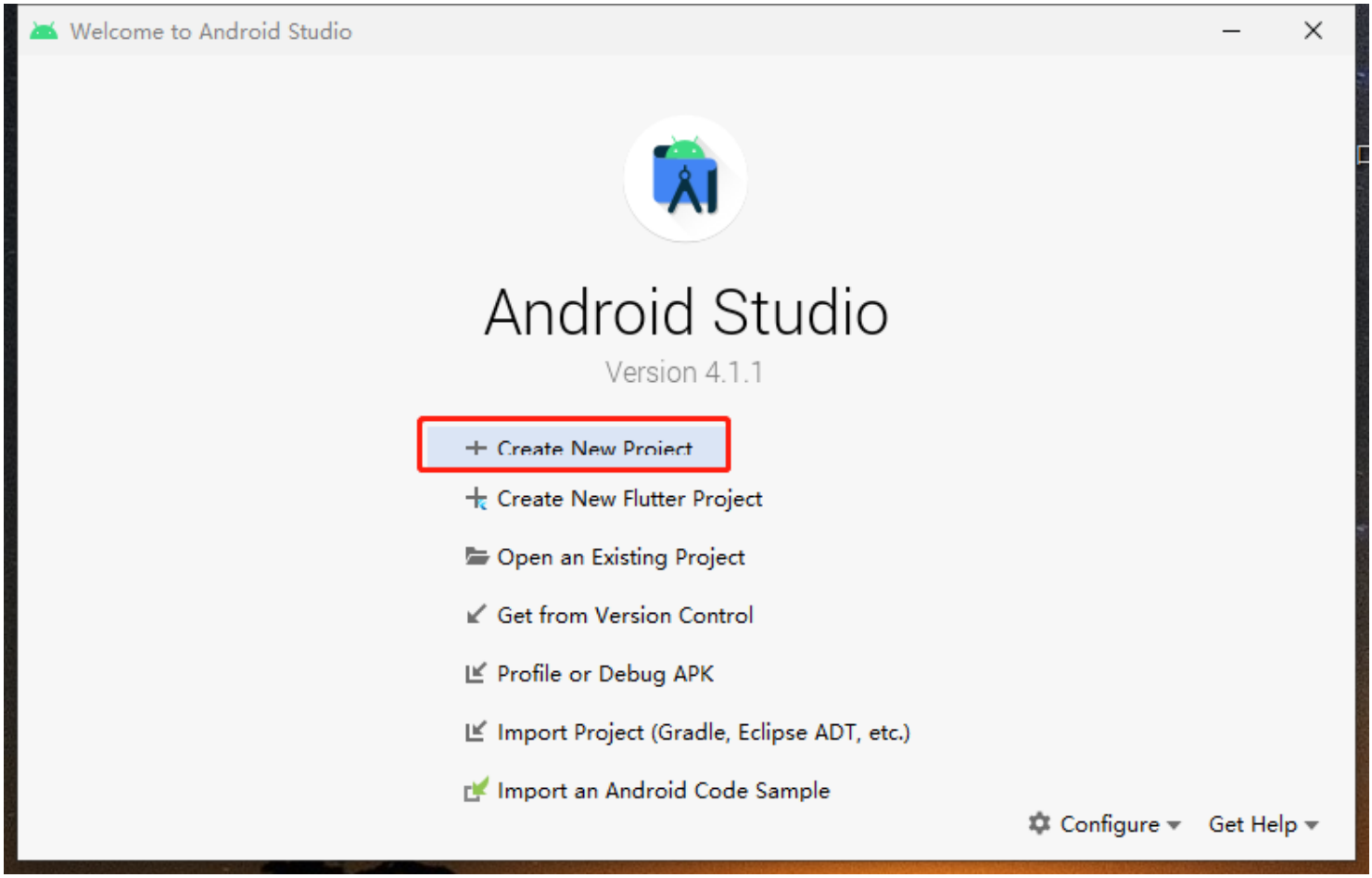
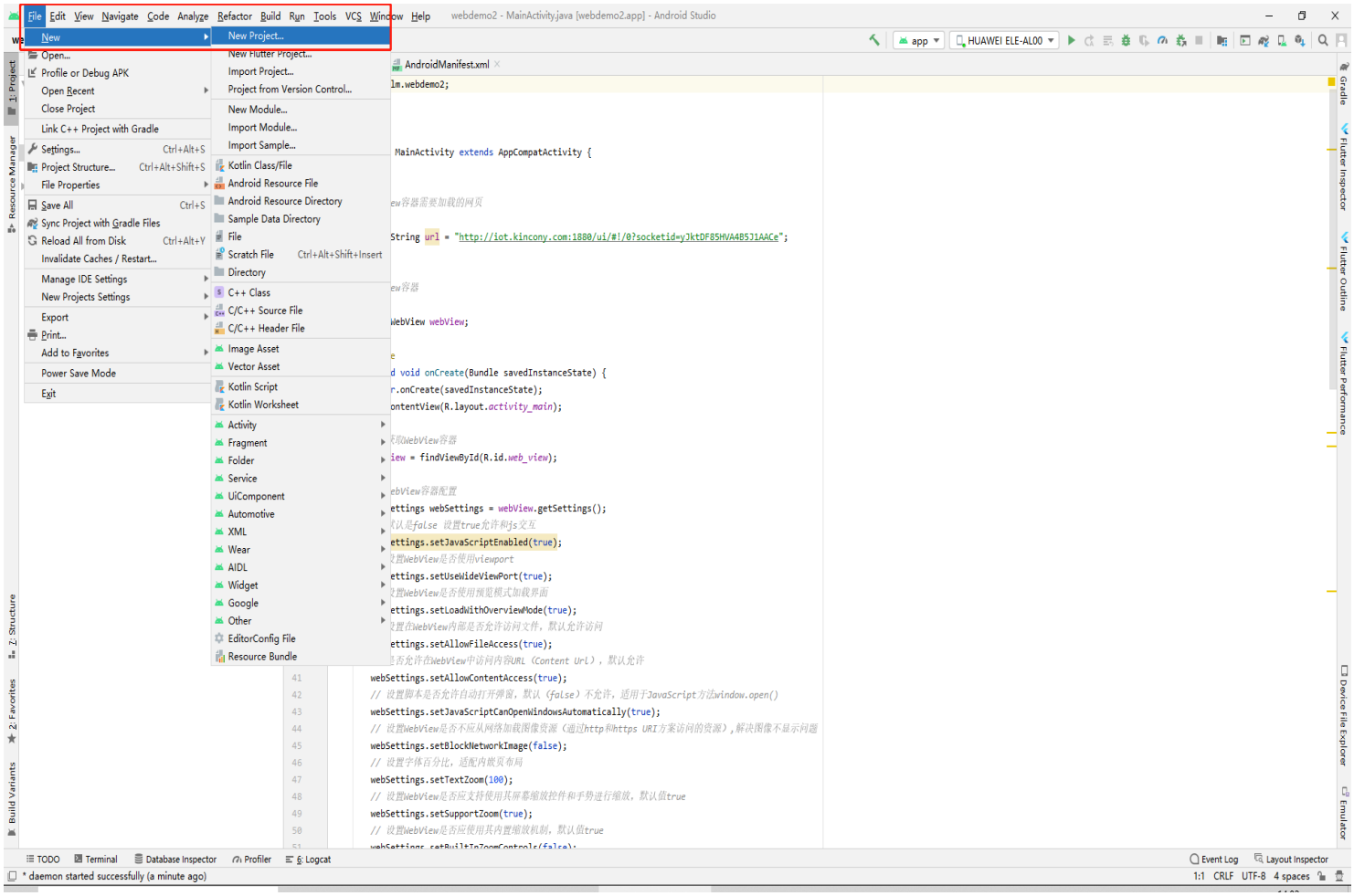
how to make android app from webpage - KinCony

