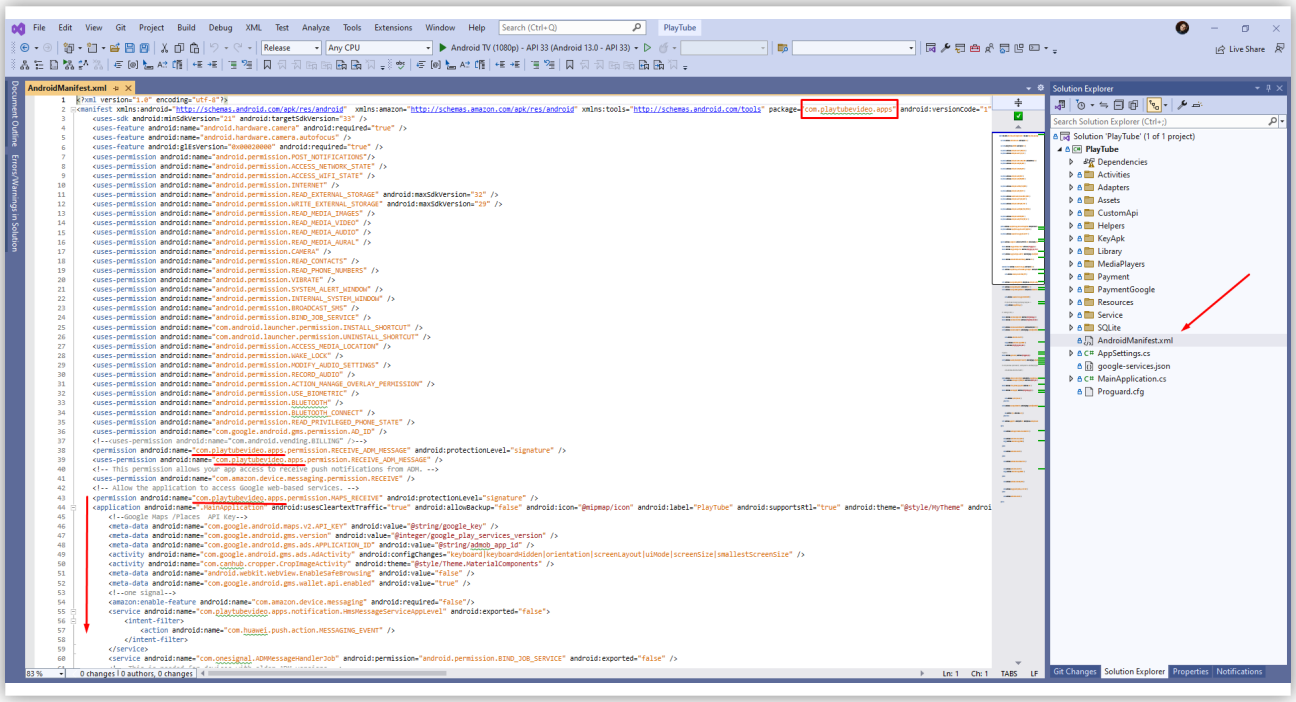
- Go to the **Values folder** open the **Styles.xml** file and you can control your theme colors and status bar and text colors and dividers and etc.



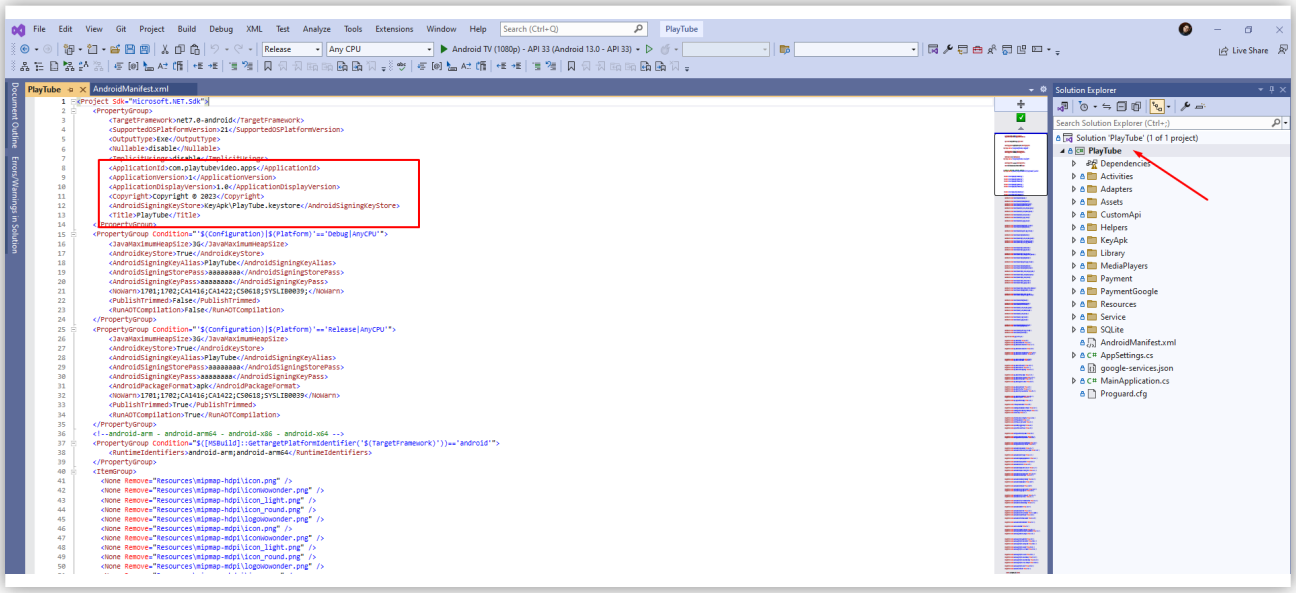
- Set Theme **Full-Screen App**:

Package Name

- From your main solution right click on the `PlayTube > AndroidManifest.xml` file to your own package name, also depend on your google play the last version if you have one.



- From your main solution double click on the `PlayTube` file to project properties.



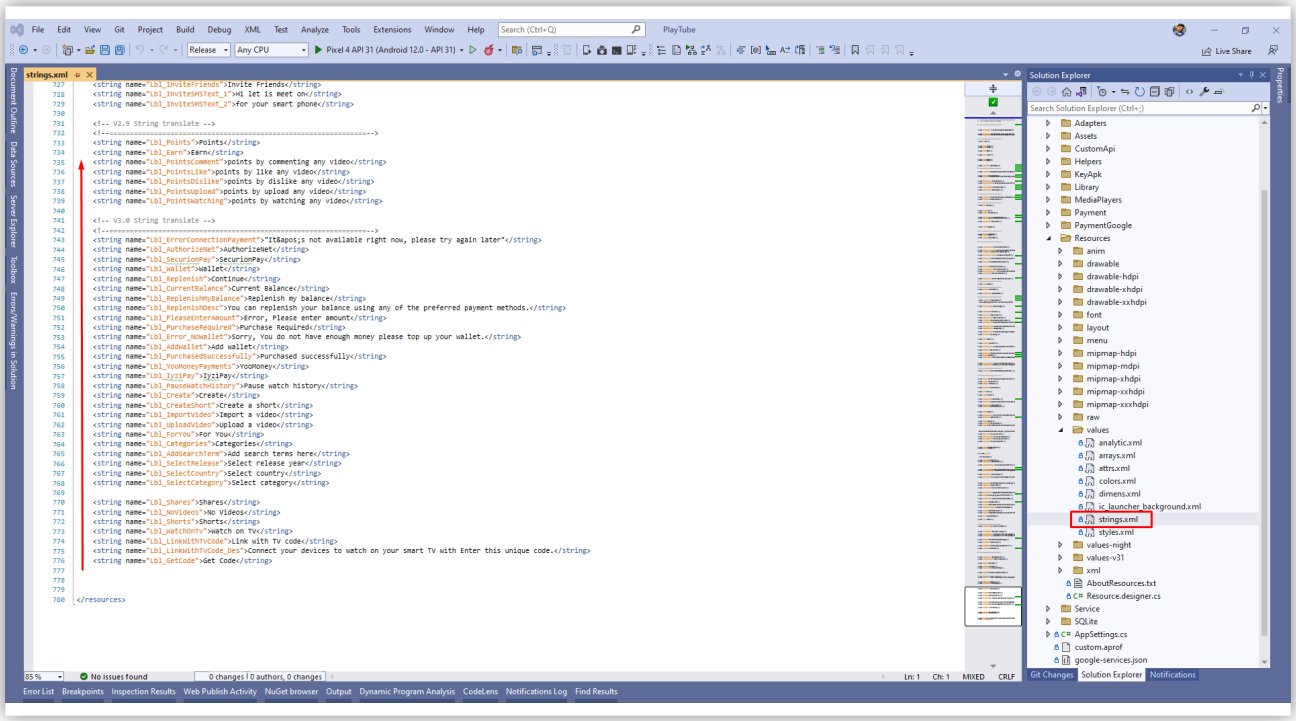
Language and Translate

App made it for you easy to translate your own words in your app and change the labels as you want, our android app support 200 languages to be added, so let's start.

English strings (default locale), /values/strings.xml:

Spanish strings (es locale), /values-es/strings.xml:

By default, the folder `Values > String.xml` contains all the English words in the app.



Generally, android considers **English** as a default language and it loads the string resources from `/res/values/strings.xml`. In case, if we want to add support for other languages, we need to create a values folder by appending the Hyphen and ISO language code.

For example, if we want to add support for **Japanese**, then we need to create a **values** folder named **values-ja** under the **res** folder and add a **strings.xml** file in it with all the strings that need to translate into **Japanese** Language.

The value-ar or value-ru has a **String.xml** file that contains the same labels but is translated to **Turkish** or **Russian** or **Arabic** and etc.

If you want to add your own new language, Create a new **value-**** folder under the **Resources folder**
then copy the main English string file string.xml and paste it inside the new **value-**** folder then you can translate your strings as you like

The folder name of Android string files is formatted as the following:

- without region variant: values-[locale]
- with region variant: values-[locale]-r[region]
- For example: values-en, values-en-rGB, values-el-rGR.

EX: In your case where you want to add greek let's say, you will just need to create a folder values-el for the Greek translation, and values-el-rGR for the country-specific Greek translation if you like to extend the lang more.

For example, suppose you have a string called "**R.string.title**" and the locale is 'el-GR', Android will look for a value of "R.string.title" by searching the files in the following order:

- `res/values-el-rGR/strings.xml`
- `res/values-el/strings.xml`
- `res/values/strings.xml`

Once we create the required files and change the device language through `Settings > Language & Input > Select Language` (Japanese), Android OS will check for the appropriate language resources available in the app.

In case, if the app supports a selected language, then android will look for the string resources in the values-(ISO language code) folder of the project. For example, if the selected language is Japanese, then it will load the string values from a values-ja/strings.xml file.

If any string value missing from the supported language (**strings.xml**) file, then android will load the missing strings from the default strings.xml file i.e. **values/strings.xml**.

Force your App to use a default language?

From your AppSettings.cs class, you can set your own default language which the app will open the first time. Also, you can force the RTL system by the variable **FlowDirectionRightToLeft** to True.

You can find what is name ISO Language Code from here

Notes

- If the text is marked **'** Please replace with the symbol **'** and add at the beginning and end of the sentence
  Example: Let's get started! >> **"**Let**'**s get started!**"**

- If the text is marked **&** Please replace it with the symbol **&**
  Example: Movies **&** Animation >> Movies **&** Animation

Note

From your **AppSettings.cs** class, you can set the user language on the site as language on your phone by the variable **SetLangUser** to True.

Logos and Icons and SplashScreen

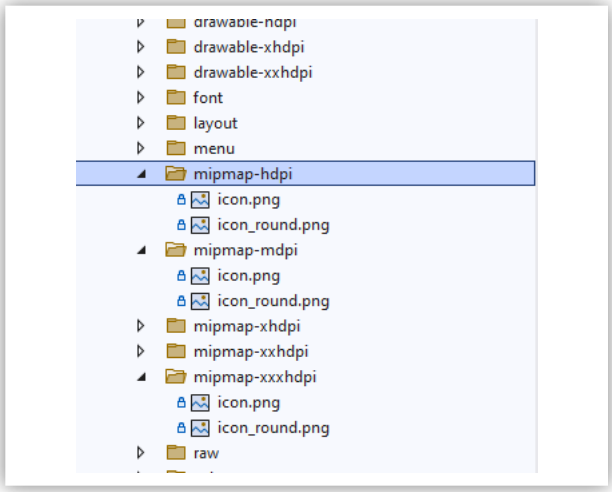
In your main solution, you will find 4 main folders with the following names

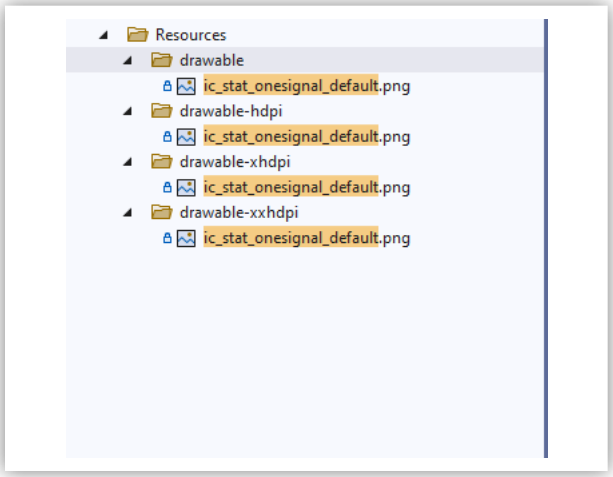
- **drawable**: for all android screen sizes by default
- **drawable-hdpi**: for small devices screen
- **drawable-xhdpi**: for normal devices screen
- **drawable-xxhdpi**: for High HD screens such as Samsung s20 and s21

Replace the icons and images you want and add them with the same name and type with an extension to an image and do not change the names of your images.

In the folders **drawable** for logo and Push Notification Icon, and in the folders **mipmap** for icon app.

For the accuracy of the icon and logo, please use this website https://appicon.co

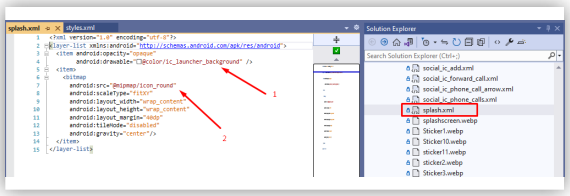




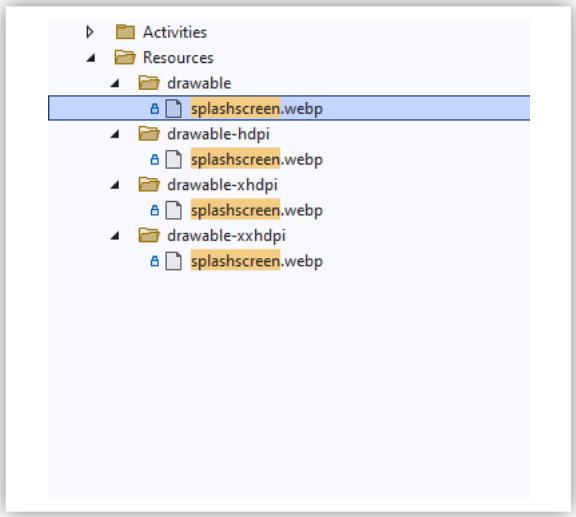
Change SplashScreen

You have two types to display the splash screen either a static image or a color with an icon/logo using an XML file.

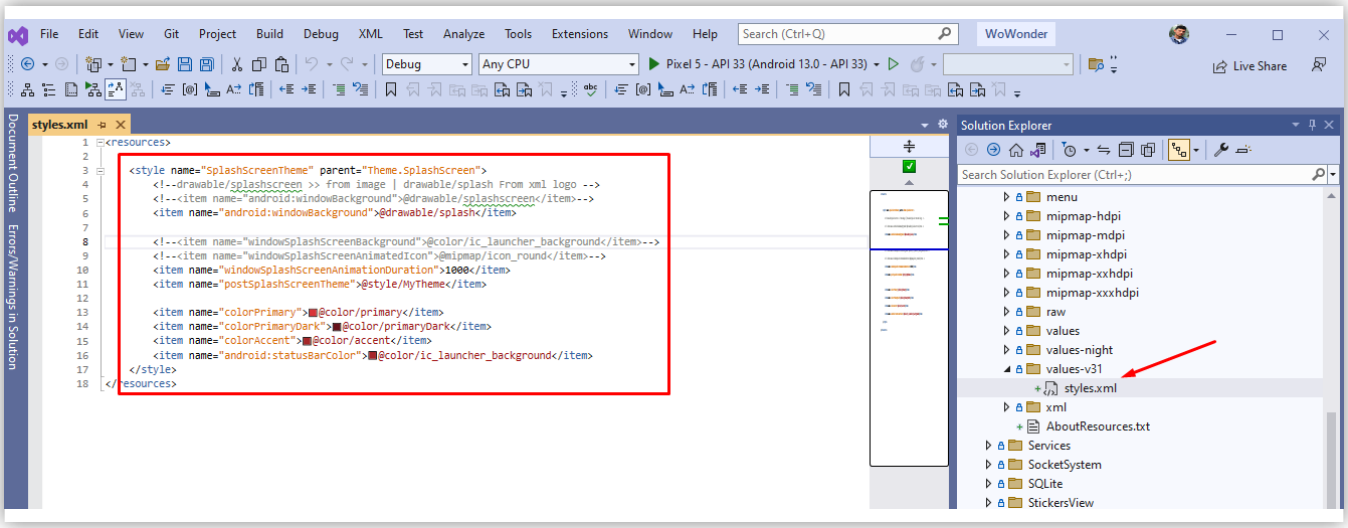
- **XML File**: In this default option
  1. You can select a background color from the file **ic_launcher_background.xml**
  2. You can select the source image icon or logo in the `drawable > splash.xml` file.



- **Static Image**: Go to each drawable folder and replace **splashscreen.webp** file with your own splash screen. The format must be the same. And then go to each `values > style.xml` splash to **splashscreen**.



- *On API 31+ (Android 12+) and up* the SplashScreen is static, just you can change the color and icon `values-v31 > style.xml`.

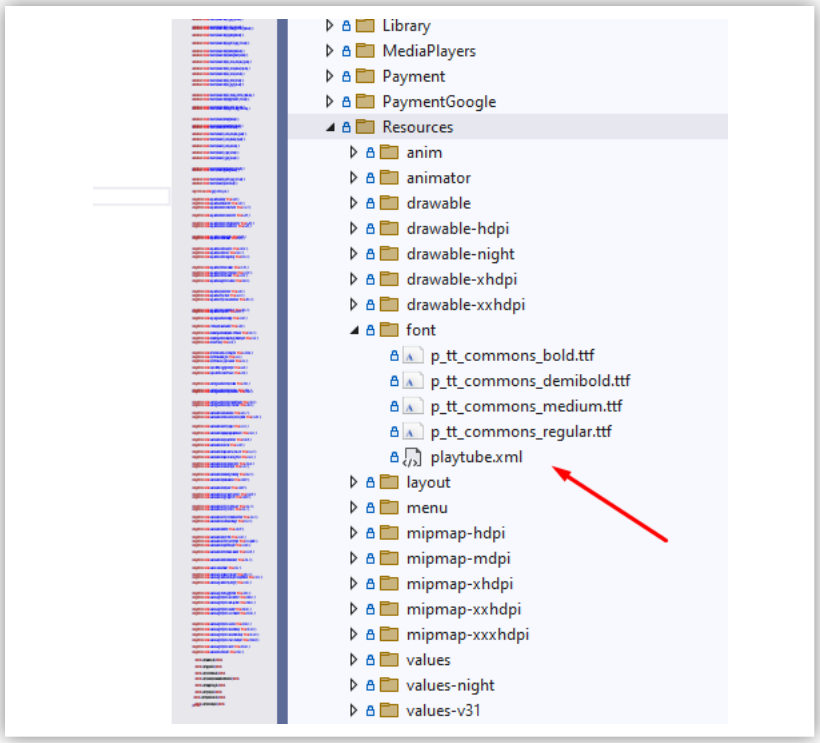


> **Note**
>
> When you change the **type** to display splash in the theme on style.xml you should change in all files **style.xml** on all folders **values**.