Use Android Studio create new project

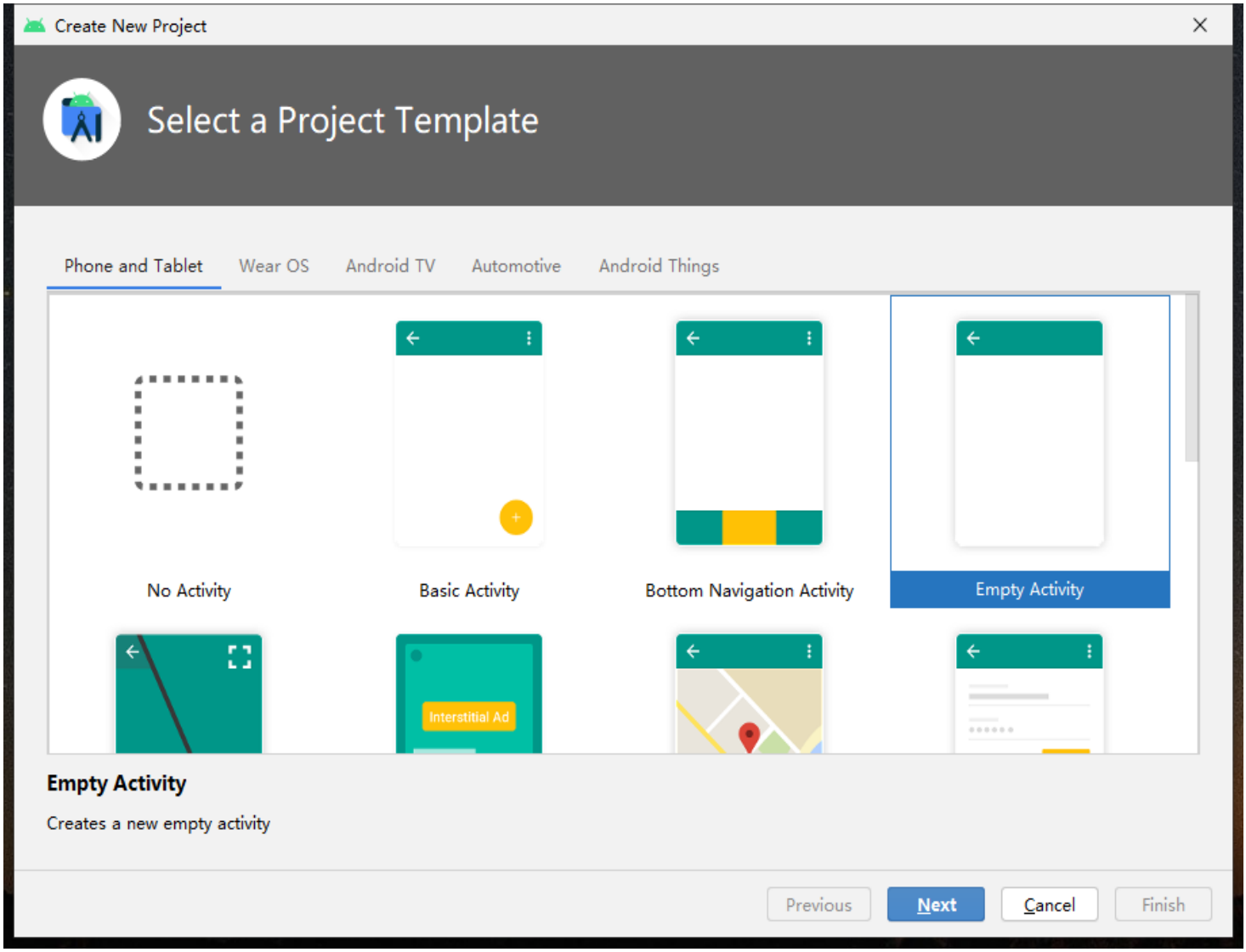
Open Android Studio, if you first time use Android Studio, click "`create New Project`".



if not first time use Android Studio, Android Studio will auto open last project， now you can click `File->New->New Project....`



Select “`Empty Activity`”, click “Next”.

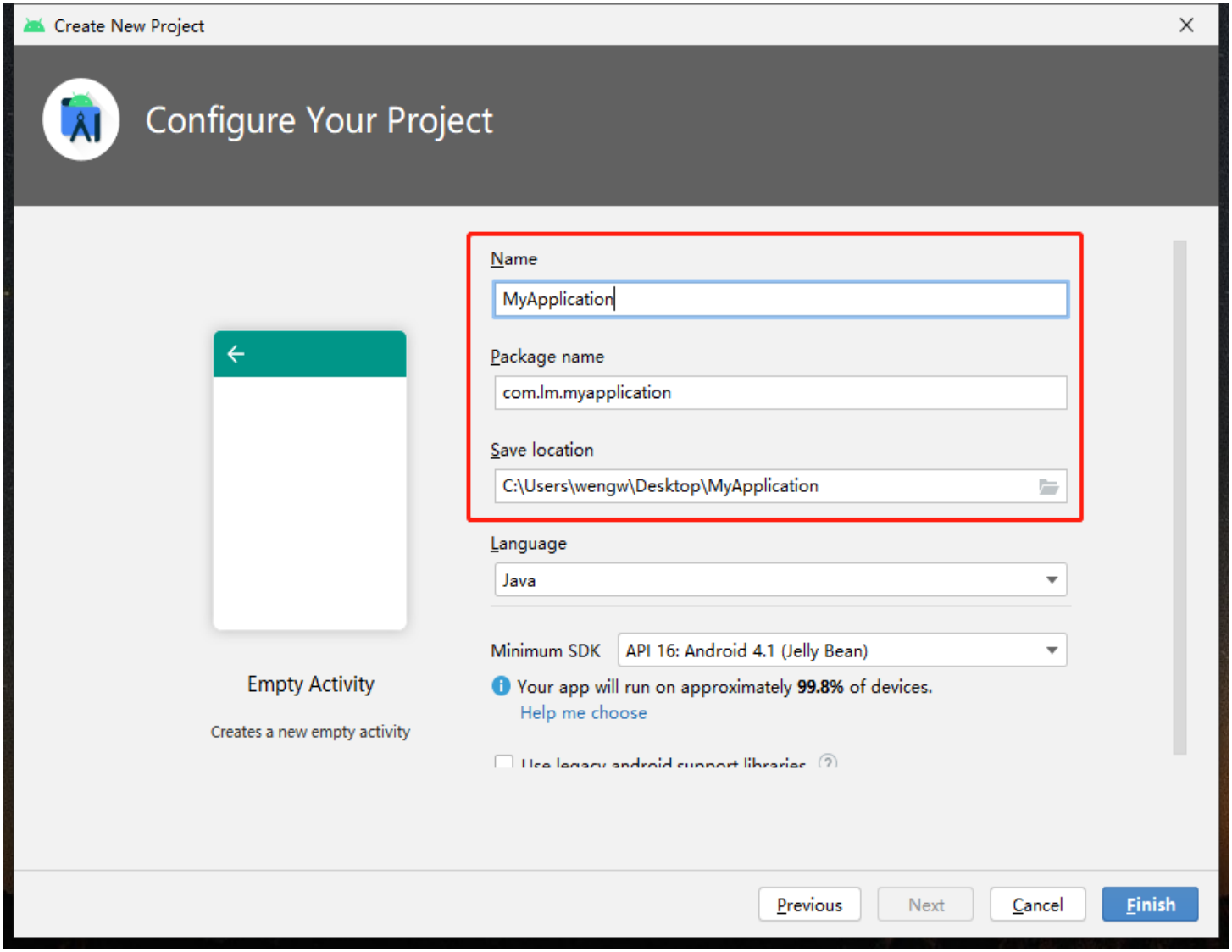


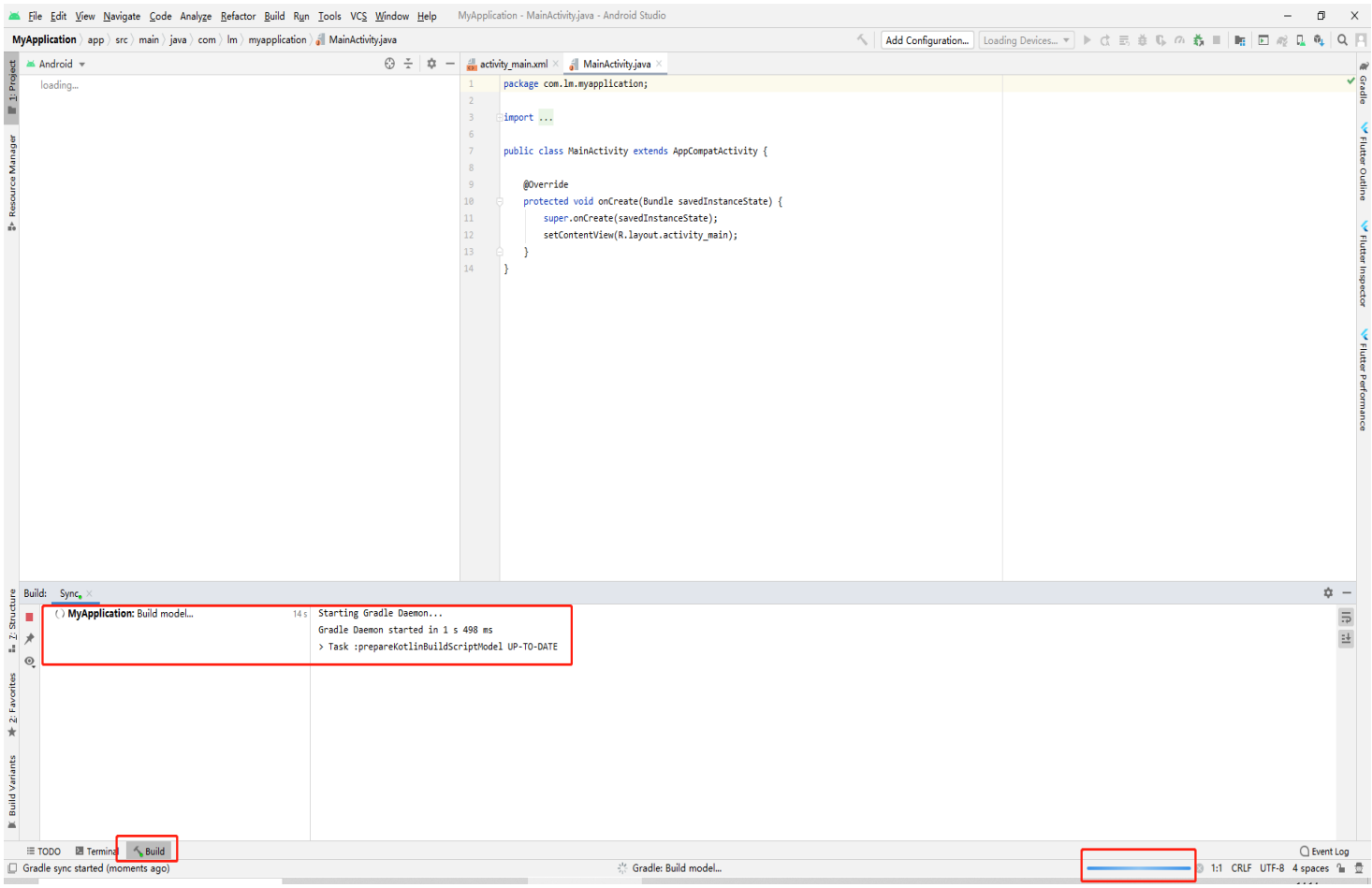
Input text for "`Name, Package name, Save location`" , others as default, not need to change, click "Finish".

`Name`: Project name.

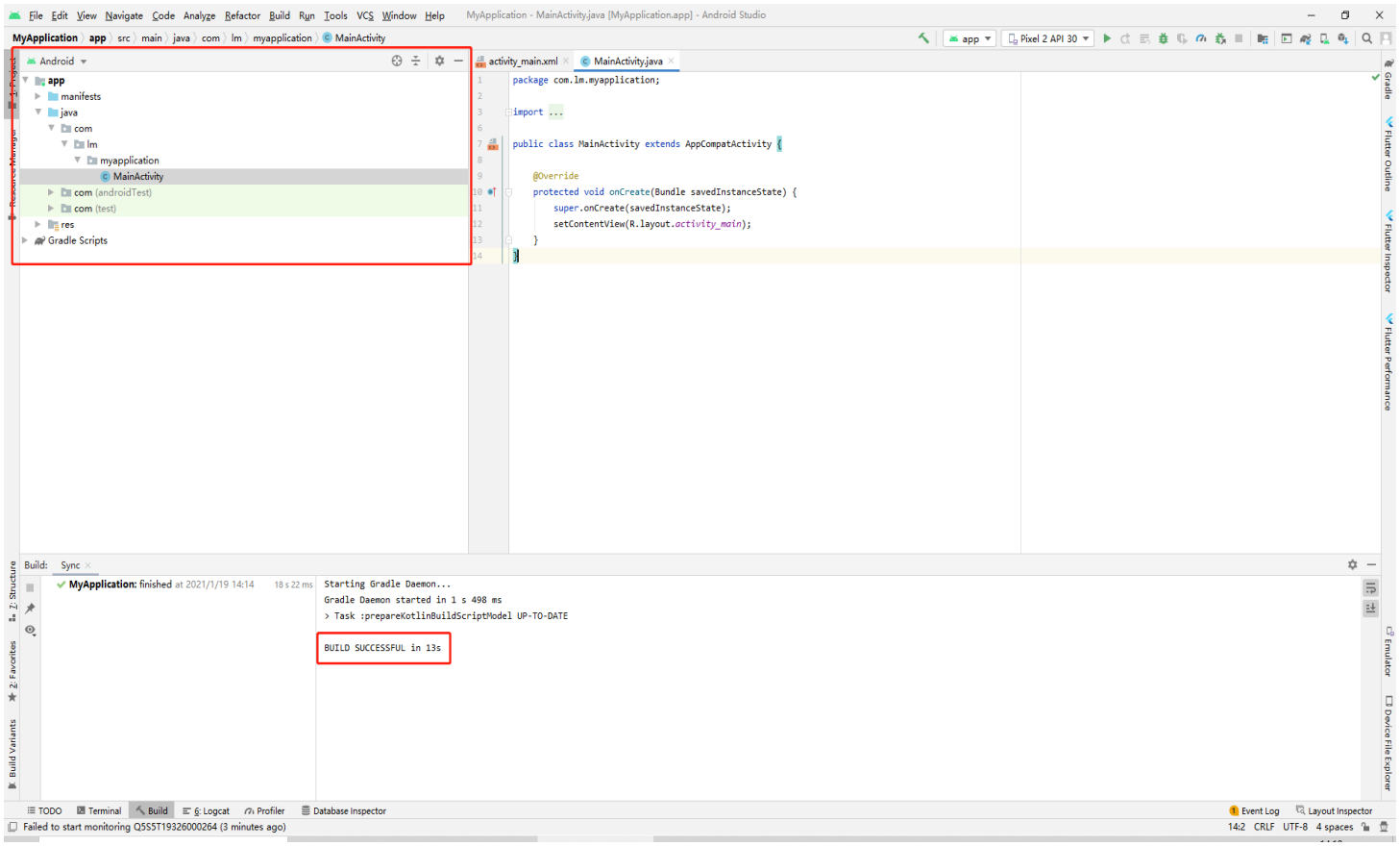
`Package name`: `The unique ID of the project cannot be the same as other applications, otherwise the installation will fail. It's better to start with your own domain name, such as com.kincony`

Save location: project file save path.


Create New Project
Configure Your Project
Name
MyApplication
Package name
com.lm.myapplication
Save location
C:\Users\wengw\Desktop\MyApplication
Language
Java
Minimum SDK
API 16: Android 4.1 (Jelly Bean)
Your app will run on approximately 99.8% of devices.
Help me choose
Empty Activity
Creates a new empty activity
Previous
Next
Cancel
Finish



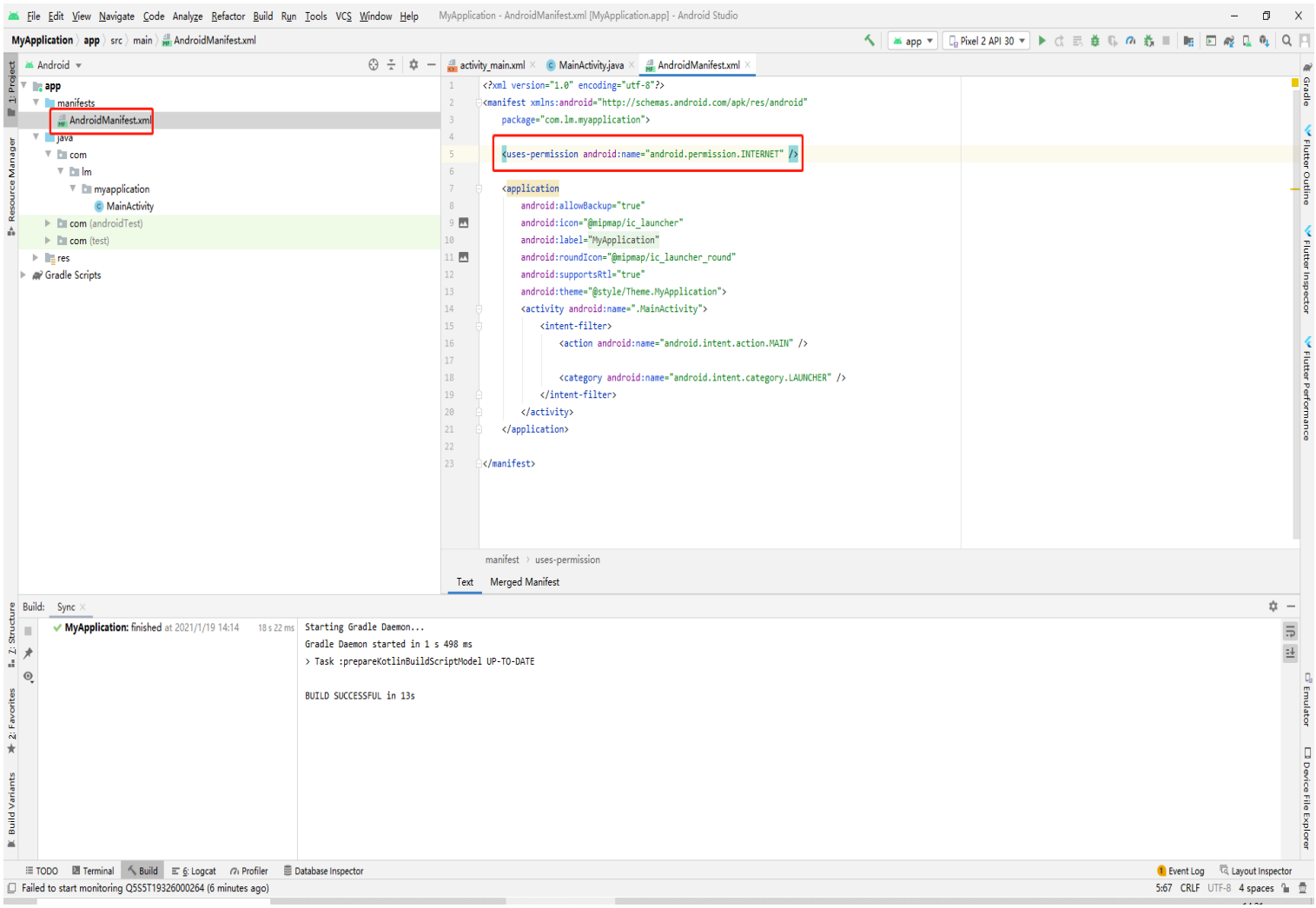
When Andorid Studio startup, you can write code now.