Font Style

To customize and change the main font of the application follow the steps below.

- if you want to add a custom font you can add a new font "***.ttf***" to the font folder and change the name on the **playtube.xml**



- Go to the **font folder** open the **playtube.xml** file and you will see all the XML file which is responsible for the main font of the main application.

General Setup

Articles

- OneSignal Notifications
- Firebase Account

OneSignal Notifications

What is OneSignal?

OneSignal is a high-volume and reliable push notification service for websites and mobile applications. They support all major native and mobile platforms by providing dedicated SDKs for each platform, a Restful server API, and an online dashboard for marketers to design and send push notifications.

- Sign in to the Onesignal Console at https://onesignal.com.
- Click on **New App/Website**
- Choose your app name and platform then click next.
- Fill out the form with your website information, then click save.
- On the next page, ignore the page and click on Finish.
- On the top navbar, click on **Keys & IDs**
- Grab your OneSignal App ID & Rest API Key then go to `Admin Panel -> API Settings -> Push Notifications Settings`
- Edit the following options for Android & IOS:

1. **OneSignal APP ID** – Enter the OneSignal App ID you created in the previous chapter
2. **REST API Key** – Enter the Rest API Key you created in the previous chapter

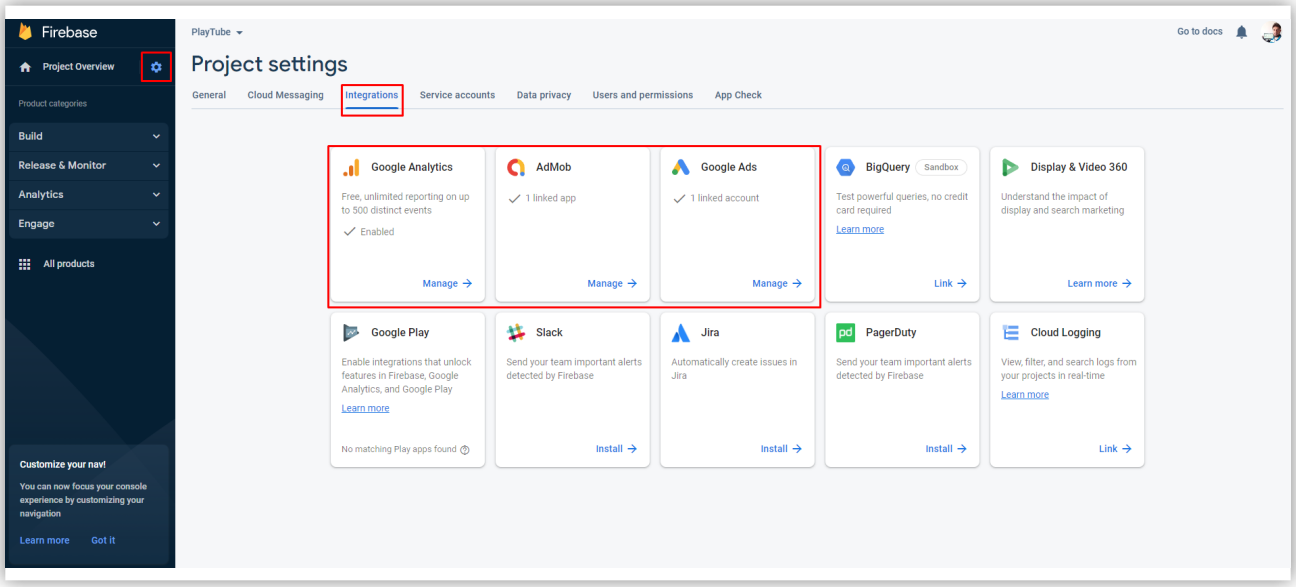
- Finally, go to the file **AppSettings.cs** in the project and add the app id.

> **Note**
>
> If You want to Disable Notification and Onesignal on your app set the variable **ShowNotification**.

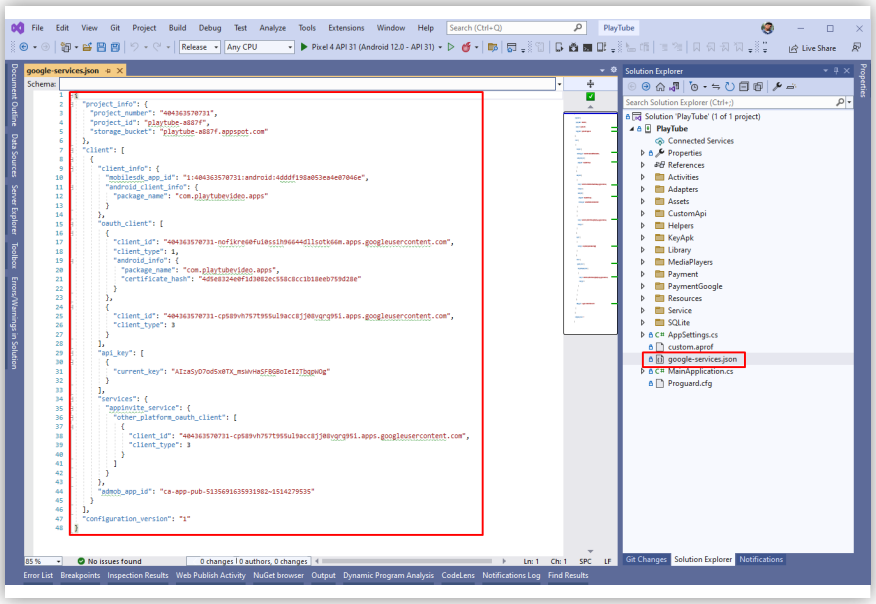
Firebase Account

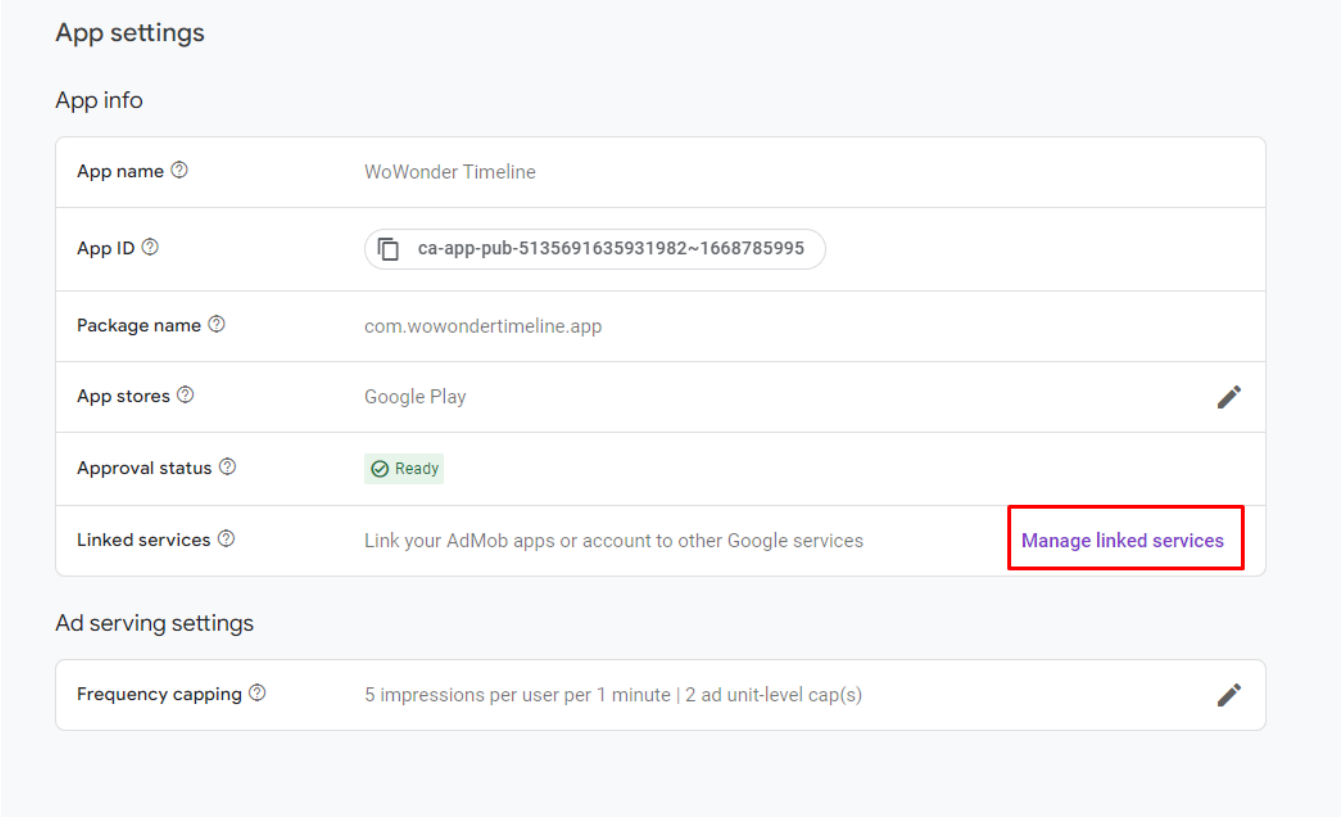
Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

- After creating a firebase account from the link you will need to create a new project as well.
- then go to **project settings**.
- Go to the **Integrations** tab and select the item below as the screenshot:



- Finally Go to **General** Tab and scroll down then Download the file JSON `"google-services.json"` and add it to your project code.





Add your Package android app name and press the link icon

Advertisement Setup

Wherever you are. whatever your app can do ads can help you grow lasting revenue.

- You can select when it can show ads in the app for **all users** or for **Non-Professional "not Pro"** users.

- You can limit the number of times the ad appears according to the number of events.

Articles

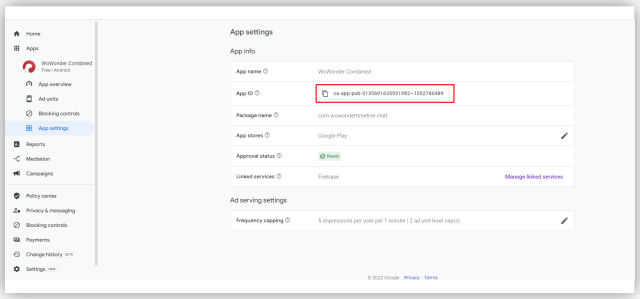
- Integrate AdMob (Google ADS)
- Integrate Facebook ADS
- Integrate AdsAppLovin

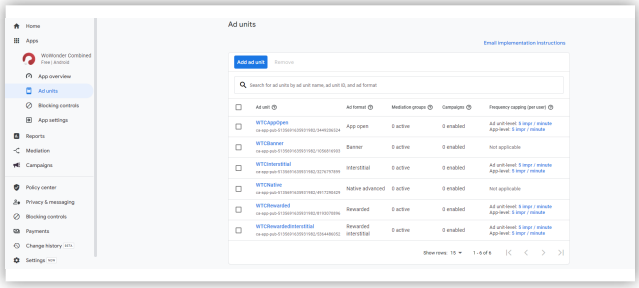
Integrate AdMob (Google ADS)

After creating a Google AdMob account and you are all ready to start your own AdMob ads system on your mobile app.

You will need to follow a few steps before seeing your ADS appearing in your mobile application.

- Find your app IDs & ad unit IDs
  An app ID is a unique ID number assigned to your apps when they're added to AdMob. The app ID is used to identify your apps.
- An ad unit ID is a unique ID number assigned to each of your ad units when they're created in AdMob. The ad unit ID is added to your app's code and used to identify ad requests from the ad unit.
- **Find an app ID:**
  1. Sign in to your AdMob account at https://apps.admob.com.
  2. Click Apps in the sidebar.
  3. Click View all apps.
  4. Click the icon in the App ID column to copy the ID of an app.
- **Find an ad unit ID:**
  1. Sign in to your AdMob account at https://apps.admob.com.
  2. Click Apps in the sidebar.
  3. Click the name of the app associated with the ad unit. Note: If you don't see it in the list of recent apps, click View all apps. Then, click the name of the app.
  4. Click Ad units in the sidebar.
  5. Click the icon in the Ad unit ID column to copy the ID of an ad unit.





- Copy the APP_ID and Banners ads and put them on your **analytic.xml** file in your solution code as below.

- Copy the Interstitial & Rewarded & Native & AppOpen and put them on your **AppSettings.cs** class file in your solution code as below.

Warning about new privacy on google Admob

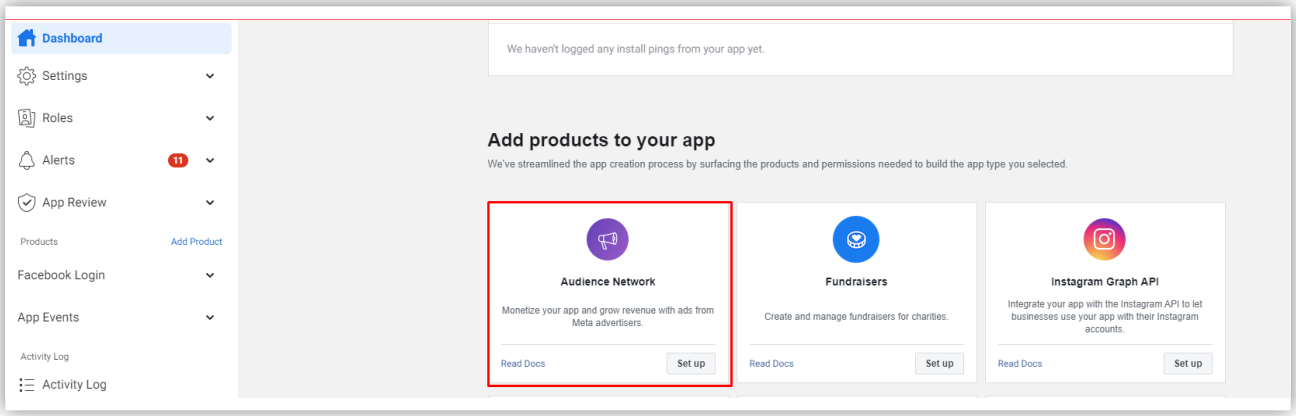
Should be added app-ads.txt in the server:

- Should this content file text https://prnt.sc/vm086w
- and the link should be `http(s)://www.demo.com/app-ads.txt`

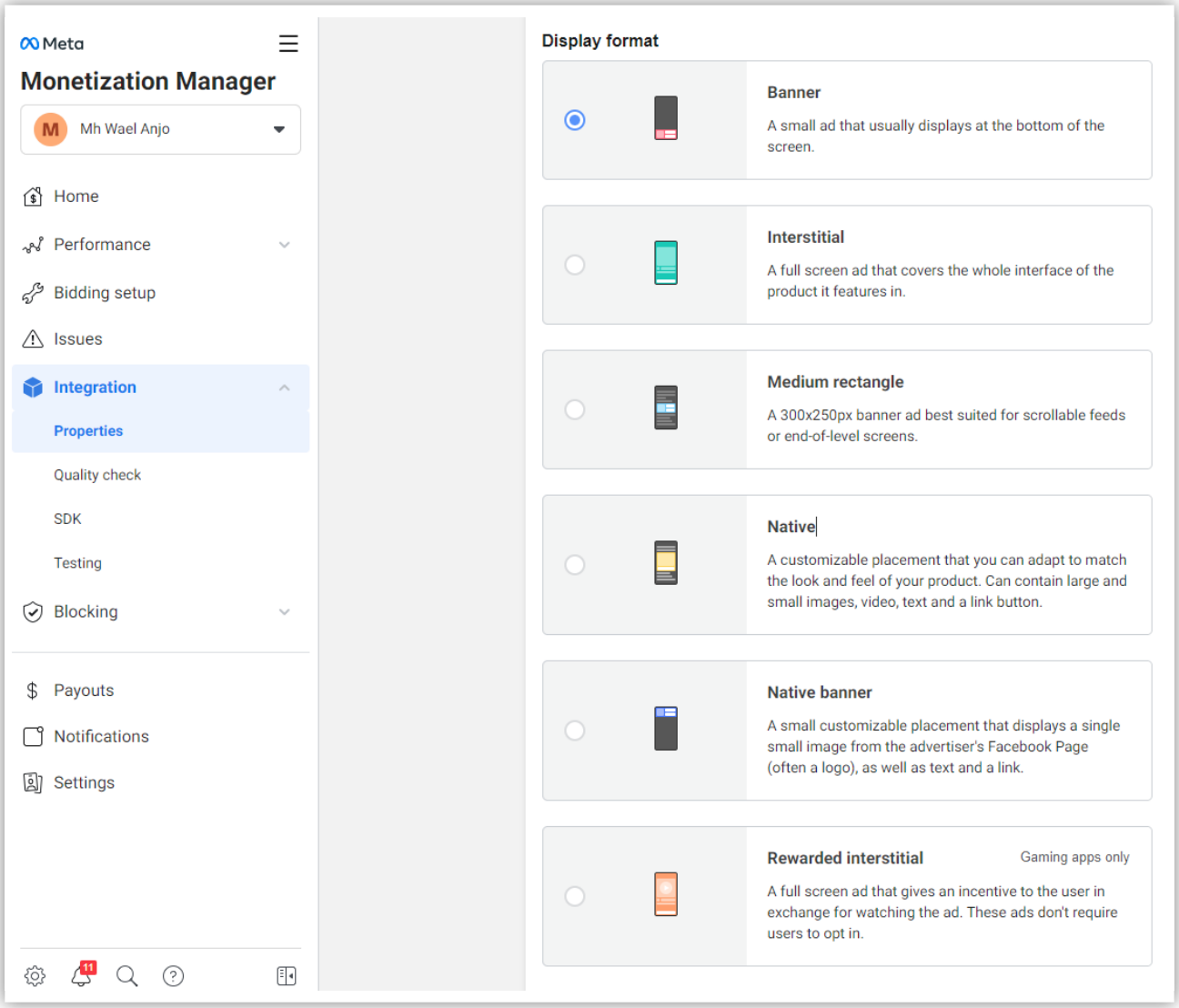
For more details, you can see the video

Integrate Facebook ADS

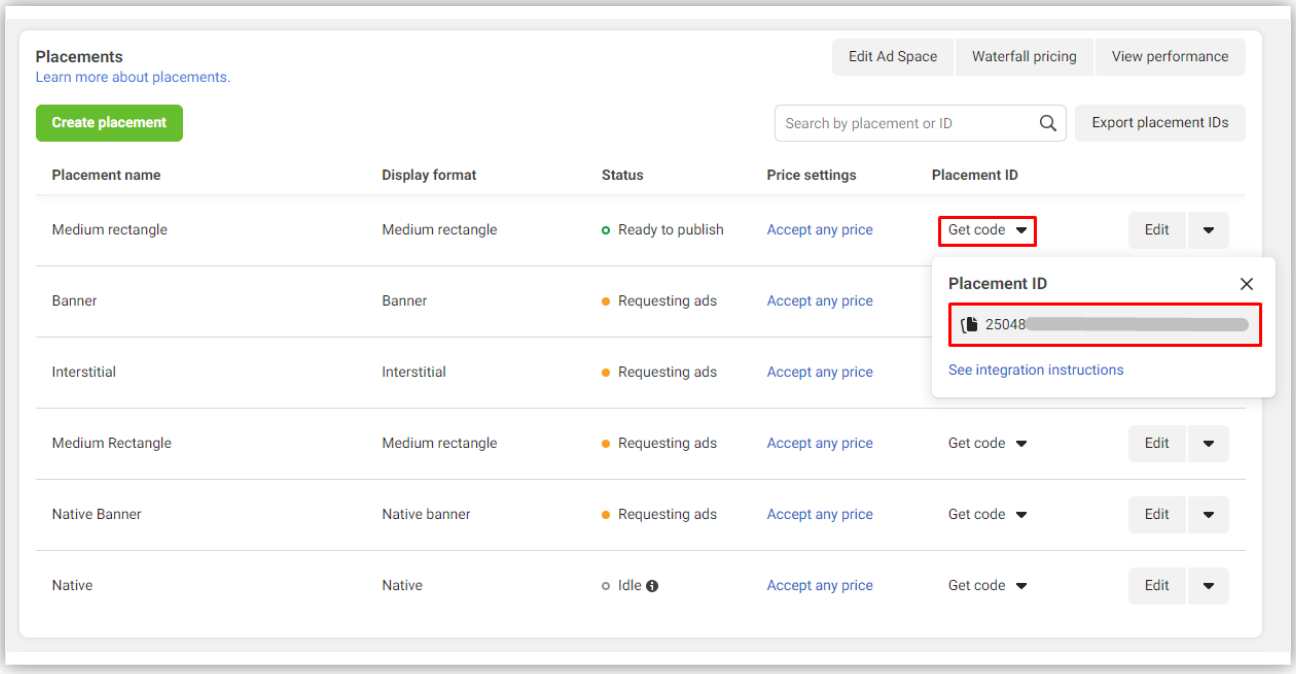
After creating a Developer's Facebook account and you are all ready to start your own Facebook audience network on your mobile app.



- From your Facebook audience network dashboard, choose to add a new Android Application.
- From the list of placement types below, choose the one you want to create and use in our app (Banner, Interstitial, Rewarded Video).



- After choosing the placement type, you will get the placement ID as shown below.



- Copy the Placement and go to file **AppSettings.cs**.

- You can activate and deactivate according to the type of advertisement on your **AppSettings.cs** class file in your solution code as below.

Note

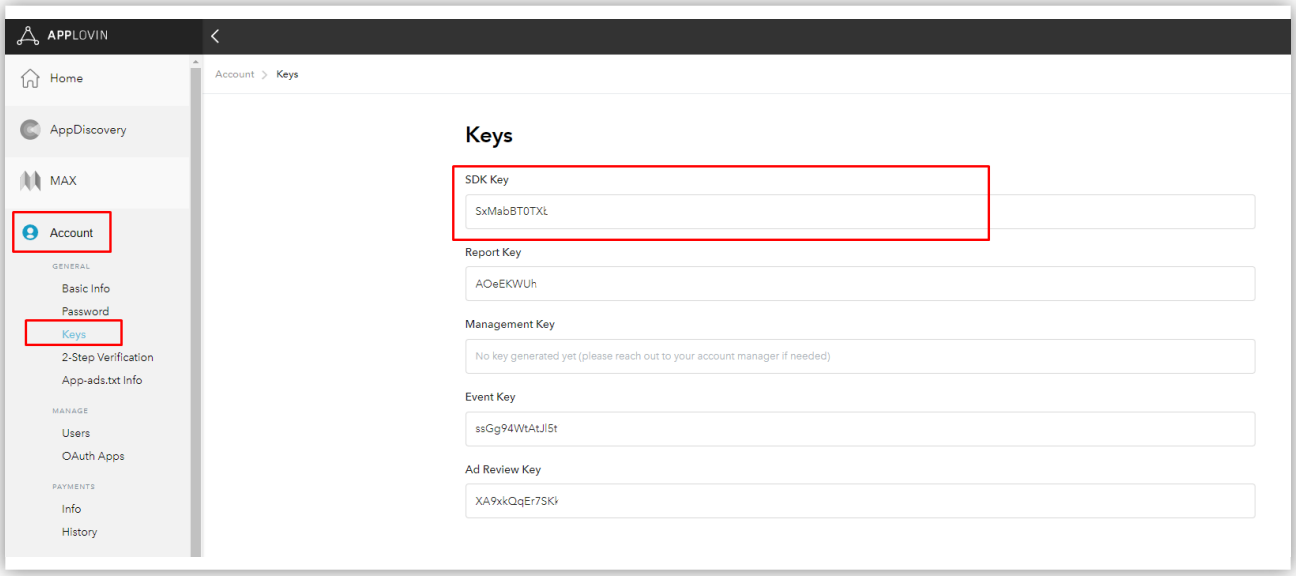
When you need to enable Facebook ads don't forget to add app_id on the **analytic.xml** file.

```
<string name="facebook_app_id">362090*****</string>
```

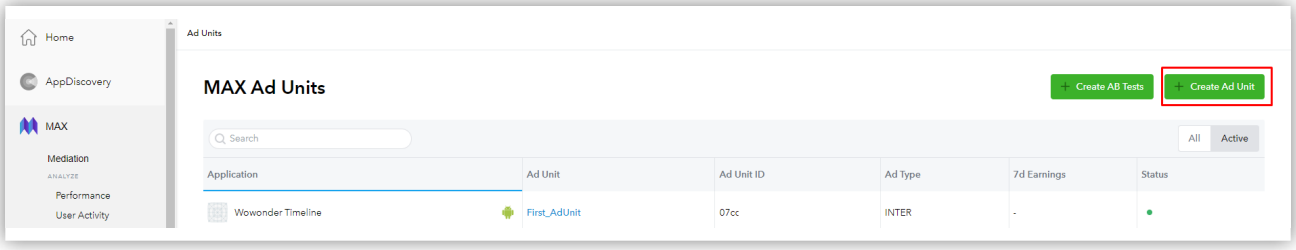
Integrate AdsAppLovin

After creating an AppLovin account and you are all ready to start your own ads on your mobile app.

- From Your Account on the AppLovin Dashboard, copy the SDK Key and go to file **analytic.xml**.



- From Your AppLovin Dashboard, choose to Create New App for Android.
- Then create a new **MAX Ad Unit** (Interstitial, Rewarded, Banner).



- Copy the Placement and go to file **AppSettings.cs**

- You can activate and deactivate according to the type of advertisement on your **AppSettings.cs** class file in your solution code as below.

Social Logins

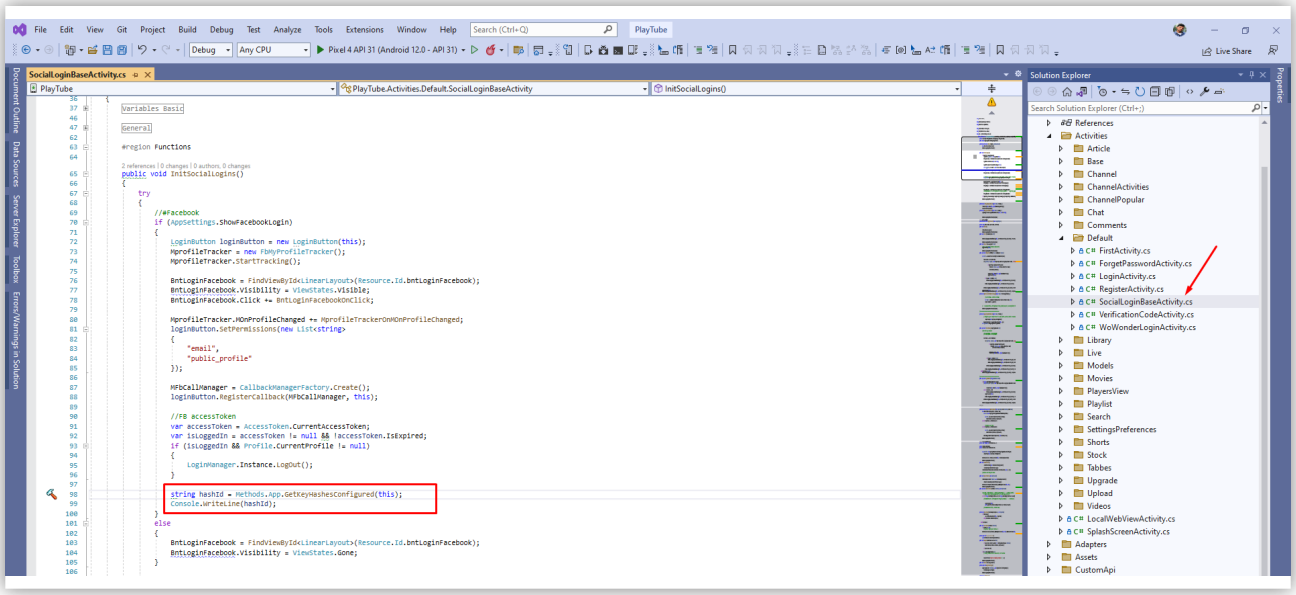
Articles

- Facebook Login
- Google Login
- WoWonder Login

Facebook Login

- Log in to your Facebook account.
- Go to Facebook for Developers, click on **My Apps** and press **Create App**.
- Select **Consumer**. and click **Next**.
- Set the **Display Name** of your application.
- Enter the **Contact Email**.
- Click on **Create App**.
- Navigate to **Facebook Login** and press the **Set up** button.
- Select **Android** from the displayed platforms.
- Add your **package name** that uniquely identifies your Android app, and add Activity Class Name {**packageName}.LoginActivity**
- Add **Key Hashes**, you can get this from **LoginActivity.cs** on the code:

```
string hashId = Methods.App.GetKeyHashesConfigured(this);
```



- Enable Single Sign On
- Add in the file **analytic.xml** `string` elements with the names `facebook_app_id`, `fb_login_protocol_scheme` and `facebook_client_token`, and set the values to your App ID and Client Token.

- Then go to the **Settings** on the left nav bar section and choose **Basic**.

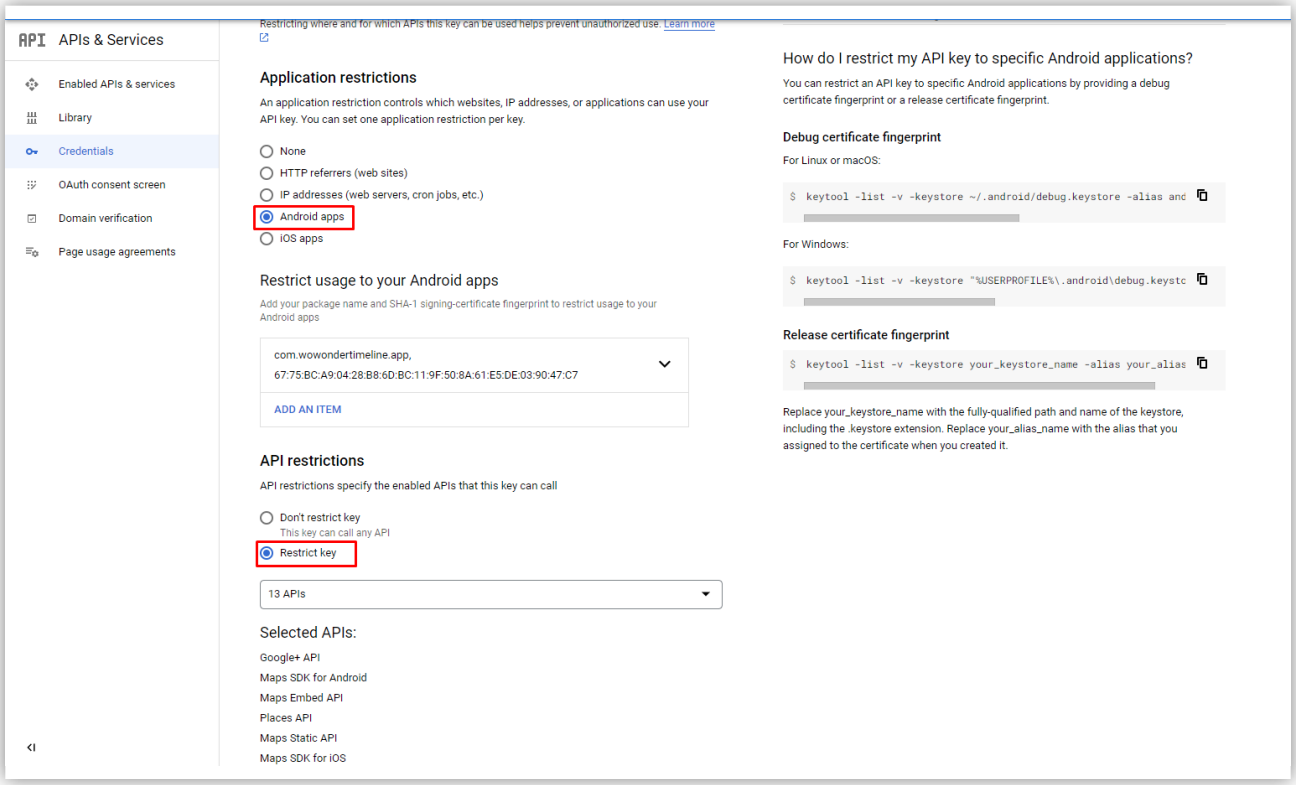
- In the appeared menu some of the fields are generated automatically. All you need to do is fill in the rest of them, such as **App Domains**, **Contact Email**, **Privacy Policy URL**, and **Terms Of Service** in particular.
    1. **App Domains** – Enter your domain name, without HTTP or HTTPS (example.com, www.example.com).
    2. **Privacy Policy URL** Enter your website's privacy policy URL (example.com/terms/privacy).
    3. **Terms of Service URL** Enter your website's Terms of Service URL (example.com/terms/terms).
    4. **User Data Deletion** Enter your website's Terms of Service URL (example.com/terms/terms).
    5. **Category** – Choose your app category (Social networks & dating).
- Click on **Save changes**.
- Go to the left nav bar section and choose **Facebook Login** then click on **Settings**.
- Scroll down a bit and find **Valid OAuth Redirect URIs**
- Add the following Redirect URIs:
    1. `https://yourwebsite.com/login-with.php?provider=Facebook`
    2. `https://www.yourwebsite.com/login-with.php?provider=Facebook`
- Click on **Save changes**
- On the top, you'll see this message **Your app has Standard Access to public_profile. To use Facebook Login, switch public_profile to Advanced Access** Click on **Get Advanced Access**
    1. Enter **email** in the search box, then click **Get Advanced Access**, and confirm the form.
    2. Enter **public_profile** in the search box, then click **Get Advanced Access**, and confirm the form.
- The last step is to set the app mode to **Live**
    1. On the top navbar, you'll find **App Mode**, click the toggle, and set the app to **Live**
    2. Complete **Data Use Checkup** (If required)

> **Note**
>
> You can activate and deactivate your app by setting the variable **ShowFacebookLogin**.

For more details, you can see the video

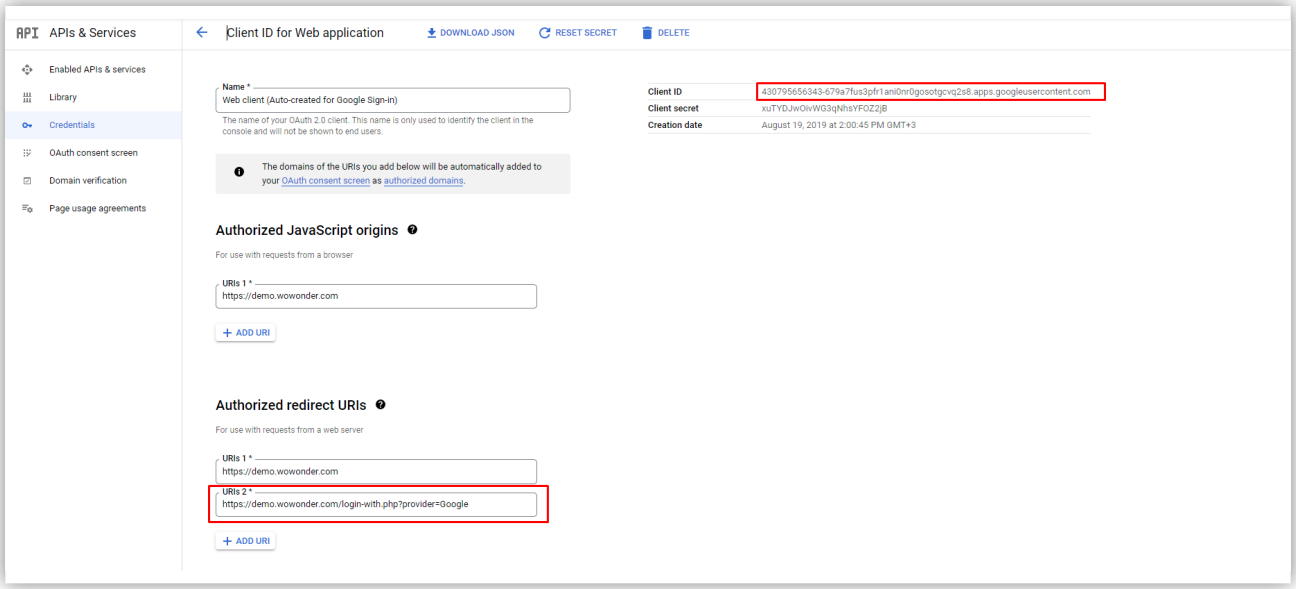
Google Login

- Start Integrating Google Sign-In into Your Android App
- After this click on Create Project link to create a new project.
- Enter **Project Name** and click on **Create** button.
- Select the project from the dropdown menu beside **Android**.
- Add **Package name** from your AndroidManifest.xml file, and **SHA-1**.
- Click on APIs & Services then click on **+ENABLE APIS AND SERVICES**.
- Search for **Google+** and enable **Google+ API**.
- Once enabled, click on Credentials, on the top nav-bar, click on **+Create credentials** then click on the **API key**
- Select Application restrictions **Android apps** with **Package name** from your AndroidManifest.xml file, and **SHA-1**.
- Select API restrictions specify the enabled APIs that this key can call
- Grab the key and Click **Save**.



- Once enabled, click on Credentials, on the top nav-bar, click on **+Create credentials** then click on **Oauth client ID**
- In **OAuth consent screen** page, choose **External**, fill the form and click **Save and Continue**.
    - In **Authorized domains** section, make sure to add your own domain name with or without www.
- On the next page, you'll see the button **Add or remove scopes**, click on it.
    - Look for **auth/userinfo.email** and select it.
    - Look for **auth/userinfo.profile** and select it.
    - Scroll down and click on **UPDATE**.
- If you successfully added the new scopes, click on **SAVE AND CONTINUE**.
- Ignore the **Test users** section and click on **SAVE AND CONTINUE**.
- Scroll down and click on **BACK TO DASHBOARD**.
- On the next page, click on **PUBLISH APP** and publish your application.
- On the left nav bar, click on **Credentials**, on the top nav-bar, click on **+Create credentials** then **Oauth client ID**
    - In **Application type**, choose web application.

- In **Authorized redirect URIs**, add the following URIs:
    1. `https://yourwebsite.com/login-with.php?provider=Google`
    2. `https://www.yourwebsite.com/login-with.php?provider=Google`



- Finally, go to the file **analytic.xml** add the key as below:

- Also, add Client ID in your **AppSettings.cs** class file in your solution code as below:

Note

You can activate and deactivate your app by setting the variable **ShowGoogleLogin**.

Info

By integrating Smart Lock for Passwords into your Android app, you can automatically sign users into your app using the credentials they have saved.

You can activate and deactivate your **AppSettings.cs** class file in your solution code **EnableSmartLockForPasswords**.

For more details, you can see the video

WoWonder Login

If you are using WoWonder script, you can integrate your site.

- Login into your website that is using **WoWonder** script as admin.
- Go to https://yourwebsite.com/create-app, replace **yourwebsite.com** with your own domain name.
- To correctly create the application, please fill in the application sections, which are explained below.
  - **Name** – Your application name. This is used to attribute the source user-facing authorization screens. 32 characters max.
  - **Domain** – Enter your website URL (That is using **PlayTube** Script).
  - **Redirect URI** – `https://yourwebsite.com/wo_login.php`, replace **yourwebsite.com** with your own domain name (That is using **PlayTube** Script).
  - **Description** – Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.
  - **Image** – Upload an icon for your application.
  - Click on **Create**.
- Get your **App ID** and **App Secret**.
- Go to `Admin Panel -> Settings -> Social Login Settings -> WoWonder Configuration` & Edit the following options:
  1. **API Key** – Enter the App ID.
  2. **API Secret** – Enter the App Secret.
  3. **WoWonder Domain** – Enter your website URL (That is using **WoWonder** Script).
  4. **WoWonder Icon** – Enter your icon image URL (That is using **WoWonder** Script), example: `https://demo.wowonder.com/themes/default/img/icon.png`
  5. Enable **WoWonder Login**.
- Finally, add info keys in your **AppSettings.cs** class file in your solution code as below:

Note

You can activate and deactivate your app by setting the variable **ShowWoWonderLogin**.