Add network access

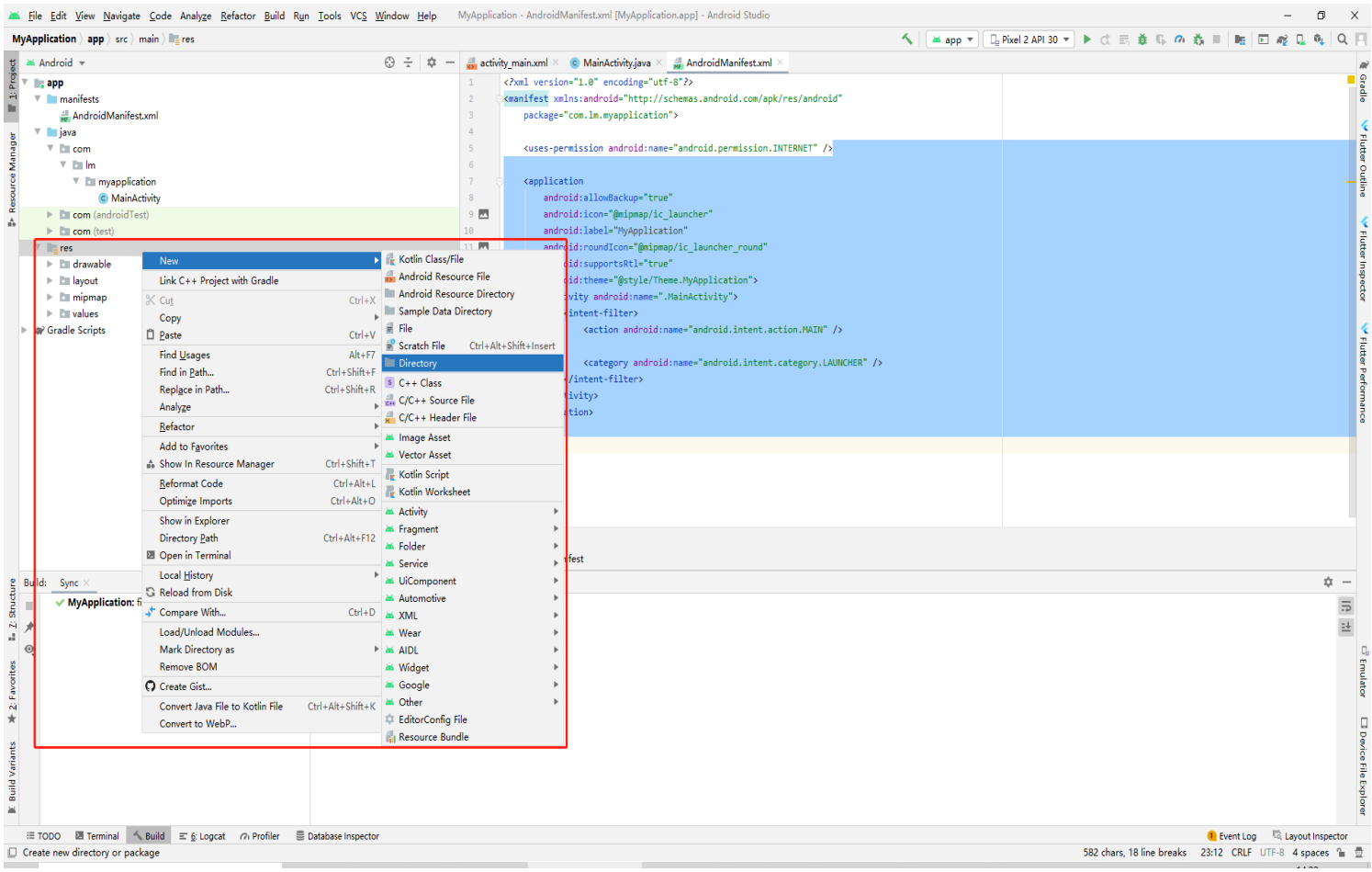
Open AndroidManifest.xml file, before <application insert code:

<uses-permission android:name="android.permission.INTERNET"/>

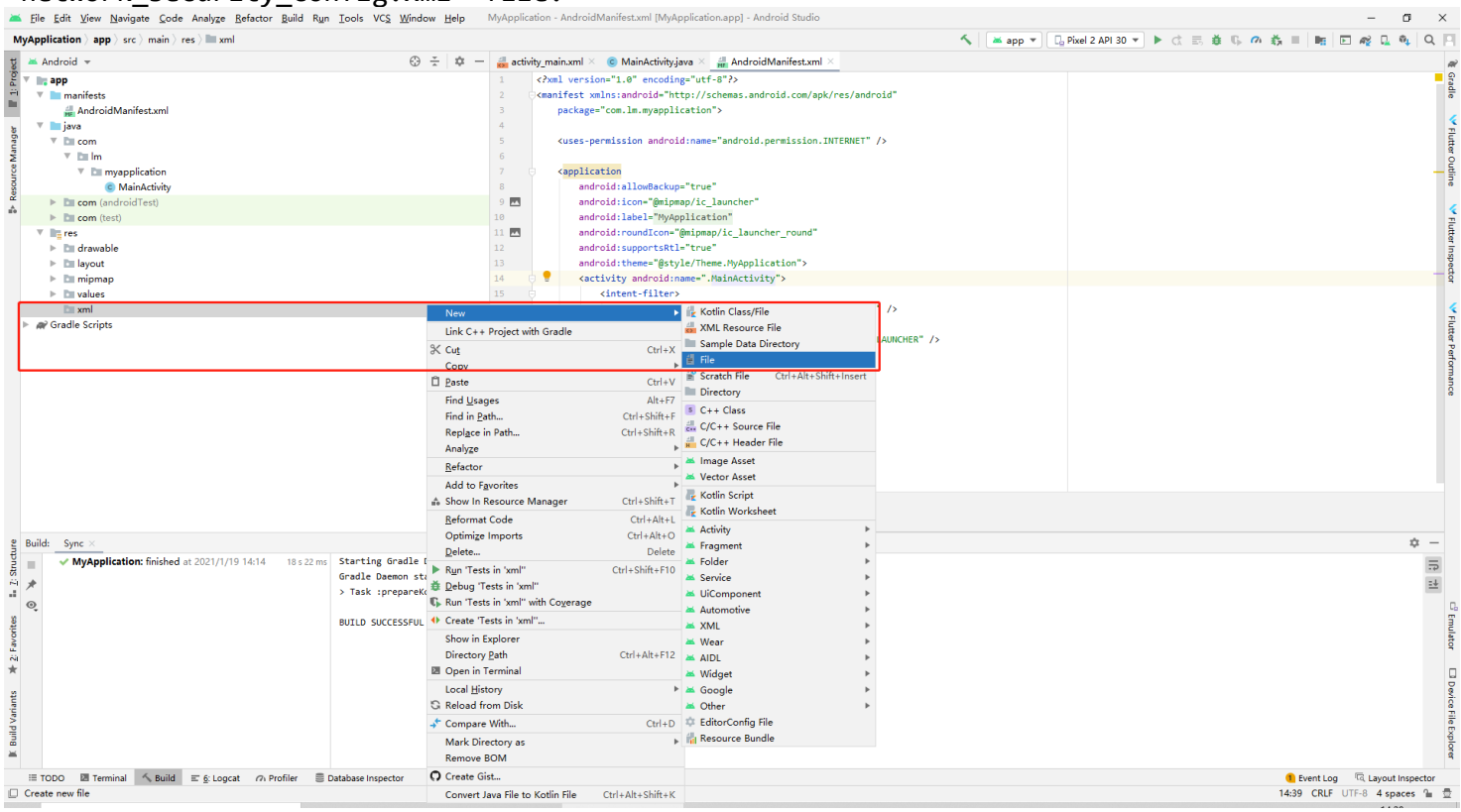


Support http protocol

Click “res” folder, mouse right button click “New-->Directory” create “xml”folder