Payment Integration

Articles

- Stripe
- PayPal

- CashFree
- RazorPay
- PayStack
- PaySera
- IyziPay
- 2CheckOut
- Authorize.Net
- SecurionPay
- Yoomoney
- In-App Billing Purchases
- AamarPay
- FlutterWave

Stripe

- Go to Stripe & create a new account.
- Login to your dashboard & turn your account on by flipping the switch from “Test” to “Live”.
- Click **Activate Account** in the popup window to fill out the standard business details Stripe needs to send you payments.
- Once you’ve completed the entire form, click **Activate Account**.
- Now that your account is live, go to **Your Account** and click on **Account Settings**.
- Click the **API Keys** tab & copy the **Live Secret Key** & **Publishable Key**.
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find Stripe and edit the following options:
  1. Enable **Stripe Payment Method** by clicking on the toggle.
  2. **Stripe API Secret Key** – Your Stripe secret key that starts with sk_
  3. **Stripe Publishable Key** – Your Stripe publishable key that starts with pk_

Note

You can activate and deactivate your app by setting the variable **ShowCreditCard**.

PayPal

With **Braintree**, you can seamlessly accept both **PayPal** and **credit cards** with a single integration. Your customers will never even need to leave your website’s checkout page to make purchases using PayPal! Instead, they will click a PayPal button – designed exclusively for Braintree merchants – and enter their PayPal credentials in a new window or lightbox.

PayPal Account:

- Login to PayPal, then click **here**
- Set your app name, and email address then click **Create App**.
- On the top right side, you can see two tabs, SandBox and Live, Choose **Live**.
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find PayPal and edit the following options:
  1. Enable **PayPal Payment Method** by clicking on the toggle.
  2. **PayPal Mode** – Choose Live (In case your app is in Live mode).
  3. **PayPal Client ID** – PayPal application ID you created in the previous chapter.

4. **PayPal Secret Key** – PayPal application secret key you created in the previous chapter.

Braintree Gateway Account:

- Login to **Braintree gateway** with PayPal.
- Create **API Keys** from the drop-down menu.
- then go to **Processing** Options to enable customers to pay with **PayPal** with:
  - PayPal Email
  - PayPal Client Id
  - PayPal Client Secret
- Create or Find **Merchant Account ID** when clicking on **Business** from the drop-down menu & copy **Account ID** in your **AppSettings.cs** class file in your solution code.
- then again go to **API Keys** from the drop-down menu, and Click the **Keys** tab & copy the `Tokenization Keys` in your **AppSettings.cs** class file in your solution code as below:

For more details, you can see the Images:

Note

You can activate and deactivate your app by setting the variable **ShowPaypal**.

CashFree

- Go to the Cashfree website & sign in to your account with a registered email address.
- Click on **Activate** button to activate various payment gateways for your account.
- Now, you can click on **View Dashboard** to view account details.
- Go to the **Credentials** tab to access the test API credentials for payment gateways.
- Click on the **Get Credentials** tab to access the sandbox credentials
- View app ID & secret key under the **Test Credentials** tab.
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find CashFree and edit the following options:
  1. Enable **CashFree Payment Method** by clicking on the toggle.
  2. **CashFree Mode** – Choose Live (In case your app is in Live mode).
  3. **Client ID** – CashFree application ID you created in the previous chapter.
  4. **Client Secret** – CashFree application secret key you created in the previous chapter.

- You can specify the type of **base currency in Cashfree** by setting the variable
  and here you can check what **currency is accepted**

Note

You can activate and deactivate your app by setting the variable **ShowCashFree**.

RazorPay

- Go to the Razorpay website and Sign Up.
- Enter your work email address and a password for your Razorpay account and click **Create Account**.
- Fill Pre-sign Up **Form**
- Verify Email Address
- Once your account is created, you have access to the **Test mode** on the **Dashboard**. Test mode is used for testing purposes and does not involve actual money transactions. However, you would need to **activate your account** in order to accept live payments.
- To create an **Application**, follow this RazorPay Guide.
- Once your application was created, take the required keys then go to `Admin Panel -> Payments & Ads -> Payment Configuration` find RazorPay and edit the following options:
  1. Enable the **RazorPay Payment Method** by clicking on the toggle.
  2. **Application ID** – RazorPay application ID you created in the previous chapter.
  3. **Application Secret** – RazorPay application secret key you created in the previous chapter.
- Finally, go to the file **analytic.xml** in the project and add the app id.

- You can specify the type of **base currency in RazorPay** by setting the variable
  and here you can check what **currency is accepted**

Note

- You can activate and deactivate your app by setting the variable **ShowRazorPay**.

PayStack

- Sign Up with Paystack
- Verify your sign-up using a verification link sent to your email.
- The next thing is to sign in to your new **Paystack account**
- On your **Dashboard,** you will find your public and secret key
- Grab the keys, then go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find PayStack and edit the following options:
  1. Enable the **PayStack Payment Method** by clicking on the toggle.
  2. **Secret Key** – PayStack application secret key you created in the previous chapter.

Note

You can activate and deactivate your app by setting the variable **ShowPayStack**.

PaySera

- Sign up with PaySera then Log in to your account. In the menu on the left choose `Settings > Profile Settings > Service management > Change`.
- Select **Collection of payments online via e-banking and other systems** and click on **Order**
- In the left menu please select **Projects and Activities > My projects** and click on **Add a new project**.
- In the opened window, enter data of a new project: In the section **Service provider** enter your online domain name.
- If you are not using any platform, select I will enter the project address manually and enter the website address.
- If you are using a platform, then in the section I will use a platform, a list of platforms will be provided and, after selecting one of them, the address used will be entered automatically.
- In the section **Service description** briefly describe the services you provide. If you are selling physical goods, enter the refund policy and delivery terms. More information on the description of the project is provided here.
- You can change the information and service description later as needed: Enter the current phone number and email address in the section Contact details. If required, also specify other contact details, e.g. Skype. These contact details will be provided to your buyers together with a payment confirmation letter. The project ownership confirmation code is provided in the section Project confirmation; you should paste it in your website's head section. This way we will be sure that you are the real owner of the website. Having completed the requested information, click on Save project.
- Make sure that the website already contains purchase and sale, delivery and refund terms, privacy policy, and your contact details as a merchant. Submit the project for review only when the website has been fully prepared to be used by clients, the payment platform has been integrated and test payments have been performed. Your submitted project will be reviewed within one day.
- After creating a project, you will be provided with the data necessary for the technical integration of the payment platform
    1. In the menu on the left under **Projects and Activities > My projects** you will find the project ID (Project number/projectid);
    2. In the menu on the left under **Projects and Activities > My projects** click on **Project settings > General project settings** and you will find the code required to confirm ownership of your website;
    3. In the menu on the left under **Projects and Activities > My projects** click on **Project settings > General project** settings and you will find your project password.
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find PaySera and edit the following options:
    1. Enable **PaySera Payment Method** by clicking on the toggle.
    2. **PaySera Mode** – Choose Live (In case your app is in Live mode).
    3. **Project ID** – PaySera project ID you created in the previous chapter.
    4. **Account Password** – Your PaySera project password.

Note

You can activate and deactivate your app by setting the variable **ShowPaySera**.

IyziPay

- Create an account in Iyzipay
- You can access the **API and Security Key** values, where you can test the services from the **Settings** menu, without the need for email activation
- Grab the keys then go to `Admin Panel -> Payments & Ads -> Payment Configuration` paste the keys in Iyzipay form and fill the rest of the infromation.

Note

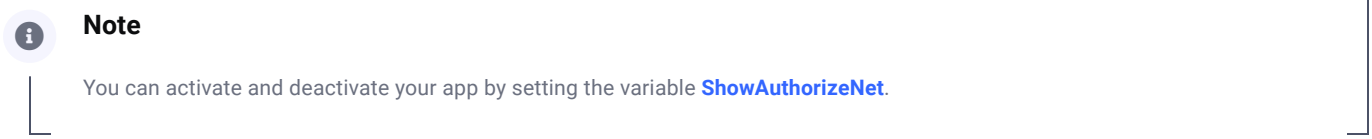
You can activate and deactivate your app by setting the variable **ShowIyziPay**.

2CheckOut

- Go to: https://secure.2checkout.com/cpanel/ and Sign in with your 2Checkout live account or create a new one.
- Now click on **Integrations** -> **Webhooks & API** and from the API section, copy **Merchant Code**, **Publishable Key,** and **Private Key**
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find 2checkout and edit the following options:
  1. Enable the **2checkout Payment Method** by clicking on the toggle.
  2. **2checkout Mode** – Choose Live (In case your app is in Live mode).
  3. **Seller ID** – Your Merchant ID.
  4. **Publishable Key** – Publishable Key you copied in the previous chapter.
  5. **Private Key** – Private Key you copied in the previous chapter.

Authorize.Net

- Log into the Merchant Interface.
- Click **Account** from the main toolbar.
- Click **Settings** in the main left-side menu.
- Click **API Credentials & Keys**.
- Select **New Transaction Key**.
- Click **Submit to continue**.
- Request and enter the PIN for verification.
- Your new Transaction Key is displayed.
- If the Disable Old Transaction Key Immediately box is not checked, the old Transaction Key will automatically expire in 24 hours. When the box is checked, the Transaction Key expires immediately.
- Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find Authorize.net and edit the following options:
  1. Enable **Authorize.net Payment Method** by clicking on the toggle.
  2. **Authorize.net Mode** – Choose Live (In case your app is in Live mode).
  3. **Authorize.Net API LOGIN ID** – Your API Login ID.
  4. **Authorize.Net TRANSACTION KEY** – TRANSACTION KEY you copied in the previous chapter.



SecurionPay

- You can find a detailed doc here: https://securionpay.helpjuice.com/90032-backoffice/529474-api-keys
- Once you got your keys, Go to
  `Admin Panel -> Payments & Ads -> Payment Configuration`, find Securionpay, and edit the following options:
  1. Enable **Securionpay Payment Method** by clicking on the toggle.
  2. **Securionpay Public key** – Your Public key you copied in the previous chapter.
  3. **Securionpay Secret key** – Secret Key you copied in the previous chapter.

Note

You can activate and deactivate your app by setting the variable **ShowSecurionPay**.

Yoomoney

- How to get your secret key: https://yookassa.ru/docs/support/merchant/payments/implement/keys?lang=en
- Once you got your keys, Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find Yoomoney, and edit the following options:
  1. Enable **Yoomoney Payment Method** by clicking on the toggle.
  2. **Yoomoney Wallet ID** – Your wallet ID.
  3. **Secret key** – Secret Key you copied in the previous chapter.

Note

You can activate and deactivate your app by setting the variable **ShowYoomoney**.

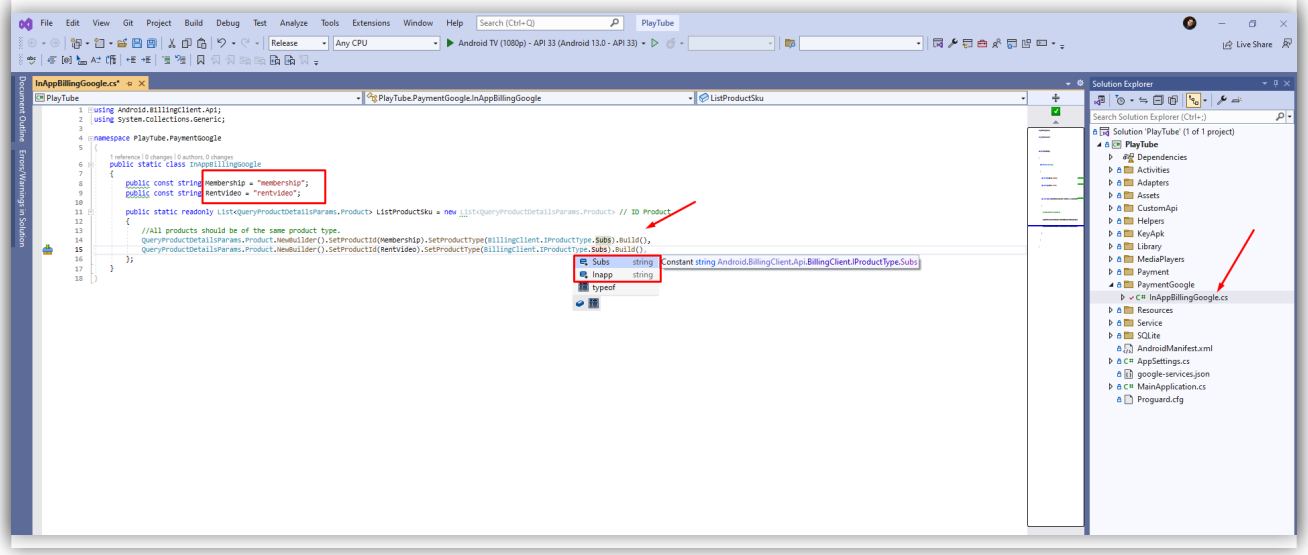
In-App Billing Purchases

This method is only available for versions with an Extended License.

In order to use the Google in-app purchase payment system in your mobile application, you will need to follow a few steps in the google play console as there are a few rules you should know.

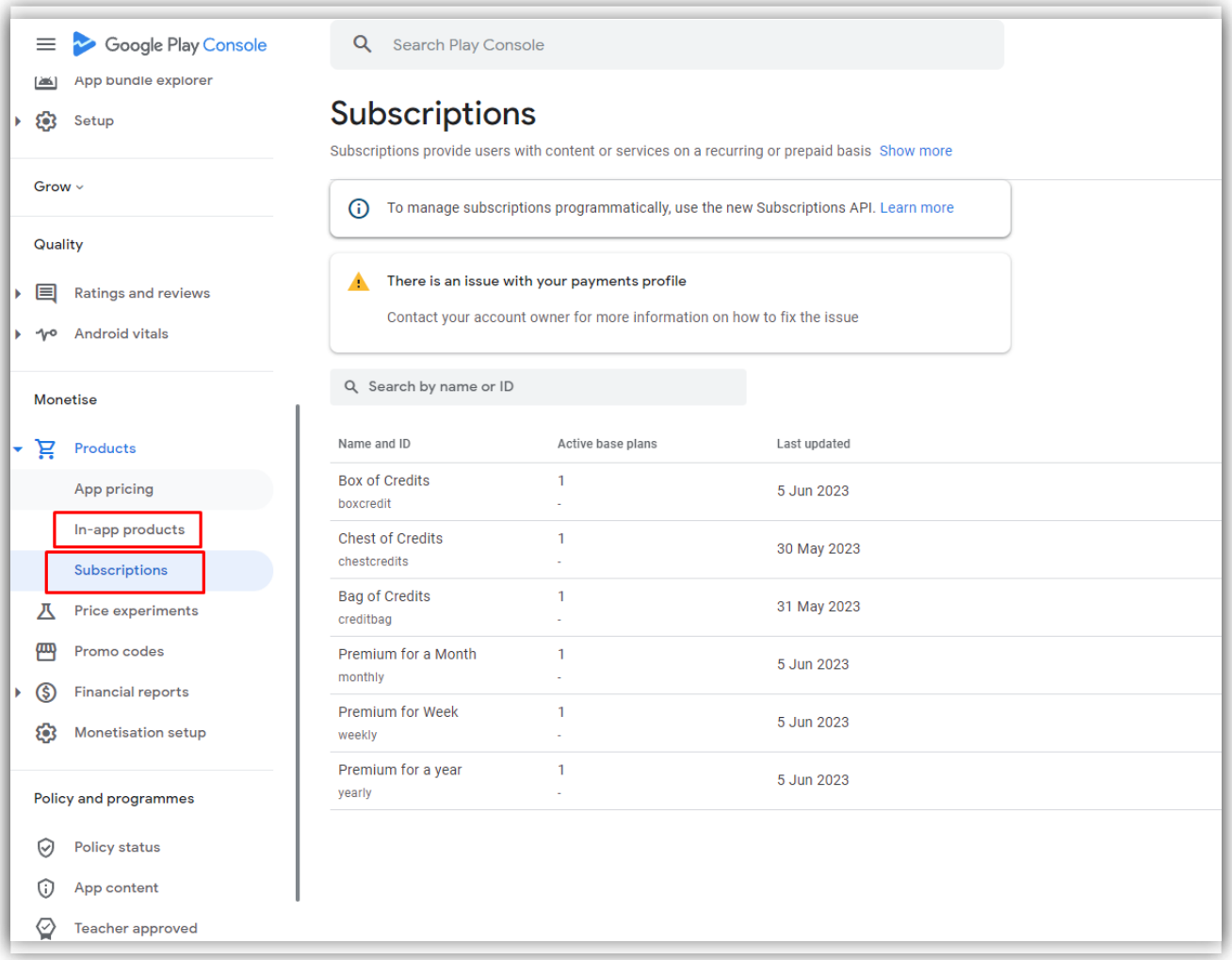
There should be a temporary APK file Uploaded to Google play in order to fetch the Product ID (In-app License code) so you can proceed with your Build and package prices.

You can select what is the type for your Products ” In-App – Subs ” and just one type is can selected.



When creating a new product, you must use the same Product ID that you wrote in your app.

You must add the product in only one place either In-app products or Subscriptions.



Note!

Please contact the ***Support Team*** in order to obtain the key to activate the extended version.

AamarPay

- After creating account on the Aamarpay
- Once you got your keys, Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find Configure Aamarpay Payment Method, and edit the following options:
  1. Select **Test mode** Choose the mode your application is using, for testing use the SandBox mode **Test** or **Live** .
  2. **Aamarpay Store Id** – Your Aamarpay Store Id.
  3. **Aamarpay Signature Key** – Your Aamarpay Signature Key.

> **Note**
>
> You can activate and deactivate your app by setting the variable **ShowAamarPay**.

FlutterWave

- After creating an account on the FlutterWave
- Once you got your keys from here then Go to `Admin Panel -> Payments & Ads -> Payment Configuration`, find Flutter Wave payment, and edit the following options:

  1. Enable **Flutter Wave Payment Method** by clicking on the toggle.
  2. **Flutter Wave API Secret Key** – Your Flutter Wave API Secret Key.
  3. Add **Public Key** and **Encryption Key** on the file **AppSettings.cs** in the App:

> **Note**
>
> You can activate and deactivate your app by setting the variable **ShowFlutterWave**.

Manage Features

Articles

- Register System
- Video Live Streaming
- General Application Settings
- Picture To Picture Mode

Register System

- You can set Disable/Enable **Register system** of the app by the variable below to True or False.

- If you want the date of the **Phone Number** field to appear on the register page, you can by the variable below

Video Live Streaming

- Sign up here to create an Agora account. After sign-up, you can log in here.
- Once you finish the sign-up process, you can create an Agora project in Agora Console, To create an Agora project, do the following:
  1. Enter the Project Management page.
  2. Click **Create**.
  3. Follow the on-screen instructions to enter a project name and use case, and check **Secured mode: APP ID + Token (Recommended)** as the authentication mechanism.
  4. Click Submit. You can now see the project on the Project Management page.
- To copy the **App ID**, find your project on the Project Management page in Agora Console, and click the copy icon to the right of the App ID.
- To get an App Certificate, do the following:
  1. On the Project Management page, click the edit icon for the project you want to use.
  2. Click the copy icon under Primary Certificate.
- Generate a Customer ID and Customer Secret
  1. In Agora Console, click the account name in the top right corner, and click **RESTful API** from the drop-down list to enter the **RESTful API** page.
  2. Click **Add a secret**, and click **OK**. A set of Customer ID and Secret is generated.
  3. Click **Download** in the **Customer Secret** column. Read the pop-up window carefully, and save the downloaded `key_and_secret.txt` file in a secure location.
- Go to `Admin Panel -> Settings -> Posts Settings -> Setup Live Streaming -> Agora API Configuration`
- Edit the following options:
  1. **App ID** – Your APP ID we got one previous chapter
  2. **App Certificate** – Your App Certificate we got one previous chapter
  3. **Customer ID** – Enter the Customer ID you downloaded from **key_and_secret.txt**.
  4. **Customer Secret** – Enter the Customer ID you downloaded from **key_and_secret.txt**
- Finally, go to the file **AppSettings.cs** in the project and add the **app id**.

> **Note**
>
> You can activate and deactivate your app by setting the variable **ShowGoLive**

General Application Settings

From your settings class on your PlayTube project, you will be able to control most of your Android **AppSettings**.

Database Name:

You can set the database name by changing the value of DatabaseName. It should always be the same name as your application name. Make sure that the name does not contain any spaces and that it has all small letters

Trending Settings:

Set the variable below to True or False To enable sections

Library Settings:

You can activate or disable library features in the application via the following variables

Import & Upload Videos:

Set the variable below to True or False To enable sections

- You can change the style of Player View by the variables with 2 options:
  - `Radius`
  - `Square`

Search View:

You can add a list of some keywords when opening the search page by the variable below

Channel Users:

Set the variable below to True or False To enable sections

Settings View:

Set the variable below to True or False to give the user ability to set his settings from his mobile or to display sections

Picture To Picture Mode

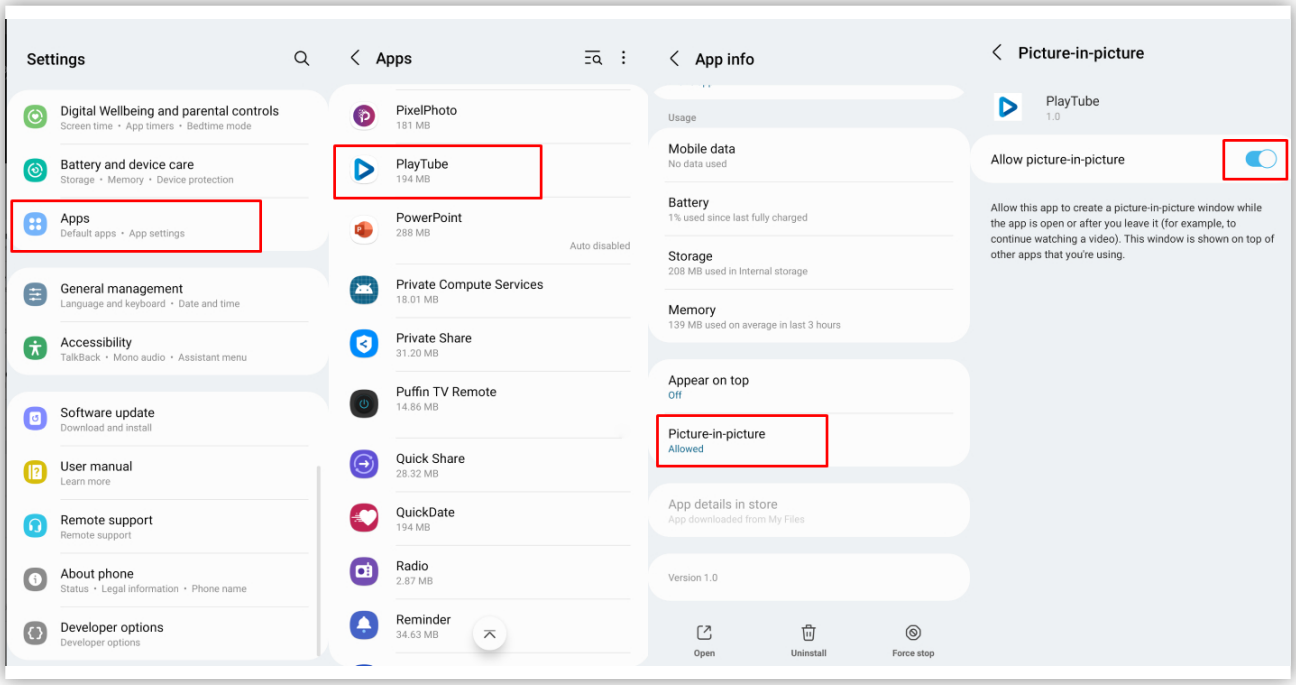
Picture-in-picture (PiP) shrinks a video into a small player so you can keep watching while using other apps on your mobile device. You can move the small player around your device's home screen and position it over other apps.

To use picture-in-picture (PiP), exit the **PlayTube** app while a video is playing. If you have the PiP setting turned on, the video will shrink into a PiP window. The PiP window can be dragged to different parts of the screen, allowing playback to continue on top of other apps. You can pause or stop playing a video before you exit PlayTube to prevent PiP from turning on.

- You can activate and deactivate according to the PiP on your **AppSettings.cs** class file in your solution code as below.

Turning picture-in-picture OFF/ON

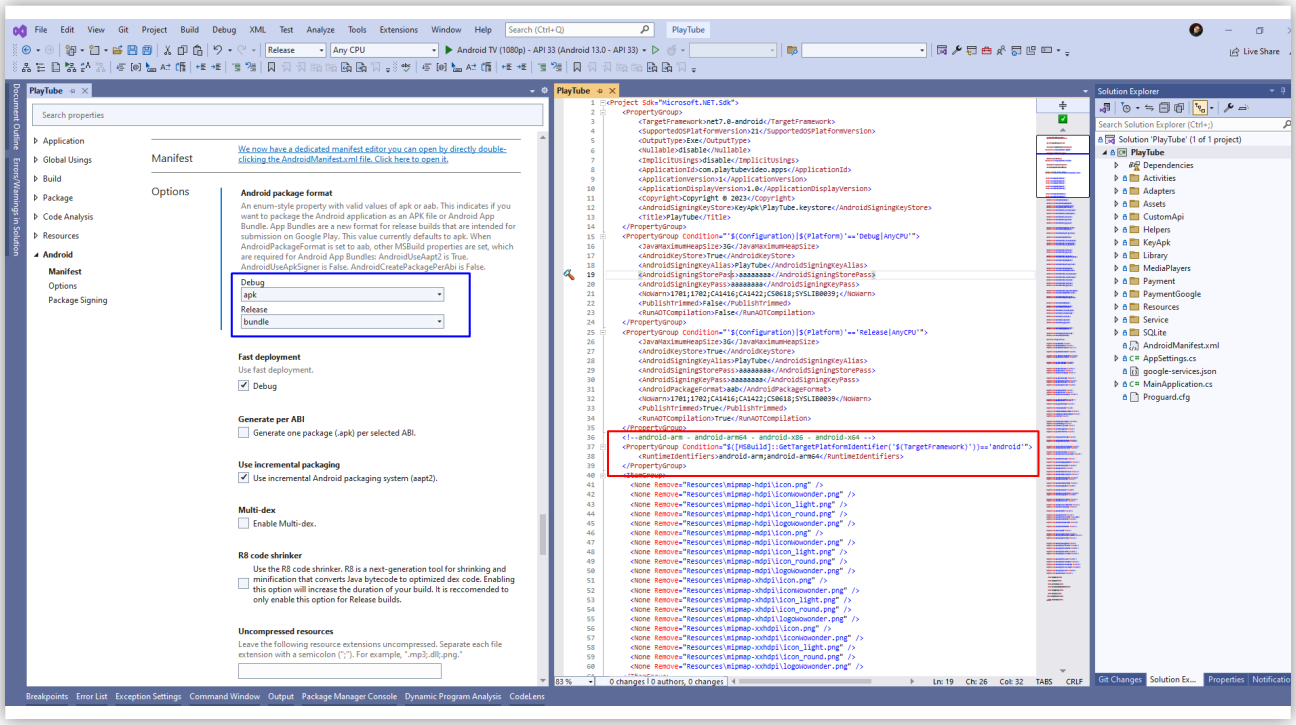
1. Go to your `Android settings > Apps > Picture-in-picture.`
2. Tap **PlayTube**.
3. To turn it on, tap Allow picture-in-picture.
4. Go to your PlayTube app settings.
5. Toggle Picture-in-Picture to off/on.

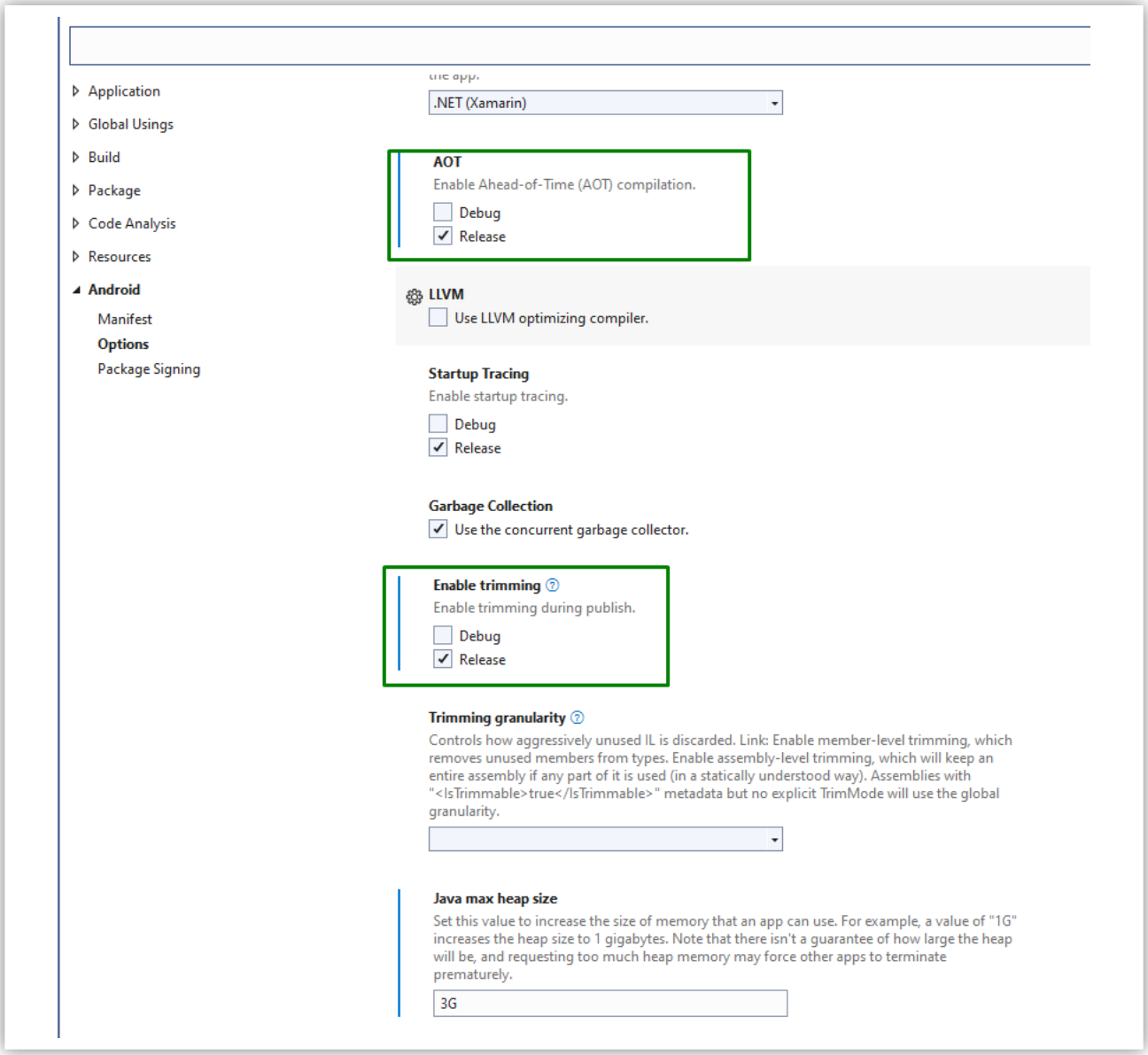


Publish The Application

After you have customized all the options on the app, the following screenshot is to see how your **configuration** should be before the archiving process.

1. **The red** color is very important.
2. **The blue** color is to enable support for Android App Bundles, you'll need to opt-in to the **bundle value** of the Android Package Format property within your Android project options. Before you do this, ensure you change your project to a Release configuration as app bundles are intended for release packages only.
3. **The green** color Is optional when selecting AOT and Enable Startup Tracing the splash screen will be loading faster by 50% but the app size will increase as well by 25% so it's your choice and it's up to you.


PlayTube
Application
Global Usings
Build
Package
Code Analysis
Resources
Android
Manifest
Options
Package Signing
Manifest
We now have a dedicated manifest editor you can open by directly double-clicking the AndroidManifest.xml file. Click here to open it.
Options
Android package format
An enum-style property with valid values of apk or aab. This indicates if you want to package the Android application as an APK file or Android App Bundle. App Bundles are a new format for release builds that are intended for submission on Google Play. This value currently defaults to apk. When AndroidPackageFormat is set to aab, other MSBuild properties are set, which are required for Android App Bundles: AndroidUseAapt2 is True. AndroidUseApkSigner is False. AndroidCreatePackagePerAbi is False.
Debug
apk
Release
bundle
Fast deployment
Use fast deployment.
Debug
Generate per ABI
Generate one package (.apk) per selected ABI.
Use incremental packaging
Use incremental Android packaging system (aapt2).
Multi-dex
Enable Multi-dex.
R8 code shrinker
Use the R8 code shrinker. R8 is a next-generation tool for shrinking and minification that converts Java bytecode to optimized dex code. Enabling this option will increase the duration of your build. It is reccomended to only enable this option for Release builds.
Uncompressed resources
Leave the following resource extensions uncompressed. Separate each file extension with a semicolon (";"). For example, ".mp3;.dll;.png."
Solution Explorer
Solution 'PlayTube' (1 of 1 project)
PlayTube
Dependencies
Activities
Adapters
Assets
CustomApi
Helpers
KeyApk
Library
MediaPlayers
Payment
PaymentGoogle
Resources
Service
SQLite
AndroidManifest.xml
AppSettings.cs
google-services.json
MainApplication.cs
Proguard.cfg


Application
Global Usings
Build
Package
Code Analysis
Resources
Android
Manifest
Options
Package Signing
.NET (Xamarin)
AOT
Enable Ahead-of-Time (AOT) compilation.
Debug
Release
LLVM
Use LLVM optimizing compiler.
Startup Tracing
Enable startup tracing.
Debug
Release
Garbage Collection
Use the concurrent garbage collector.
Enable trimming
Enable trimming during publish.
Debug
Release
Trimming granularity
Controls how aggressively unused IL is discarded. Link: Enable member-level trimming, which removes unused members from types. Enable assembly-level trimming, which will keep an entire assembly if any part of it is used (in a statically understood way). Assemblies with "<IsTrimmable>true</IsTrimmable>" metadata but no explicit TrimMode will use the global granularity.
Java max heap size
Set this value to increase the size of memory that an app can use. For example, a value of "1G" increases the heap size to 1 gigabytes. Note that there isn't a guarantee of how large the heap will be, and requesting too much heap memory may force other apps to terminate prematurely.
3G

Note

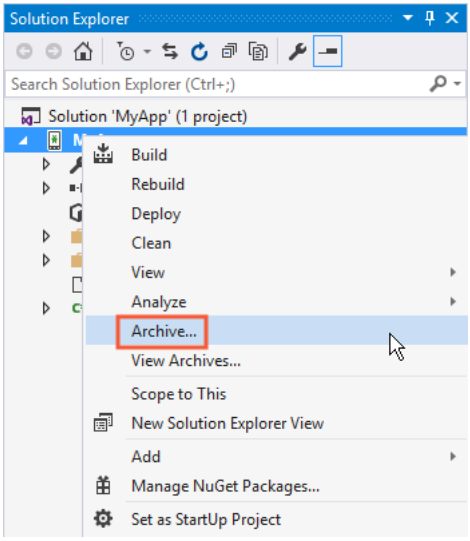
Be sure you set the **SetApisReportMode** variable in AppSettings.cs to **false** before releasing the app.

Articles

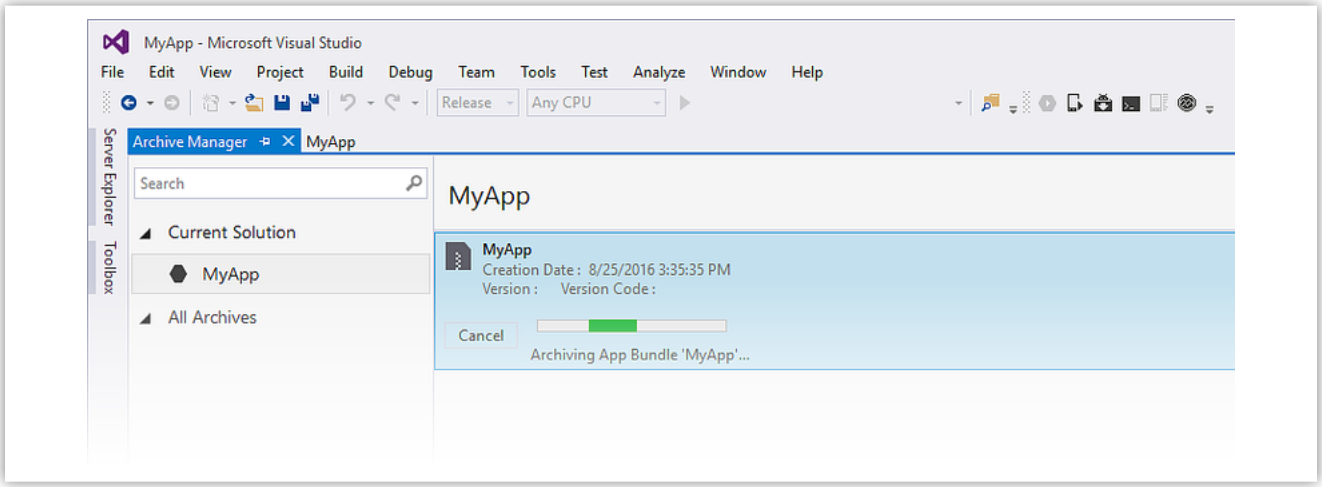
- Archive For Publishing
- Reduce The APK Size

Archive For Publishing

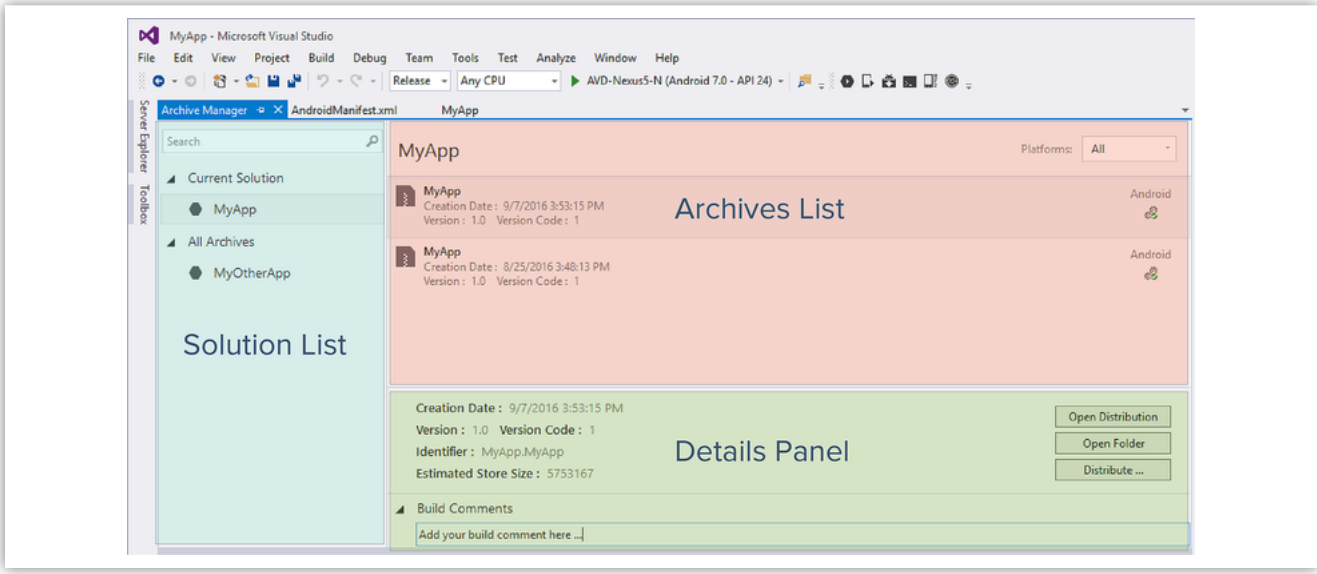
- To begin the publishing process, right-click the project in **Solution Explorer** and select the **Archive...** context menu item.