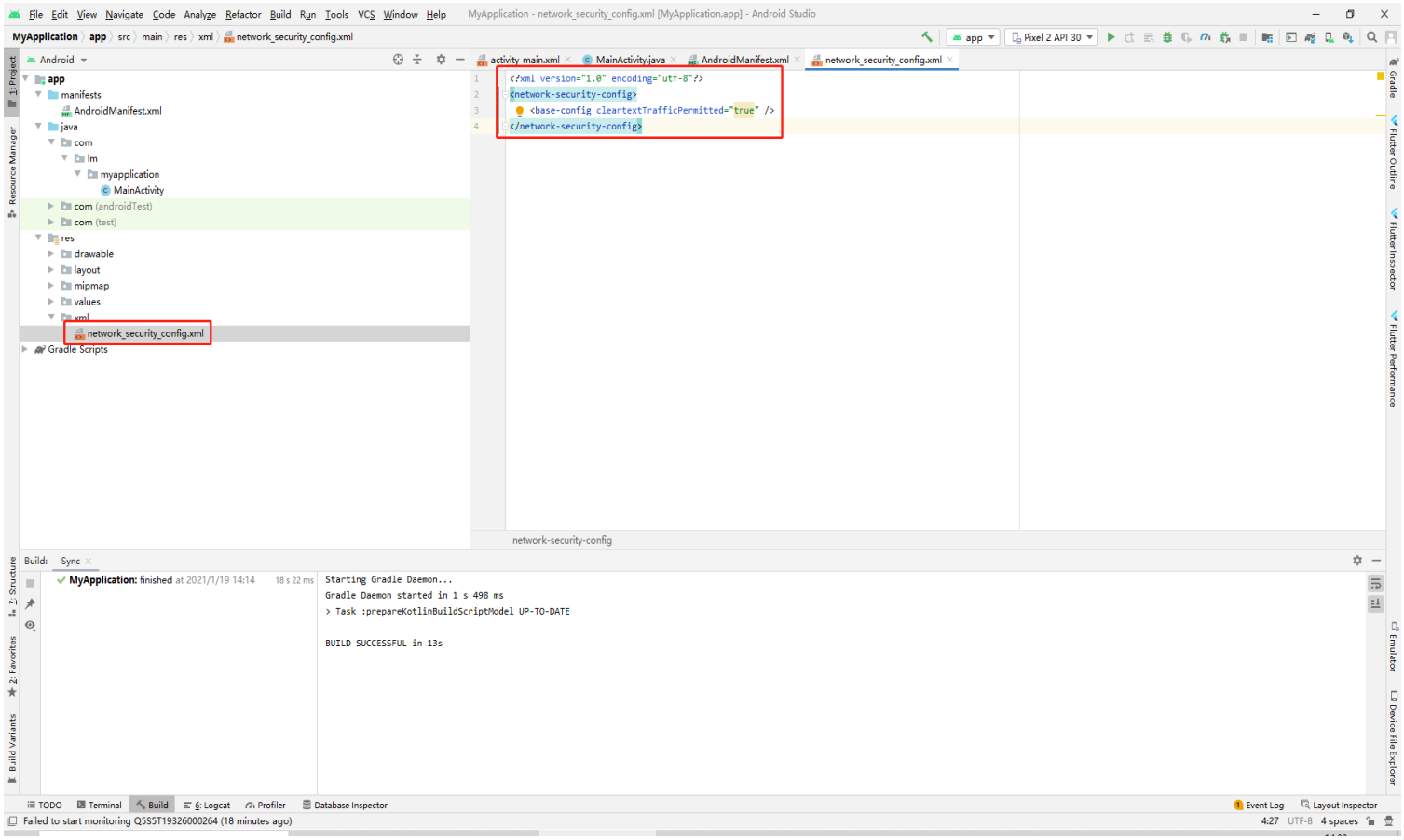


Chose “xml” folder, mouse right button click “New-->File” create “network_security_config.xml” file.



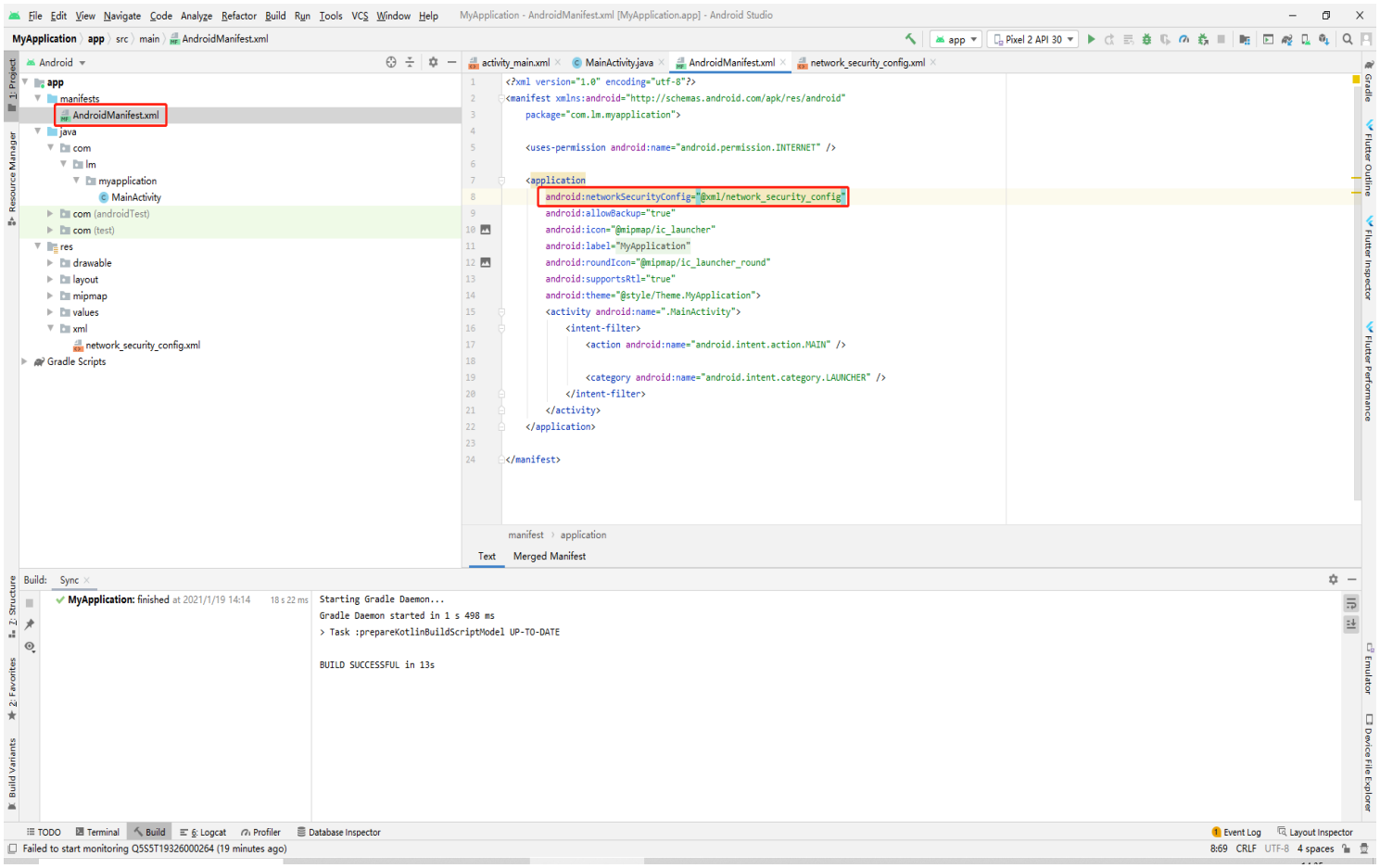
Modify “network_security_config.xml” input these code:

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted="true" />
</network-security-config>
```



Open “AndroidManifest.xml” again, between <application add code:

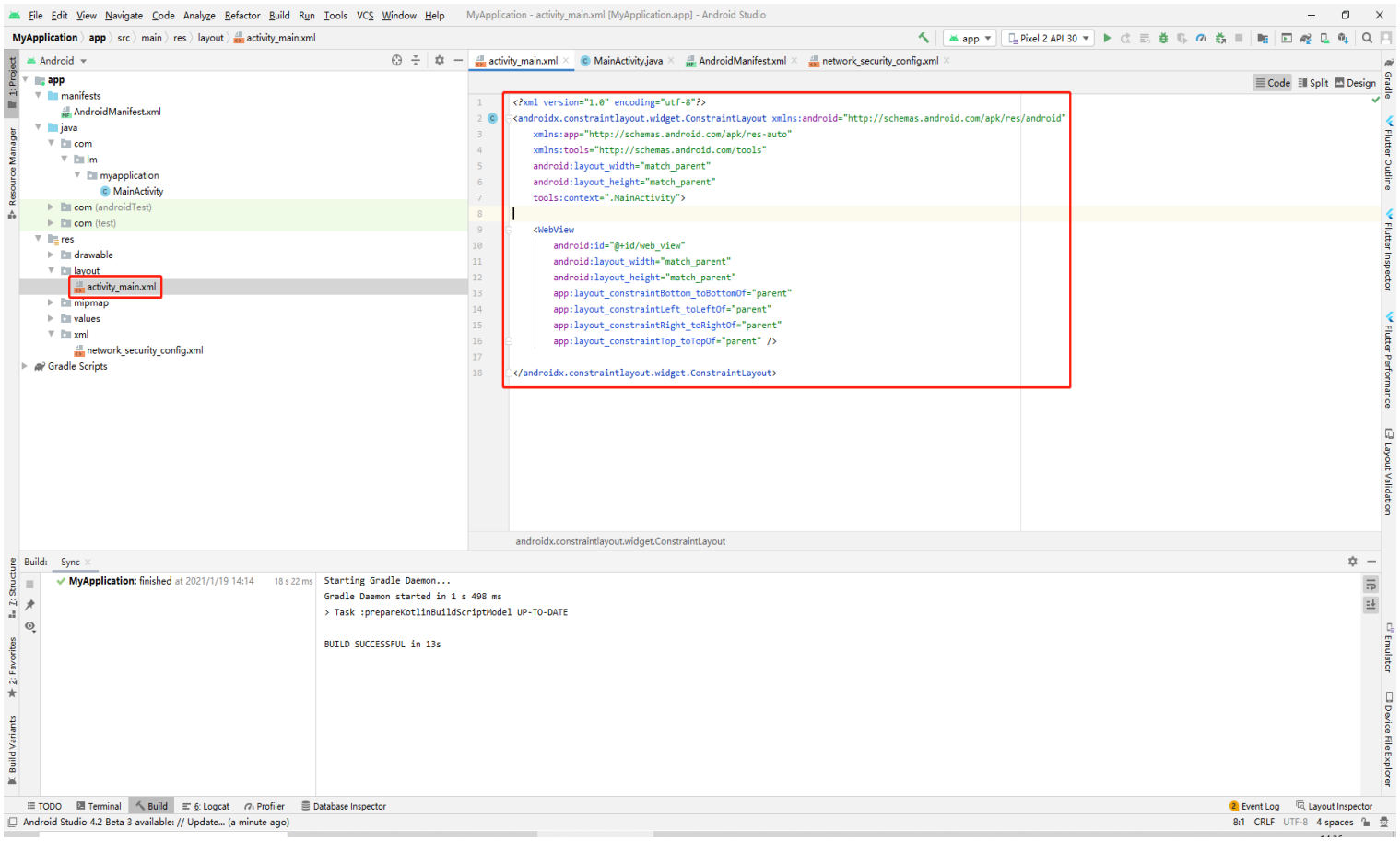
android:networkSecurityConfig="@xml/network_security_config"



Add WebView

Open “activity_main.xml”, modify code as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/web_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