- **Archive...** launches the **Archive Manager** and begins the process of archiving the App bundle as shown in this screenshot:



- The **Archive Manager** is comprised of a **Solution List** pane, an **Archives List**, and a **Details Panel**:



The **Solution List** displays all solutions having at least one archived project. The **Solution List** includes the following sections:

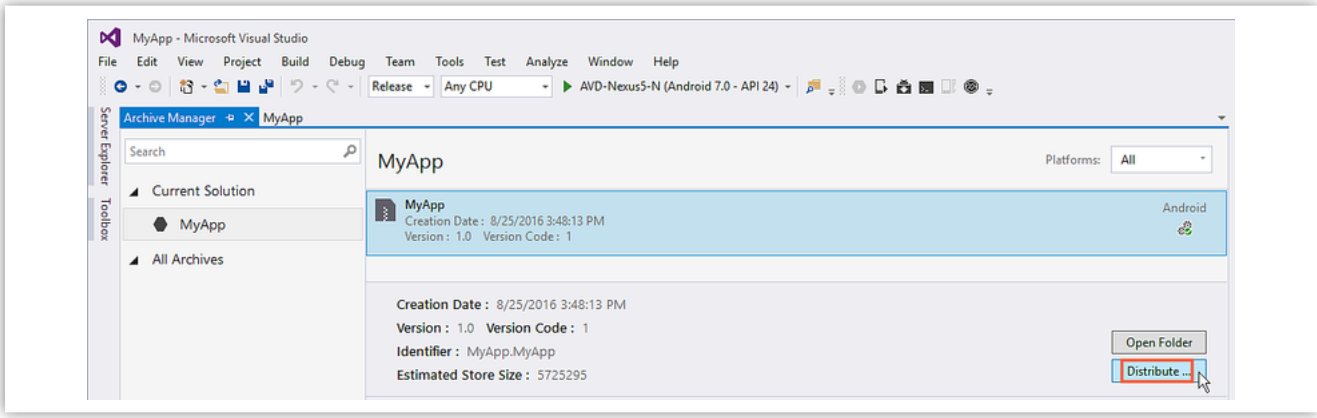
- **Current Solution** – Displays the current solution. Note that this area may be empty if the current solution does not have an existing archive.
- **All Archives** – Displays all solutions that have an archive.
- **Search** text box (at the top) – Filters the solutions listed in the **All Archives** list according to the search string entered in the text box.

The **Archives List** displays the list of all archives for the selected solution. The **Archives List** includes the following sections:

- **Selected solution name** – Displays the name of the solution selected in the **Solution List**. All information shown in the **Archives List** refers to this selected solution.
- **Platforms Filter** – This field makes it possible to filter archives by platform type (such as iOS or Android).
- **Archive Items** – List of archives for the selected solution. Each item in this list includes the project name, creation date, and platform. It can also show additional information such as the progress when an item is being archived or published.

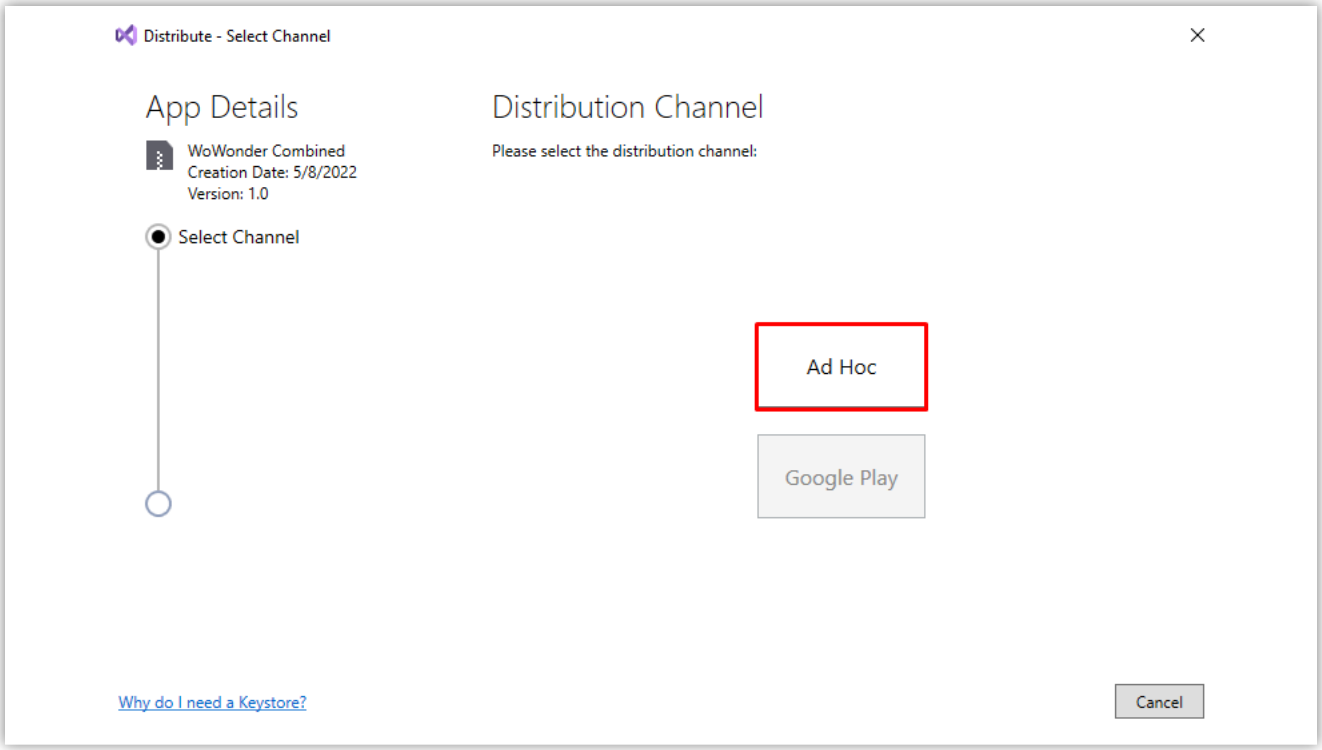
The **Details Panel** displays additional information about each archive. It also allows the user to start the Distribution workflow or open the folder where the distribution has been created. The **Build Comments** section makes it possible to include build comments in the archive.

- When an archived version of the application is ready to publish, select the archive in the **Archive Manager** and click the **Distribute…** button



- The **Distribution Channel** dialog shows information about the app, an indication of distribution workflow progress, and a choice of distribution channels. On the first run, two choices are presented:
  1. **Ad-Hoc** – Saves a signed APK to disk that can be sideloaded to Android devices. Continue to Sign the App Package to learn how to create an Android signing identity, create a new signing certificate for Android applications, and publish an *ad hoc* version of the app to disk. This is a good way to create an APK for testing.

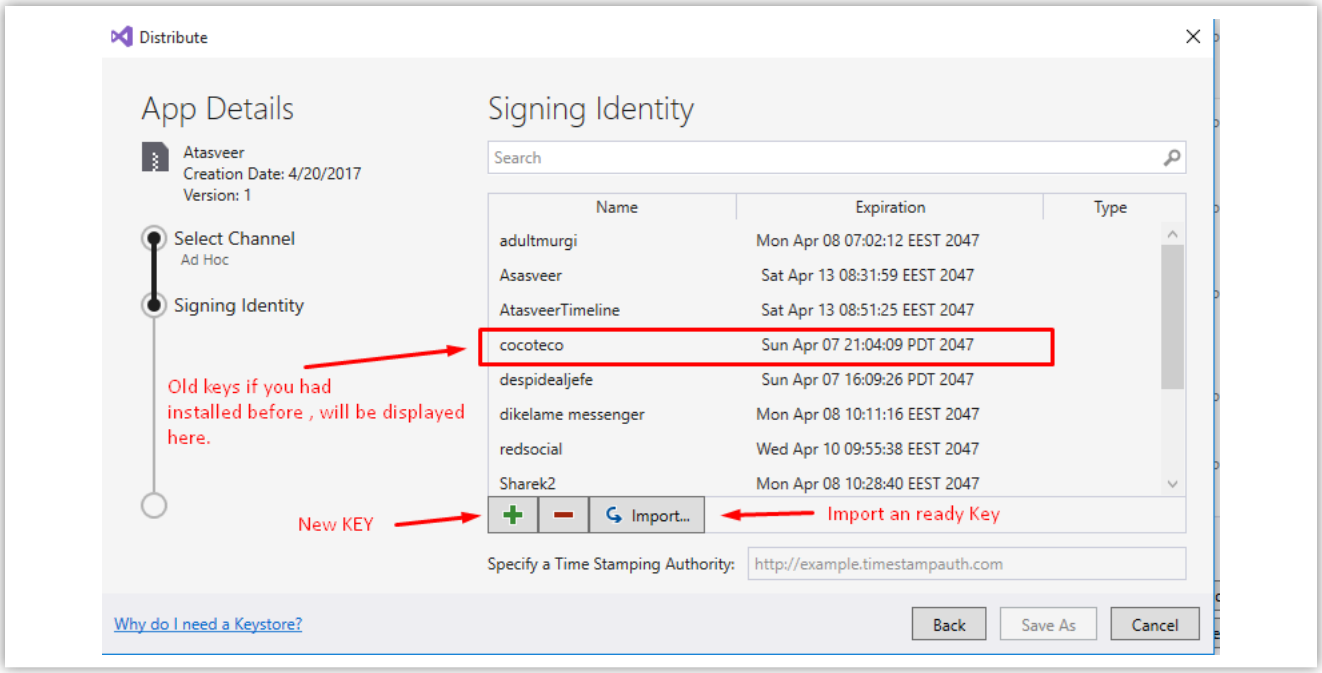
2. **Google Play** – Publishes a signed APK to Google Play. Continue to publish to Google Play to learn how to sign and publish an APK in the Google Play store.



- After **Ad-Hoc** is selected, Visual Studio opens the Signing Identity page of the dialog as shown in the next screenshot. To publish the. APK must first be signed with a signing key (also referred to as a certificate). An existing certificate can be used by clicking the Import button and then proceeding to Sign the APK.
  Otherwise, click click the + button to create a new certificate:

Create a New Certificate

After Ad-Hoc is selected, Visual Studio opens the Signing Identity page of the dialog as shown in the next screenshot. To publish the. APK must first be signed with a signing key (also referred to as a certificate). An existing certificate can be used by clicking the Import button and then proceeding to Sign the APK. Otherwise, click click the + button to create a new certificate:



The Create Android Key Store dialog is displayed, use this dialog to create a new signing certificate that can use for signing Android applications. Enter the required information (outlined in red) as shown in this dialog:

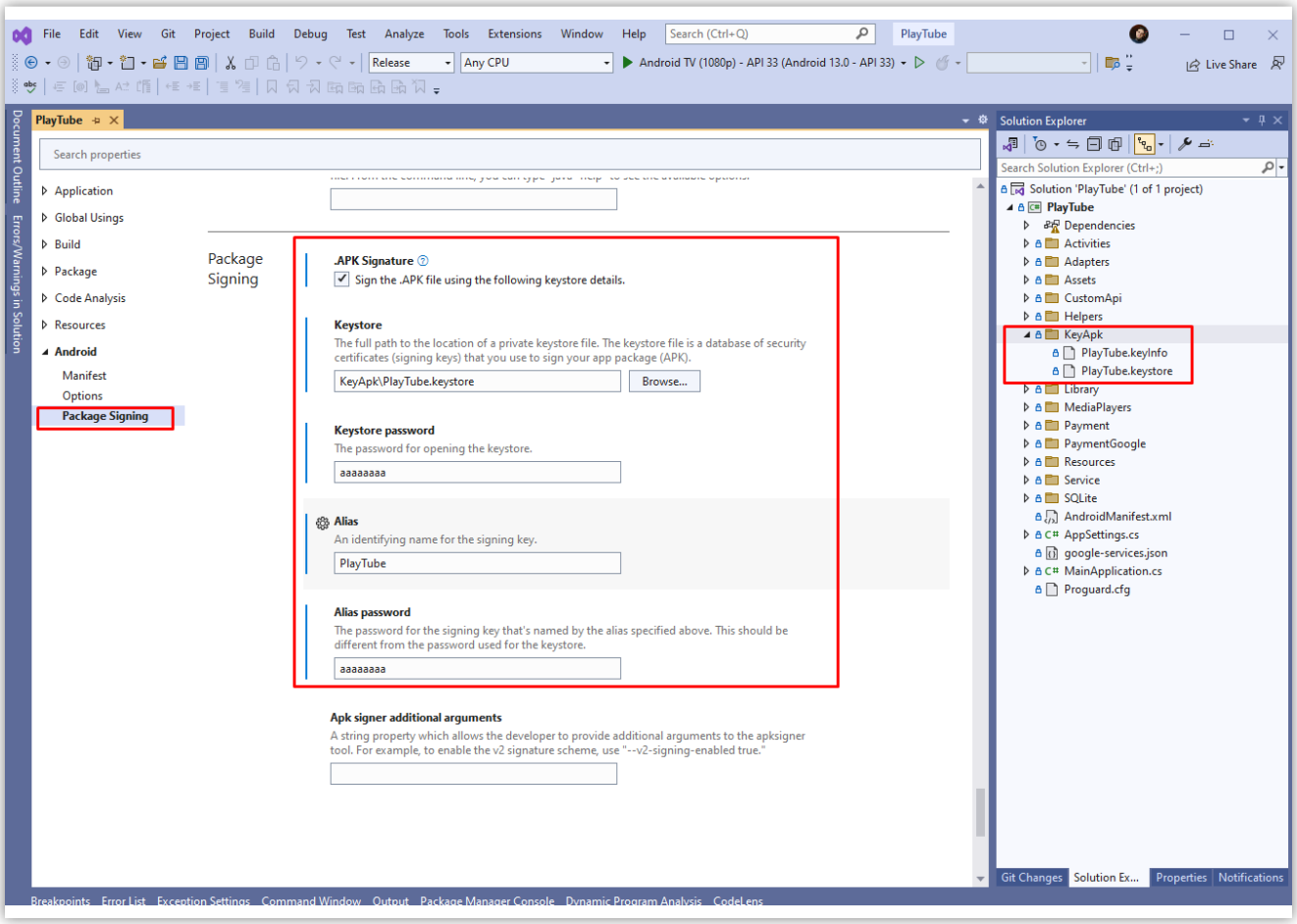
The resulting Keystore resides in the following location: (Save Them)

```
C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\alias\alias.Keystore
```

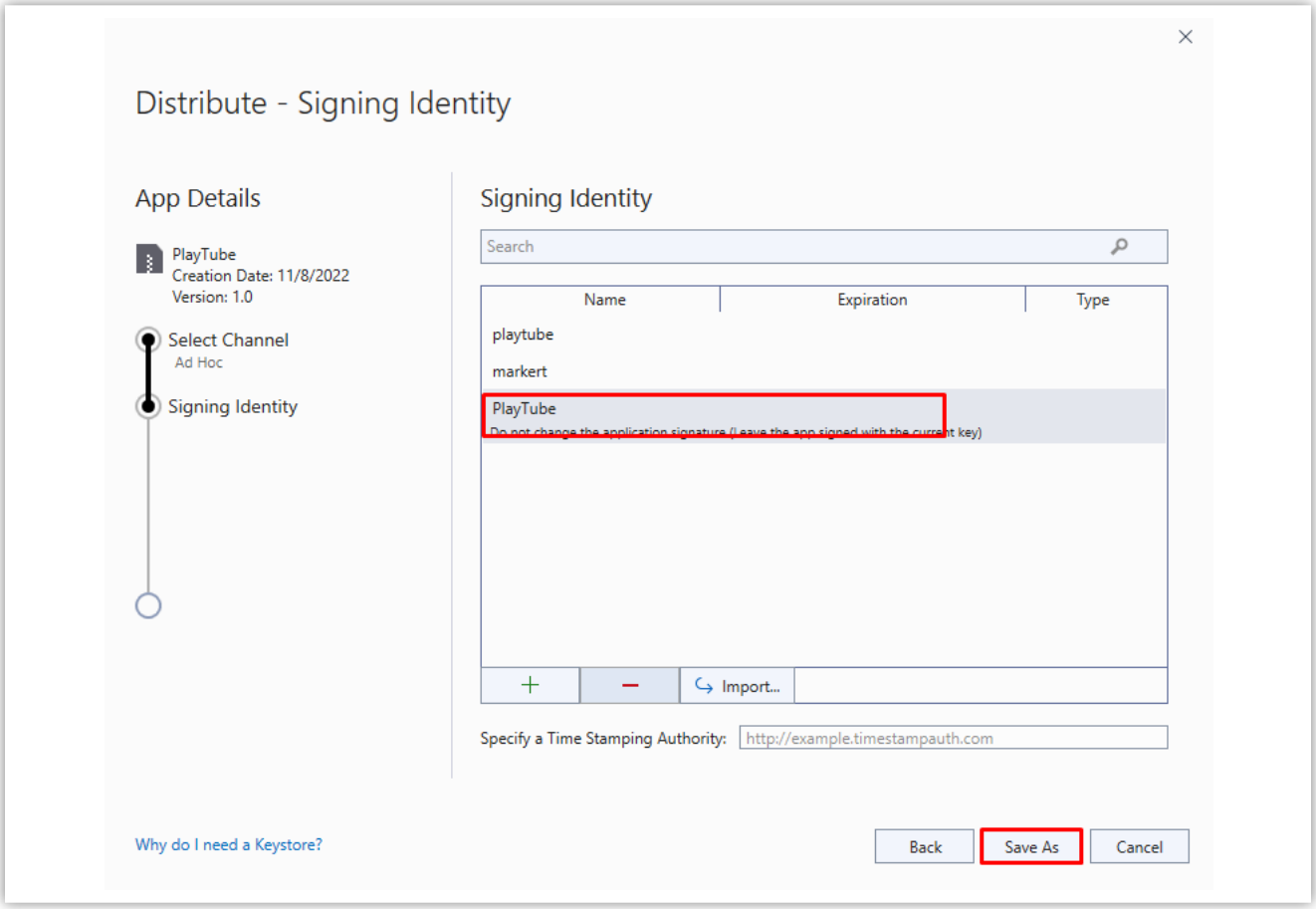
For example, the above steps might create a new signing key in the following location:

```
C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\chimp\chimp.Keystore
```

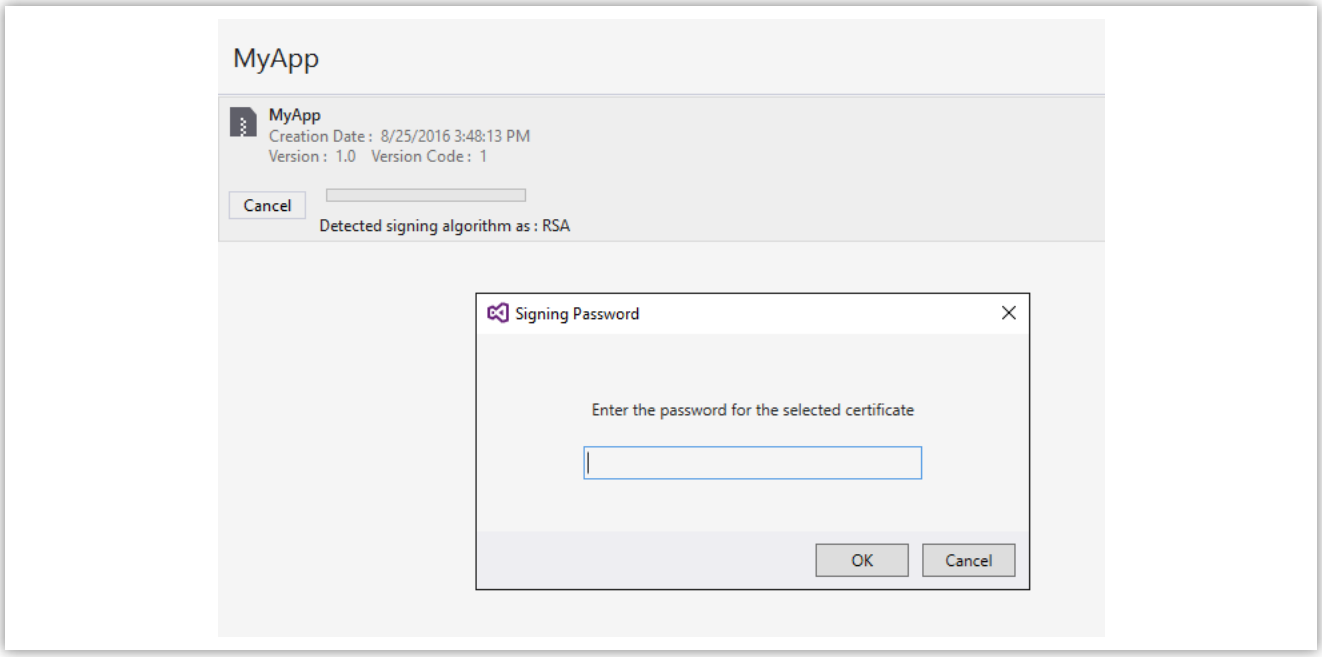
When Create is clicked, a new key store (containing a new certificate) will be saved and listed under Signing Identity, and you should add the key file to the project on the folder KeyApk as shown in the next screenshot:



To publish an app on Google Play, click Cancel and go to Part 3. To publish ad-hoc, select the signing identity to use for signing and click Save As to publish the app for independent distribution. For example, the chimp signing identity (created earlier) is selected in this screenshot:



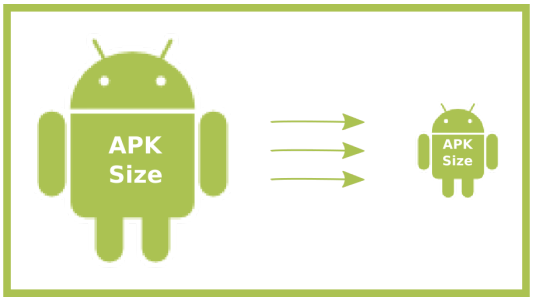
Next, the Archive Manager displays the publishing progress. When the publishing process completes, the Save As dialog opens to ask for a location where the generated. APK file is to be stored:



Now you are ready to submit your application to **Google Play**.