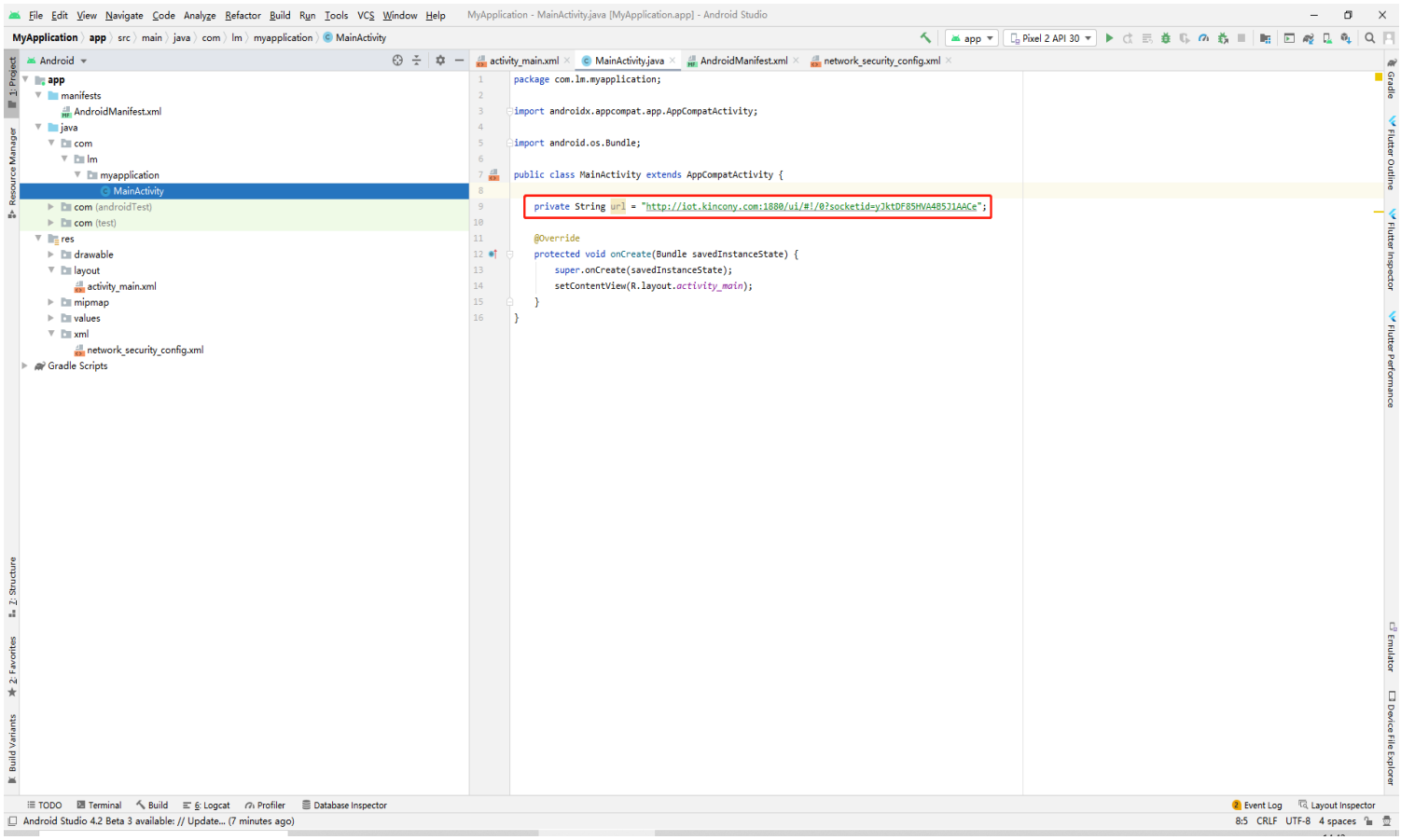


Use WebView load webpage

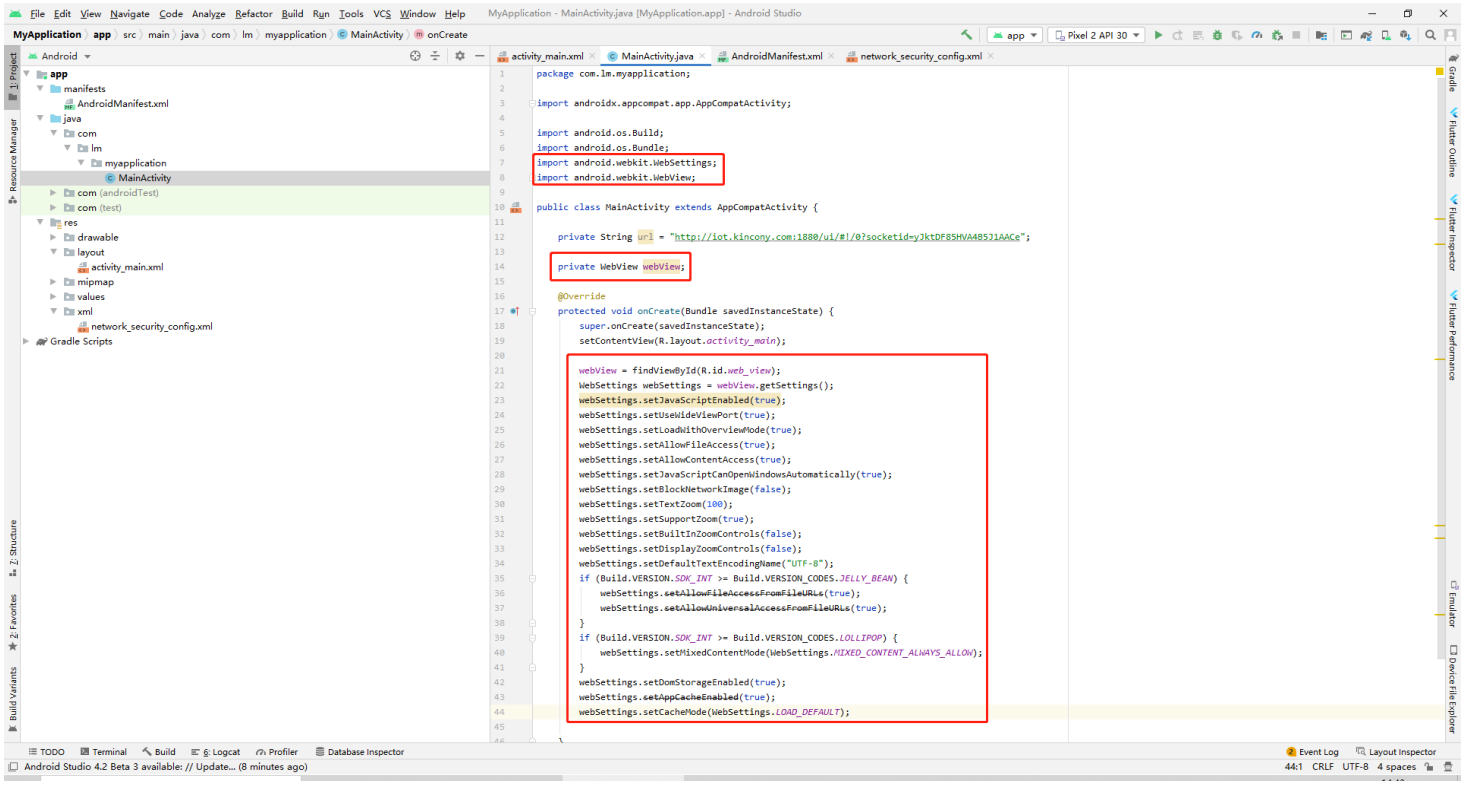
Open “MainActivity”, define website url you want to open, such as:

private String url = " http://iot.kincony.com:1880/ui/#!/0?socketid=yJktDF85HVA4B5J1AACe";

This is your url.



Config WebView as red region:



Code list:

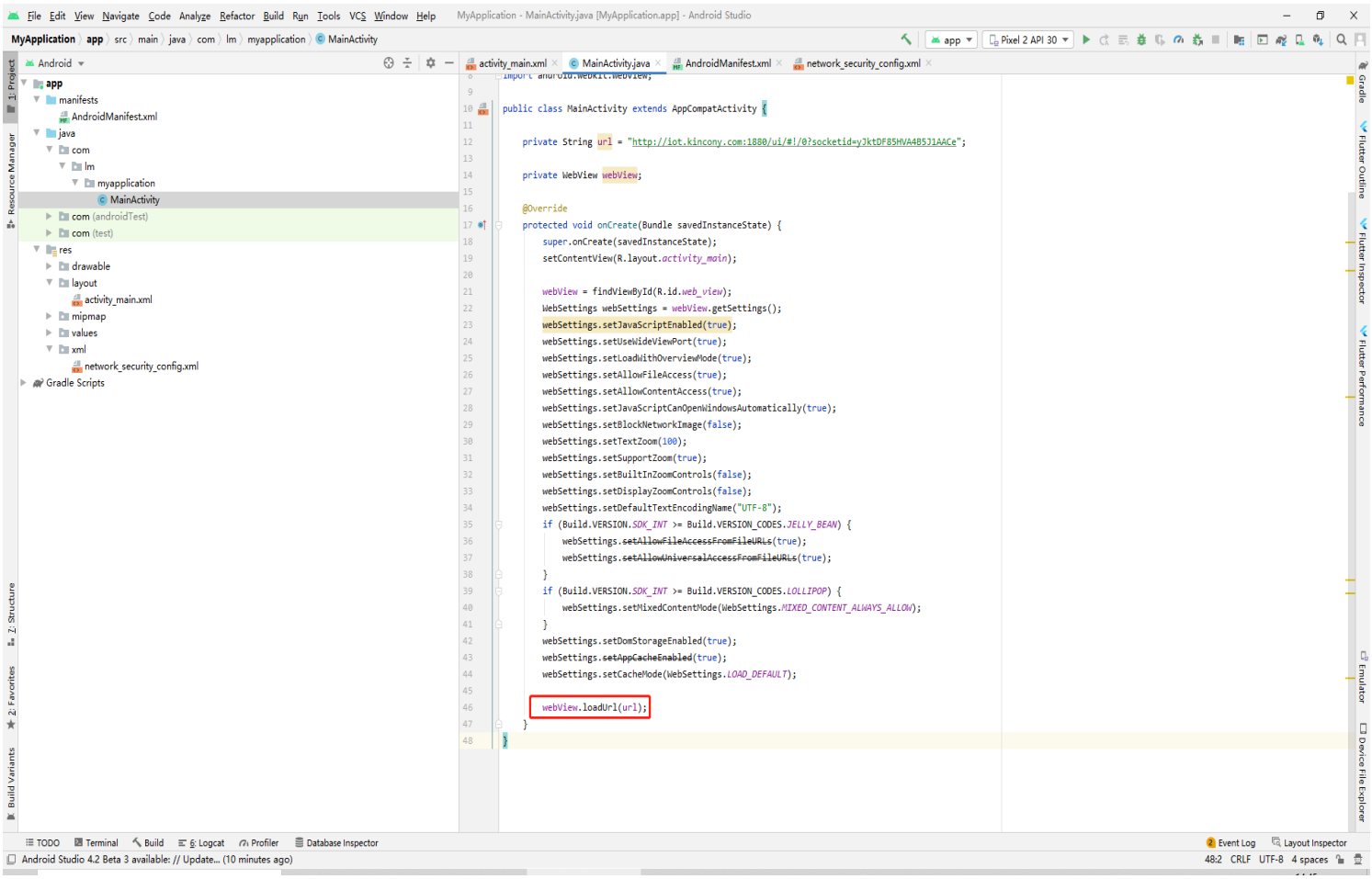
```
import android.os.Build;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;