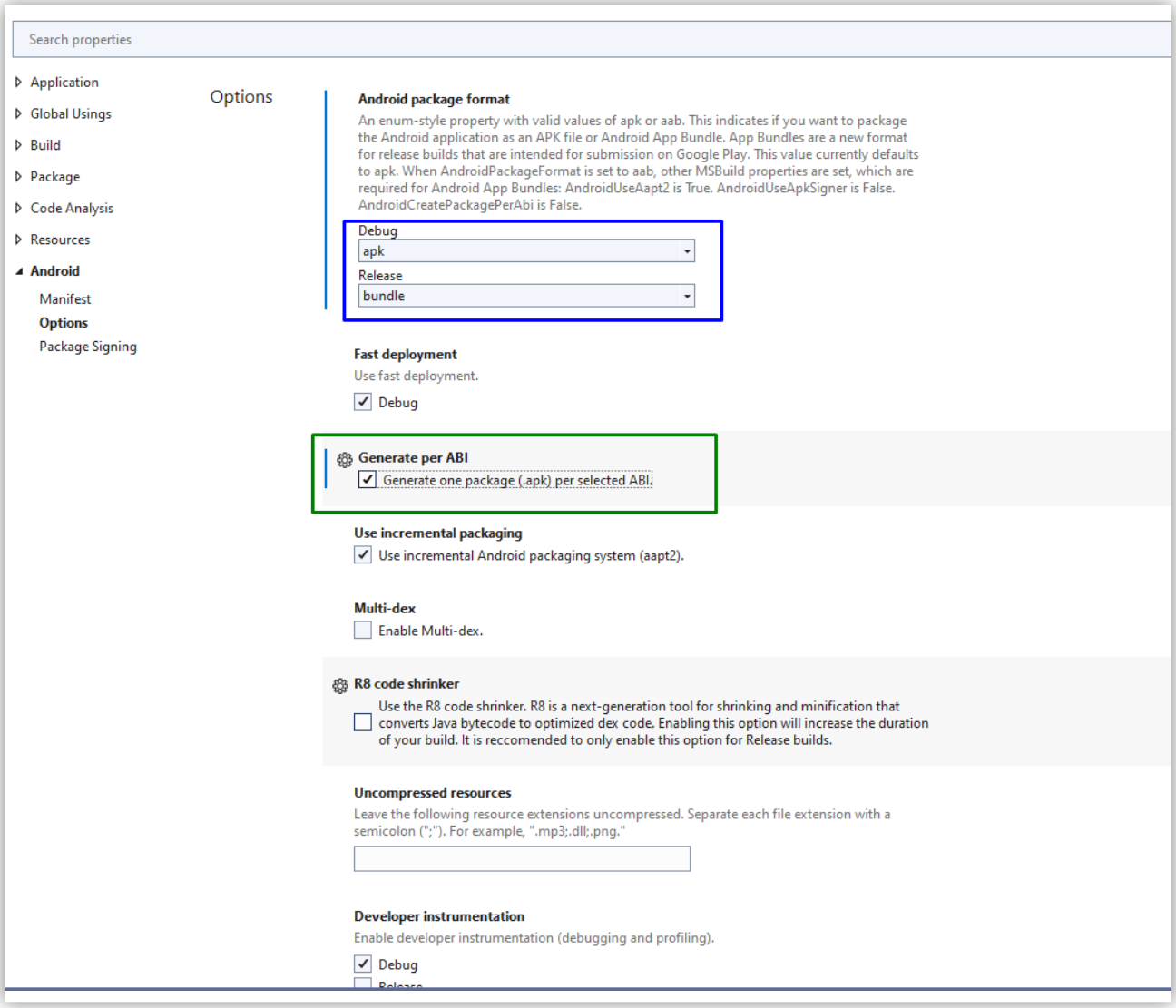
Reduce The APK Size

Users often avoid downloading apps that seem too large, This article describes how to reduce your APK size, which enables more users to download your app.

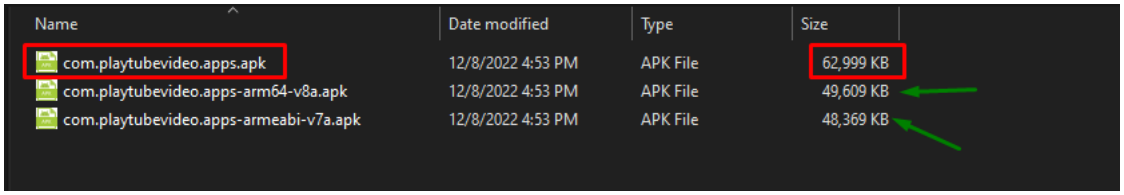


You often build a single APK to support all your target devices whenever possible, which might result in a very large APK due to files needed to support multiple screen densities or Application Binary Interfaces (ABIs). One way to reduce the size of your APK is to create multiple APKs that contain files for specific screen densities or ABIs. Gradle can create separate APKs that contain only code and resources specific to each density or ABI. To learn how to configure your build to generate multiple APKs follow this link https://developer.android.com/studio/build/configure-apk-splits.html

- Follow the picture below and your Android option should be the same as the image below Do Not select the green color frame to leave it unchecked



- After selecting Generate Package Per ABI and after the archive process is done open your Installed APK folder and you will see as below



(Generate Package Per ABI) Install type

Now Your APK is about 63 MB without the Google compressor once you submit it to the google play console it will be 48 MB to 49 MB.

Usually, you submit the red framed archive as the image below which is 63MB
instead of submitting it, you can submit 2 APKs to google play by uploading the Archives which are pointed by the **Green** arrow and is 49MB in size.

Frequently Asked Questions (FAQs)

Supposed to be commonly asked issues, questions & troubleshoot tricks are in this section.

Articles

- How To Set Custom API
- The App Crashes When You Open It
- Set Up Device For Development
- SHA-1 Fingerprint
- Report Mode

How To Set Custom API

In the event that there is any addition or modification to a new connection with API and you need information such as the `Website URL, ServerKey, UserId, and AccessToken`, you can use the class **CustomApiModel.cs** that contains this information.

When you want to use API with connection type "**POST**" you can see this example:

When you want to use API with connection type "**GET**" you can see this example:

How I can have custom work for this application?

Once You decide to customize your application and change many things or add something new #ScriptSun is your best Freelancing opportunity, Turn Your ideas into reality.
Open a custom work ticket Here.

The App Crashes When You Open It

We always ensure that the latest version of the application is used, and also in each new version, a new **CERT** encrypted key from the **DoughouzLight Checker** must be used.

Set Up Device For Development

This article will discuss how to setup an Android device and connect it to a computer so that the device may be used to run and debug **Xamarin.Android** applications.
By now, you've probably seen your great new application running on the Android emulator, and want to see it running on your shiny Android device. Here are the steps involved with connecting a device to a computer for debugging:

1. **Enable Debugging on the Device** – By default, it will not be possible to debug applications on an Android device.
2. **Install USB Drivers** – This step is not necessary for OS X computers. Windows computers may require the installation of USB drivers.
3. **Connect the Device to the Computer** – The final step involves connecting the device to the computer by either USB or WiFi.

Enable Debugging on the Device

Enable USB Debugging mode on Android 12L, 11, 10, Pie, Oreo, Nougat, Marshmallow, or any version using our easy steps guide. **Enable developer options** and USB debugging from a PC using this simple tutorial. USB Debugging (also called Developer mode or Android debug mode) is used for the connection between an Android device and your computer using Android SDK.

You can easily turn on the **Developer options** or the USB debugging on any Android smartphone including Samsung Galaxy phones, Google Pixel, Sony, Motorola, Xiaomi Redmi, Huawei, One Plus, Oppo, Vivo, HTC, Honor, Kindle fire, Asus, LG, Lenovo, Micromax and many more.

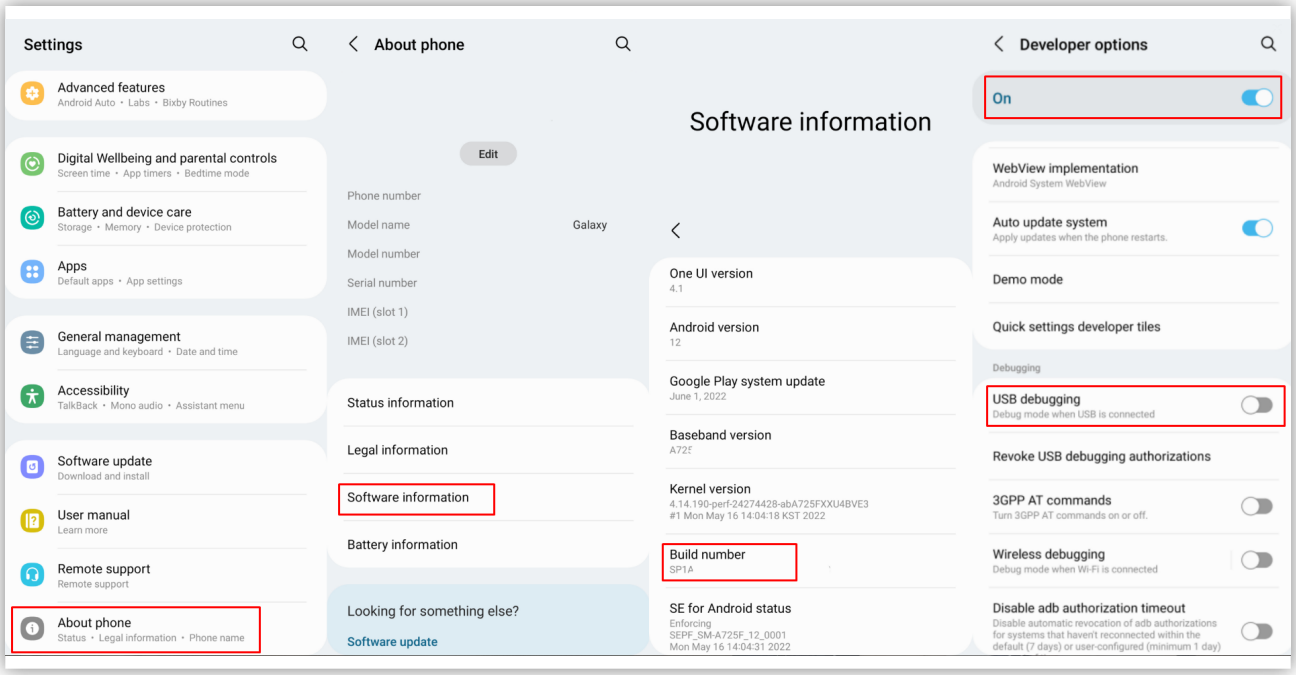
Using ADB command prompt window or a few tricks, you can turn on the USB debugging mode on broken screen phone, hard bricked phone, dead phone, FRP locked phone, password locked, black screen phone, boot-loop phone, etc. In this guide, we'll discuss the step-by-step procedure on how to enable USB debugging mode on any Android smartphone and tablet.

Android 9/10/11/12

USB debugging mode is available under the Developer options. But from Android 9 Pie, the developer options are hidden. So firstly you need to unlock it to access the debugging mode.

- Open the apps panel on your phone and launch the Settings app.
- On the next screen, scroll down to the bottom and open the System.
- In System, open the About Phone section.
- Under the About Phone section, look for the Build number.
- Just tap on Build number 7 times. It'll show you a countdown message saying, "You're 4,3,2,1 steps away from being a Developer".

- Tap on the Build number until it shows a message saying, ” You are now a developer”.
- As soon as you receive this message, the Developer options will get unlocked in your About Phone section.
- So go back to the About phone section, and click on the Advanced option.
- The advanced options will not be available on some phones, so those phones directly go to the next step.
- Look for the Developer options and open them.
- Under Developer options, you can find and enable USB debugging mode.



Install USB Drivers

This step is not necessary for OS X. Just connect the device to the Mac with a USB cable.
It may be necessary to install some extra drivers before a Windows computer will recognize an Android device connected by USB.

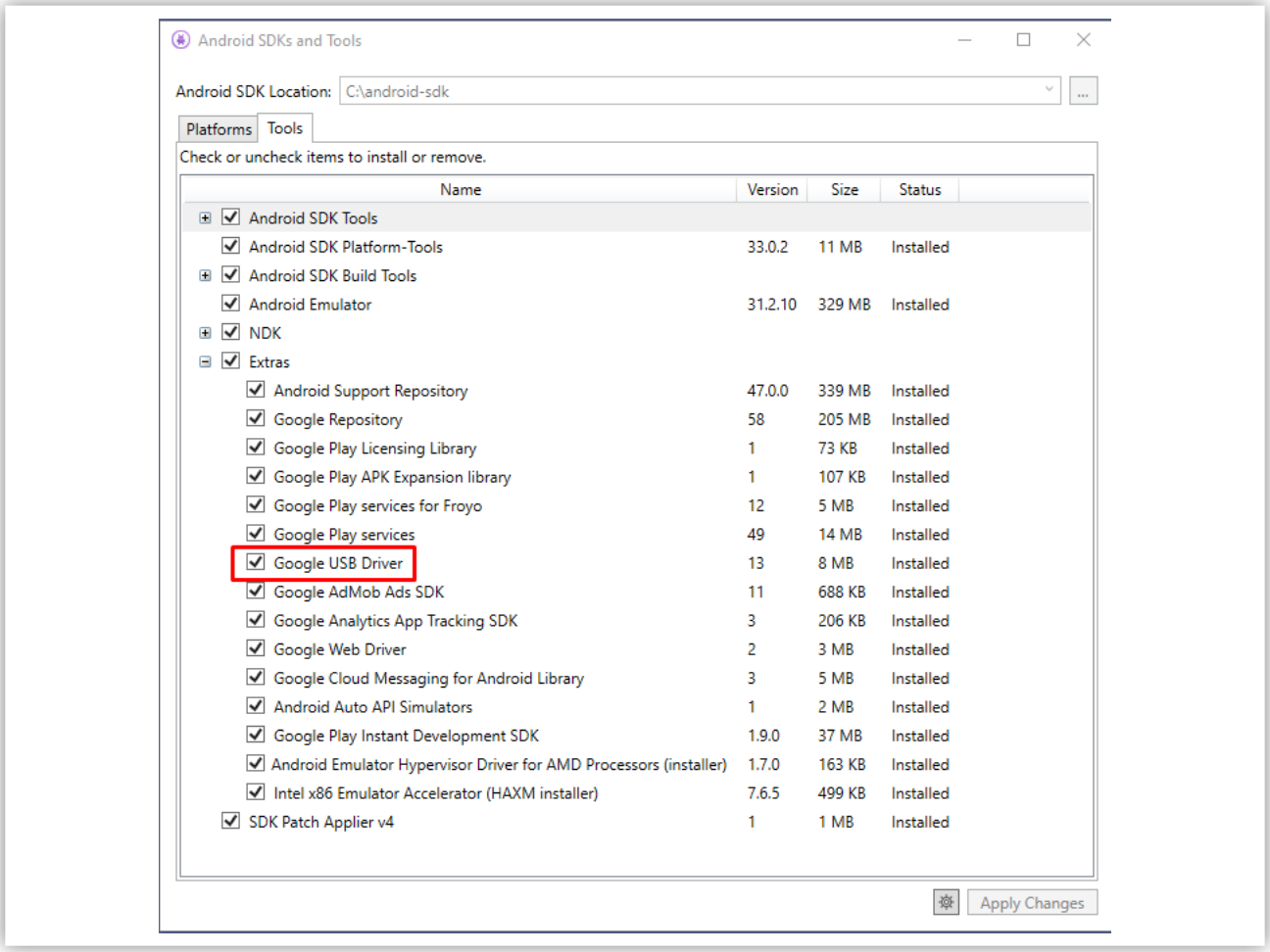
Info

These are the steps to set up a Google Nexus device and are provided as a reference. Steps for your specific device may vary but will follow a similar pattern. Search the internet for your device if you have trouble.

Download the USB Drivers

Google Nexus devices (with the exception of the Galaxy Nexus) require the Google USB Driver. The driver for the Galaxy Nexus is distributed by Samsung.

All other Android devices should use the USB driver from their respective manufacturer.
Install the Google USB Driver package by starting the Android SDK Manager, and expanding the Extras folder, as can be seen in the following screenshot:



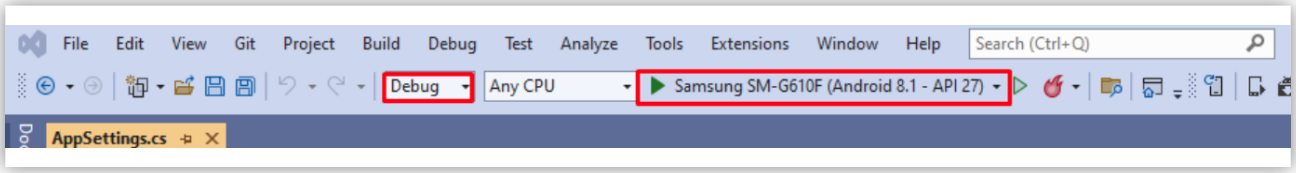
Check the Google USB Driver box, and click the Install button.

Connect the Device to the Computer

The final step is to connect the device to the computer. There are two ways to do so:

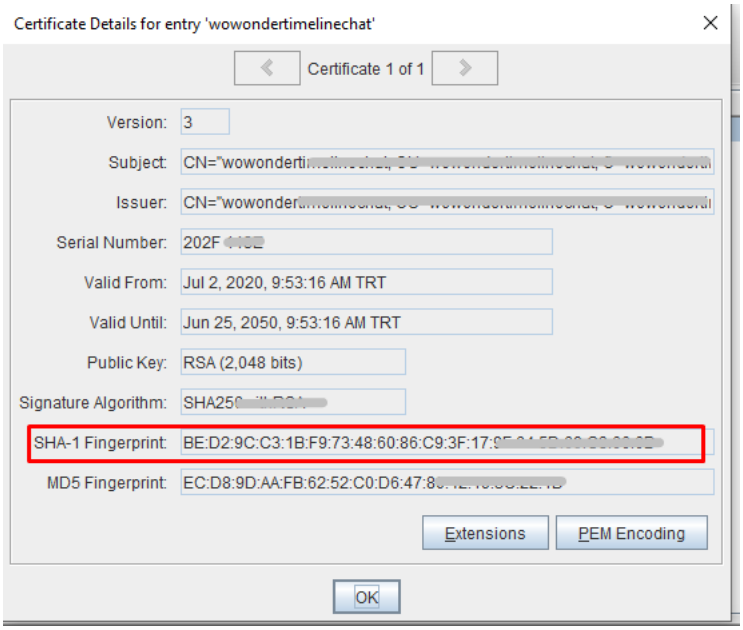
- **USB cable** – This is the easiest and most common way. Just plug the USB cable into the device and then into the computer.
- **WiFi** – It is possible to connect an Android device to a computer without using a USB cable, over WiFi. This technique requires a bit more effort but could be useful when there is no USB cable or the device is too far away for a USB cable. Connecting via WiFi will be covered Here.

Now Your device will appear in the Debug section click start and the application will be deployed to your own device and you can start your own test.



SHA-1 Fingerprint

To Display your SHA-1 Key You can use Portecle application then when to put your key and double click and copy the SHA-1 key hash from the dialog that you can use for Google maps integration or for social logins



Report Mode

How To Set your application Report Mode "API Testing"

You can set the variables below to True or False to start checking your API response from your server if the API fails you will get a message box in your application with the error of the API.

Don't forget to set the variable to **false** before releasing your app to google play.

Why I am getting such errors?

The reason behind the errors is simple so let's explain to you why you may face API problems.

EX: I cannot upload a profile picture. I cannot see any news feed. I press on like but it's not like on my website! all of these issues are miss behaving of the API from your server-side end.
Here are a few fixes you need to know.

- Your server side is using Cloudflare which should be disabled by default.
- Your Max Upload limit in your server is small it should be 150MB at least, for **more info**.
- Disable Mode_security on your server side, for **more info**.
- You are not using the last version of the PHP script.
- Your WoWonder PHP script API functions have missing files or are not updated correctly.
- Choose a good Hosting server such as **Ultahost** which installs all the plugins you need.

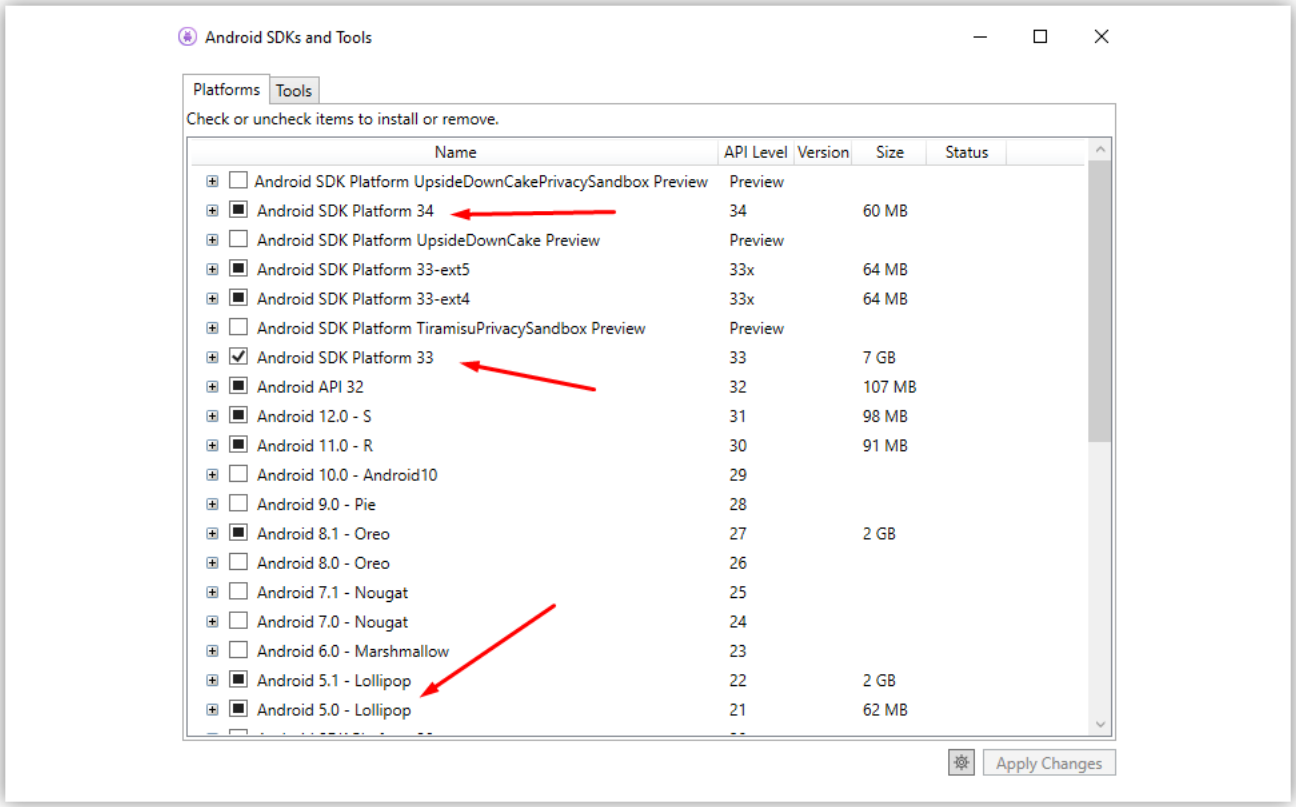
I set the report mode and I am getting an API issue so what's next?

If you are facing any API issue from your server-side you will need to contact the PHP support team from the link below **https://doughouzforest.ticksy.com** and show them your API response as a screenshot.

Errors Guide

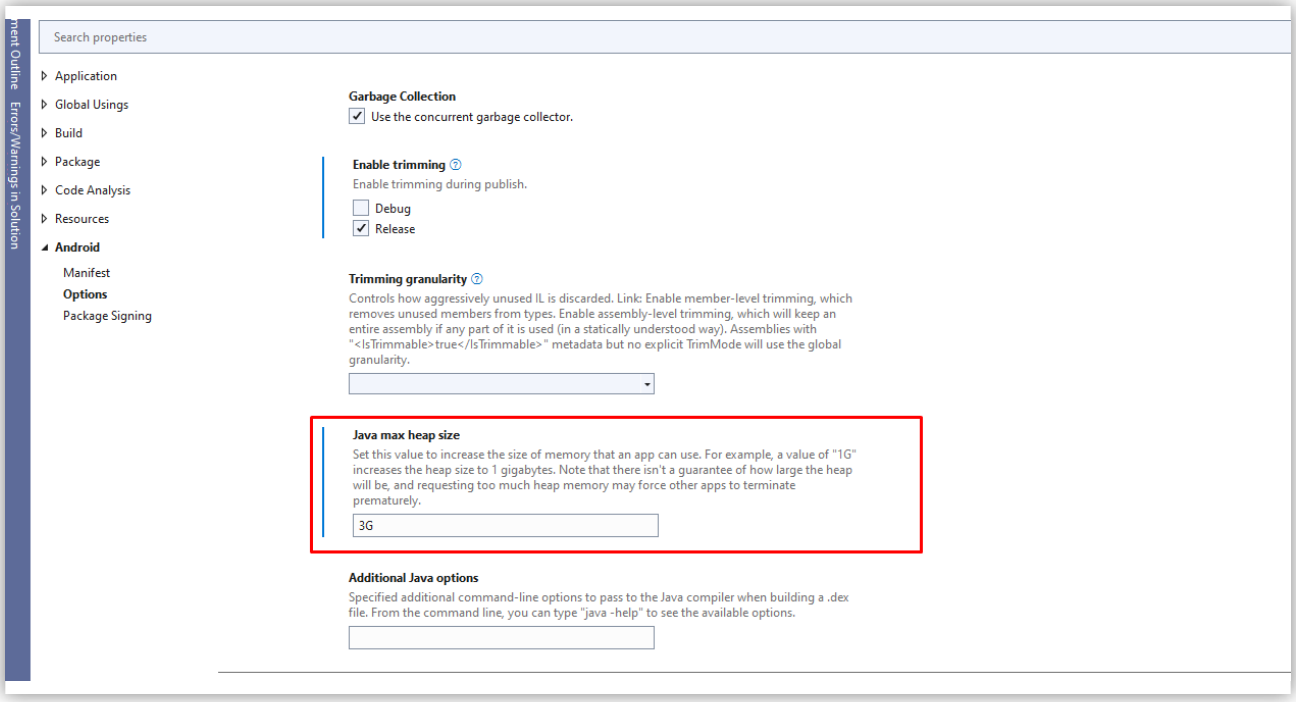
The file "obj\Debug\android\bin\packaged_resources" does not exist.

- From Visual Studio go to `"Tools -> Android -> Android SDK Manager"`. Once that opens, look in the "Extra's" folder for "Android Support Library". Make sure you have that checked, and if you do, make sure you have the latest. Update if necessary. Also, be sure you have installed all the SDKs from Android version 5.0 tell version 12 once you have selected all the APIs press on Install button, then accept terms.
- After you are sure that you installed everything that you need and the problem still exists Copy your PlayTube folder to a small path like `D://PlayTube` or `C://PlayTube` then open the solution and build the application, the path must be small and not long



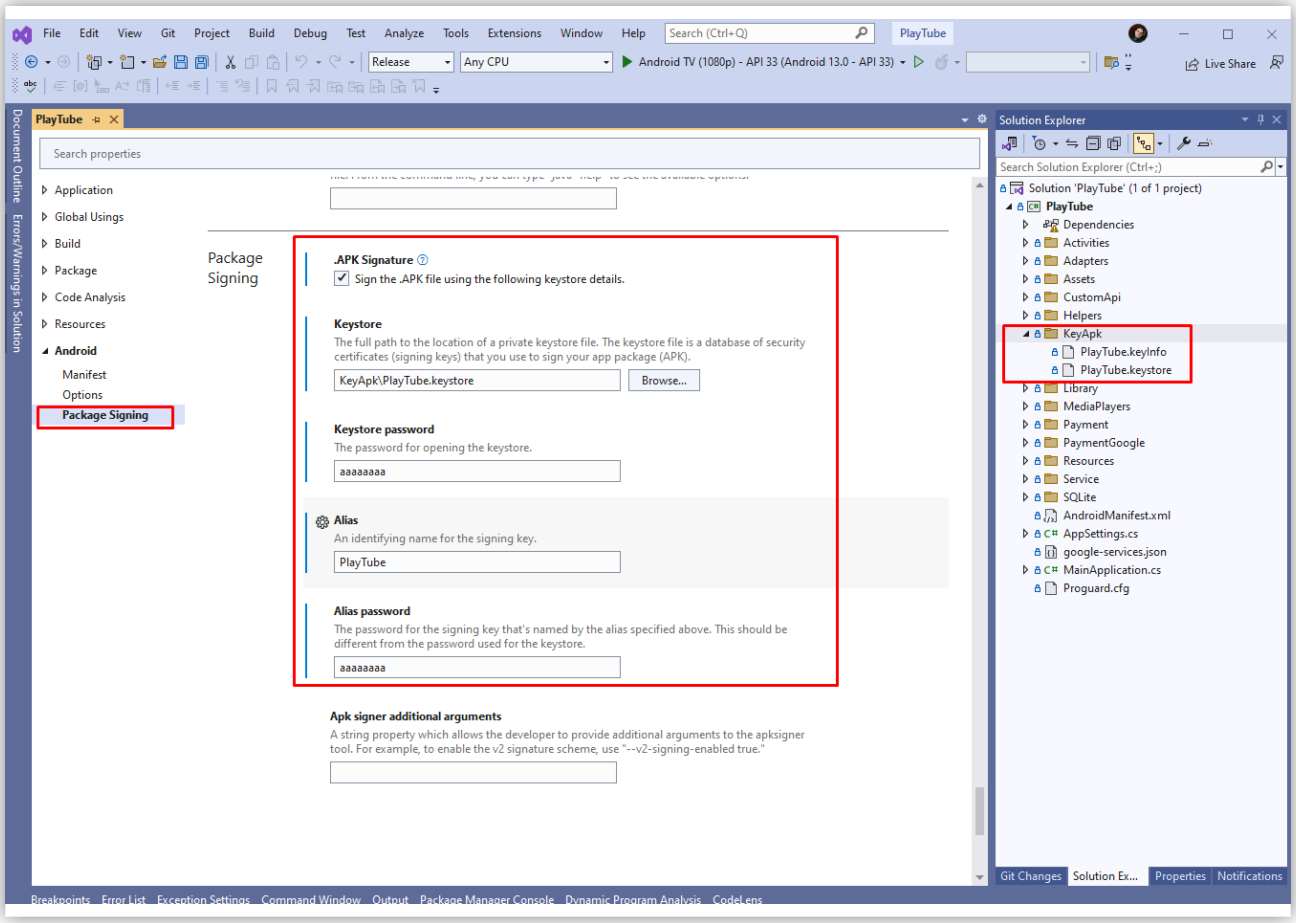
“java.exe” exited with code 1.

- Right-click on Project >> Click on Android properties >> Click on Android Option >> Select Advanced Button >> Set Java Max Heap Size to 3G (or less if needed 1G)



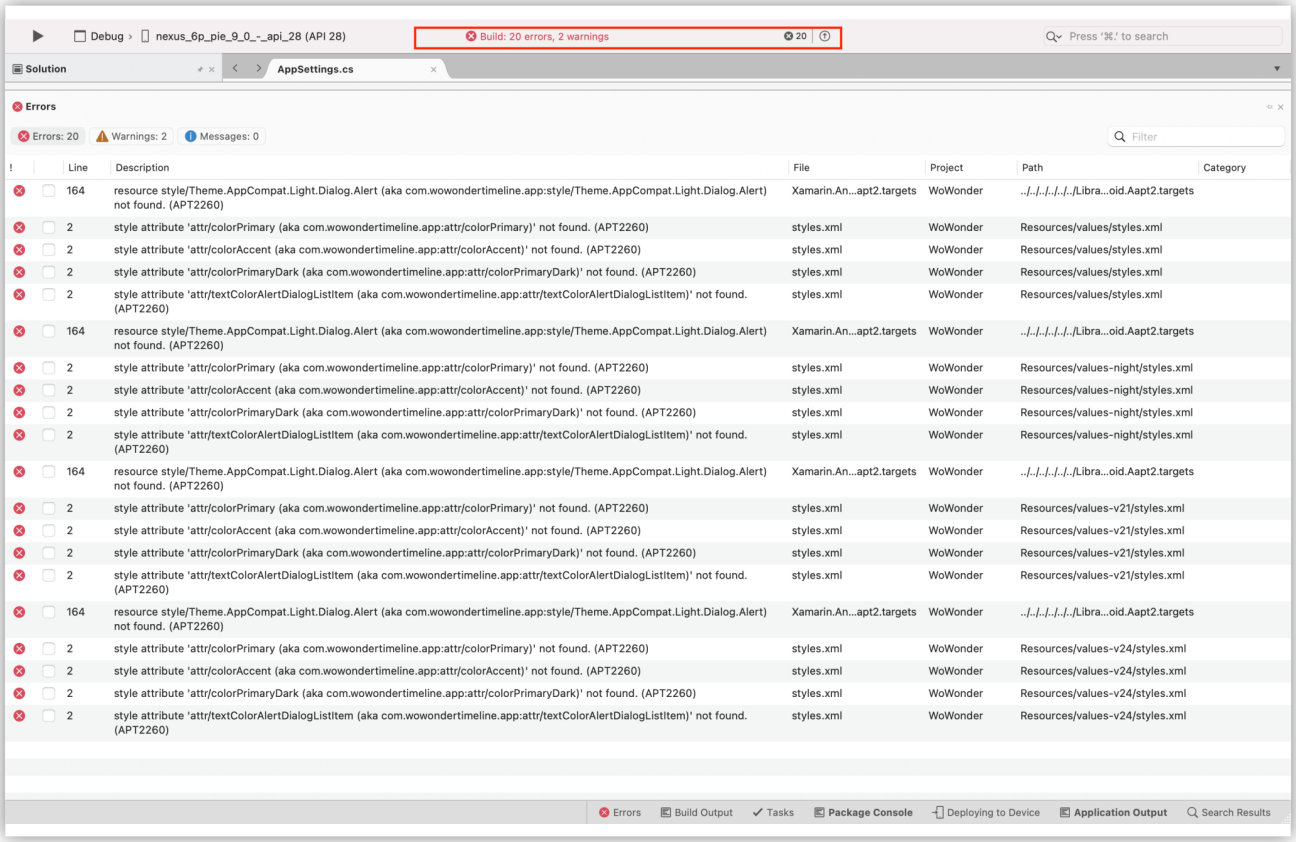
“java.exe” exited with code 2.

- You have the wrong information and authentication for your key
- Or you Uncheck the checkbox from your project properties page and rebuild your application (This suggestion is not recommended)



Facing a lot of errors in code

Dear customer, there is no syntax error at all in our code. If you are seeing errors, then that means the visual studio IntelliSense is not working properly. So the fix for that will be, to try clean and rebuild. If it did not work then, please delete the bin and obj folders. close visual studio and relaunch the VS. And give it some time and then do a clean rebuild. If it did not work, then please delete the current version. Restart your laptop, download the new version extract it on the desktop and then try to open it and give it some time to restore PKGs. And then try to rebuild it. Visual studio intelligence will start working.



This purchase code is used on the different domain names.

As a buyer of our item, you are allowed to use the item only on one domain, so if you have one license of the item, you can use it on one domain, not more.
If you need to use it on the second domain you will need to buy a new license for it.

I don't use the old domain anymore so what's next?

If your old domain is not used anymore, or you want to change the old domain to a new one you need to ask our support team for "**Domain Update**" by opening a new ticket and writing the new domain and the old domain and the purchase code, and our support team will take a look and update it for you as soon as they reach your ticket.

For information and assistance, you can contact us to **submit a ticket**.

Bypass the SSL Error

There are few things to do when you get the **SSL** error on your app

- Be sure you didn't add HTTPS:// in your domain URL if it basically doesn't support the SSL when you generated the cert key the first time. if it's added by mistake then regenerate the cert key and add your domain as below: http://mywebsite.com without the SSL. so basically don't write HTTPS:// if your website is HTTP://
- Try to go to set these 2 variables to true in `PlayTube >> AppSettings.cs`

Change Connection Type

Right-click on `Project >> Click on Android properties >> Click on Android Option >> Select Advanced Button` and set the connection type as below

HTTPClient Implementation	SSL/TLS Implementation	Server
Android	Native TLS 1.2+	This should be the Default
Default	Default (Native TLS 1.2+)	This should be the Secondery
Managed	Managed TLS 1.0+	This is the last option you have

You should build the app and run it on each connection type and check if it works.

Host Server

Go To Your Host Control Panel

1. Click the Cloudflare icon, located in the Domains section of your control panel.
2. Click the Disable button to disable Cloudflare. EX: For **Blue Host** / EX: For **Hostgator**
3. Disable your Mod-security on your Host server totally and don't use it.

Note

If the problem still exists that means your main host is using a cloud system which prevents the application to connect to your website over SSL.

1. Be sure you are not using a cloud host server or your mode-security is enabled you may need to contact your host provider to disable it totally for you.
2. Change your host to a normal Host VPS or Detected server >> We recommend you to use the UltaHost.com Server which is 100% compatible with our application >> **UltaHost**

Help and Support

Faced a problem? Need assistance with the product? No worries – our customer support team is always ready to help you.

- Support requests are being processed on business days from 9:00 to 18:00 (GMT +05.30) [generally] within 24h to 48h in the order they were received.
- We suggest, while our team reviews your support request, please read the documentation that comes in the zip file of Codecanyon. You can download it from ThemeForest: https://codecanyon.net/downloads
- We are in GMT+5:30 timezone. We address all the support queries 6 days weekly (Sunday off).
- If any support ticket has no response from the item owner for 7 days, the ticket will be considered closed. If you need further assistance you can create another ticket or drop us an email asking to re-open the ticket for you.
- Have pre-sales questions or concerns, please write to us via our **website contact page**
- If you like our product and support then please drop a rate and write a review at: **https://codecanyon.net/downloads**

Note:

- We have launched our support portal on Desky. Please raise a ticket **here**

- Join our **Official Facebook Forum** for more discussion and hints, bugs, and fixes.