  private WebView webView;



    webView = findViewById(R.id.web_view);
    WebSettings webSettings = webView.getSettings();
    webSettings.setJavaScriptEnabled(true);
    webSettings.setUseWideViewPort(true);
    webSettings.setLoadWithOverviewMode(true);
    webSettings.setAllowFileAccess(true);
    webSettings.setAllowContentAccess(true);
    webSettings.setJavaScriptCanOpenWindowsAutomatically(true);
    webSettings.setBlockNetworkImage(false);
    webSettings.setTextZoom(100);
    webSettings.setSupportZoom(true);
    webSettings.setBuiltInZoomControls(false);
    webSettings.setDisplayZoomControls(false);
    webSettings.setDefaultTextEncodingName("UTF-8");
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {
      webSettings.setAllowFileAccessFromFileURLs(true);
      webSettings.setAllowUniversalAccessFromFileURLs(true);
    }
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
      webSettings.setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW);
    }
    webSettings.setDomStorageEnabled(true);
```

webSettings.setAppCacheEnabled(true);

webSettings.setCacheMode(WebSettings.LOAD_DEFAULT);

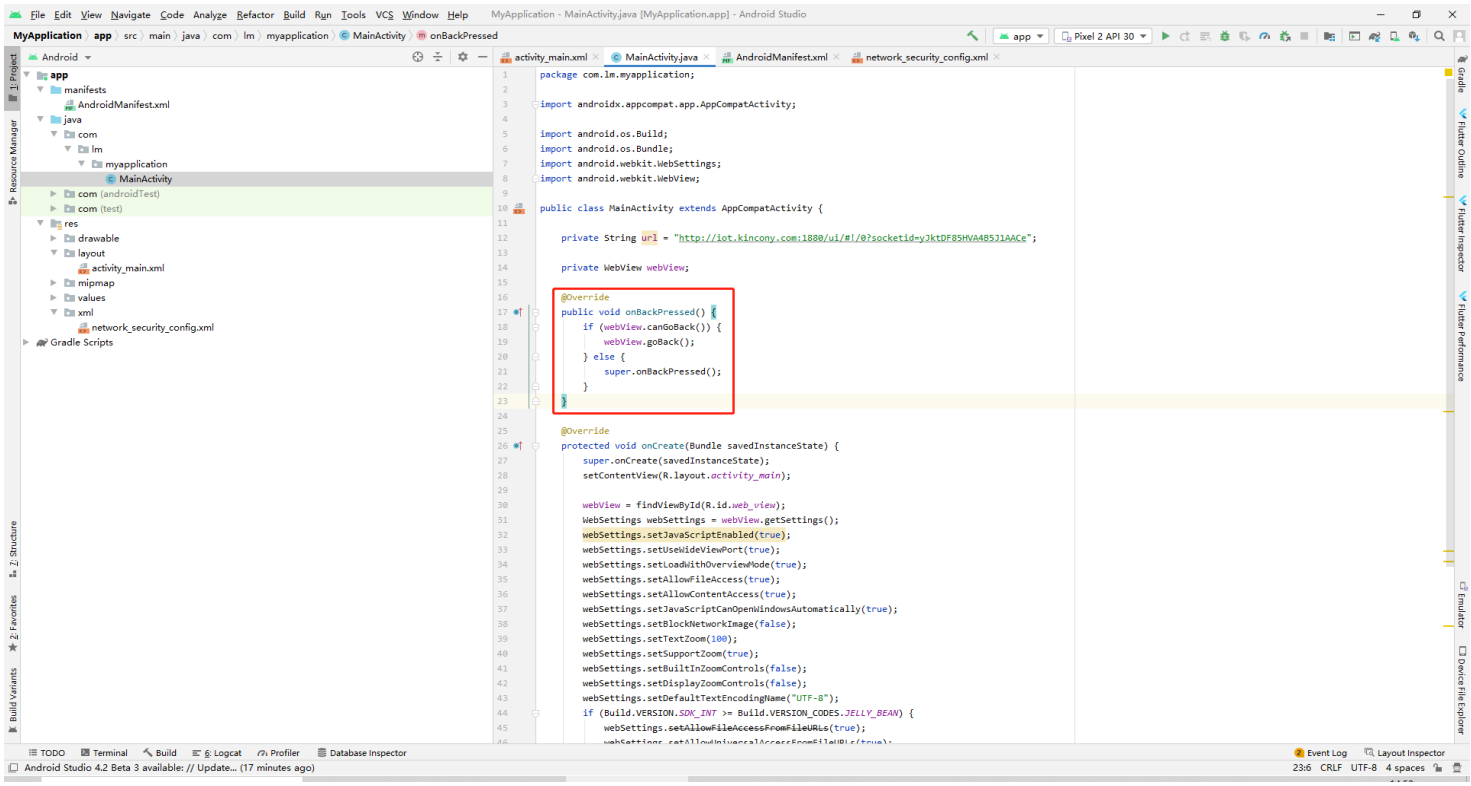
webView.loadUrl(url);

Config completed.



Return button and click event

Open “`MainActivity`” add these code:

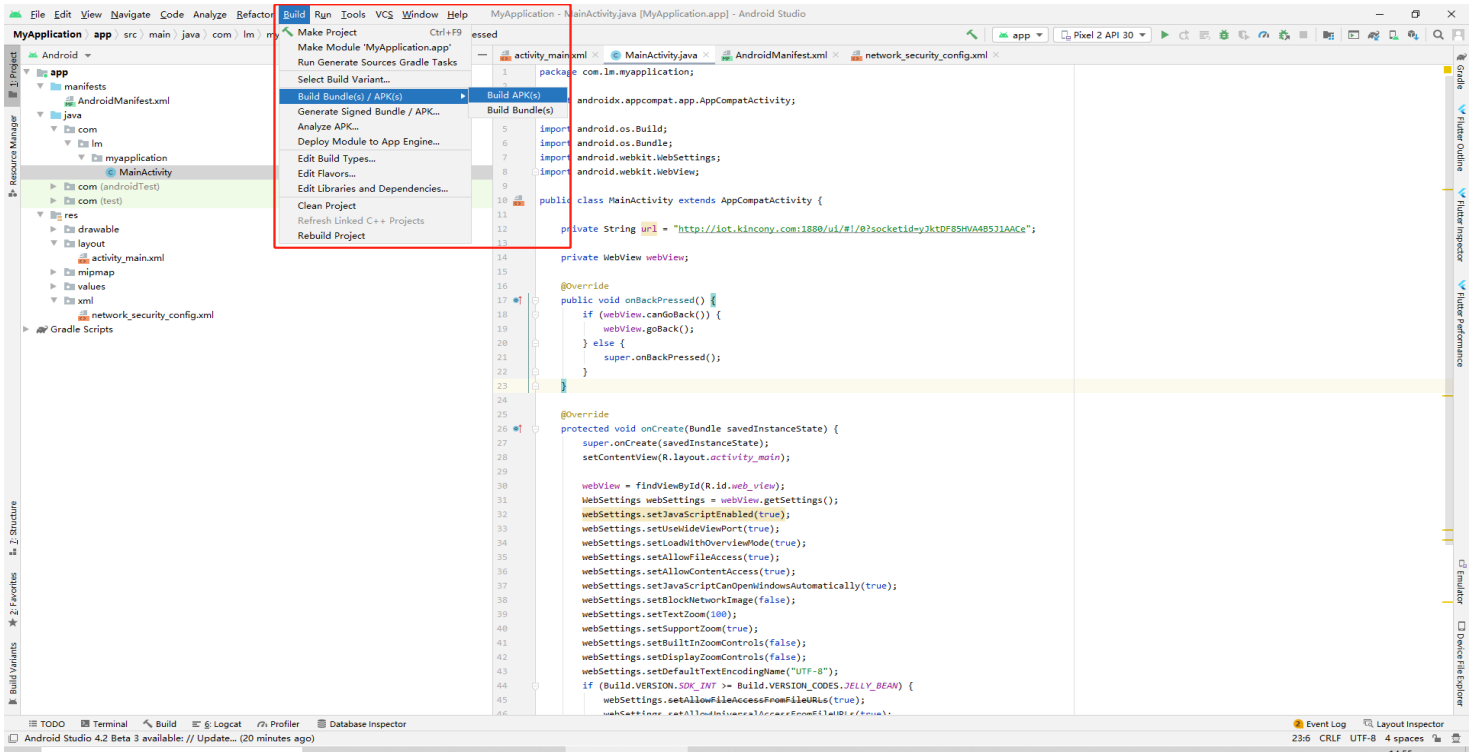


Code list:

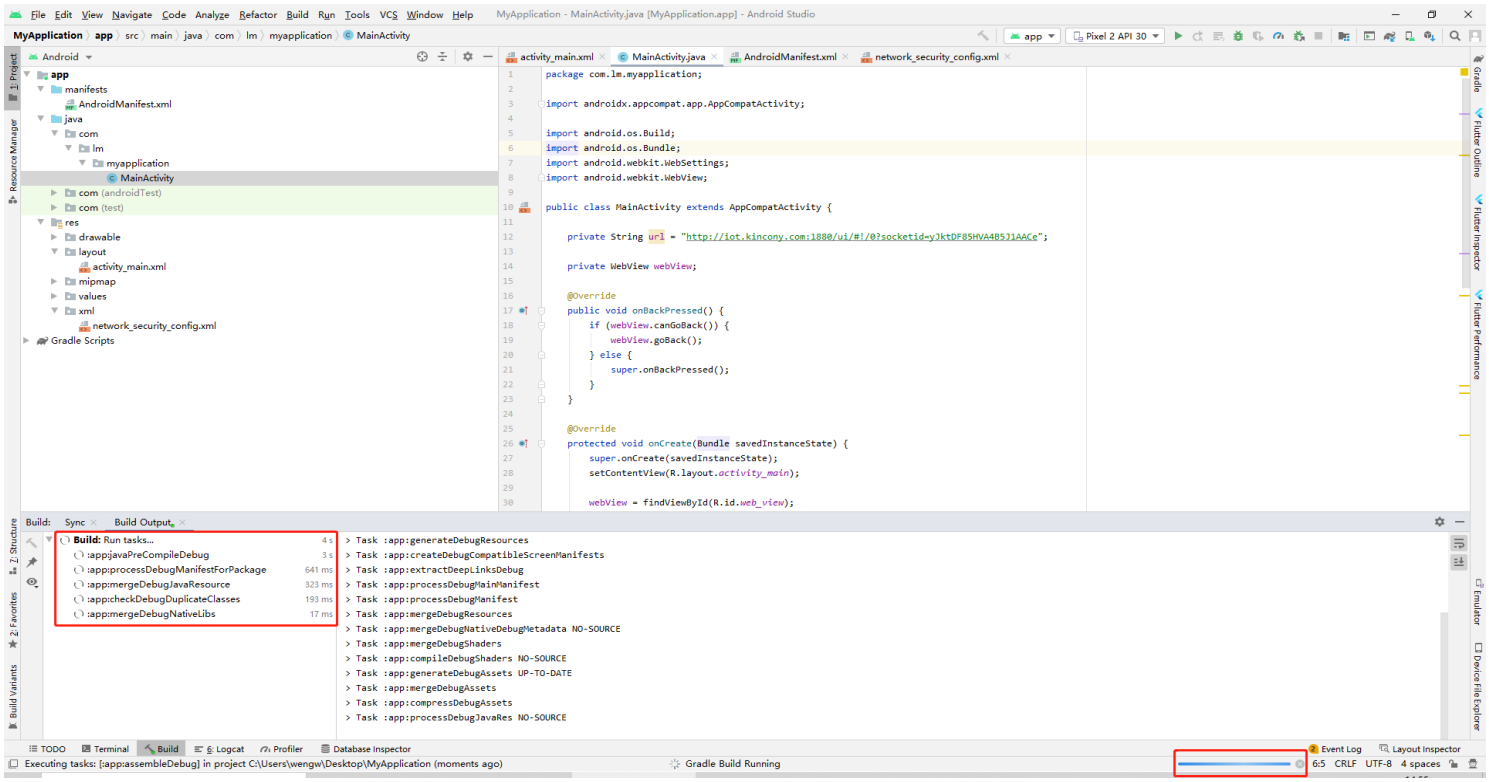
```
@Override
public void onBackPressed() {
    if (webView.canGoBack()) {
        webView.goBack();
    } else {
        super.onBackPressed();
    }
}
```

Debug

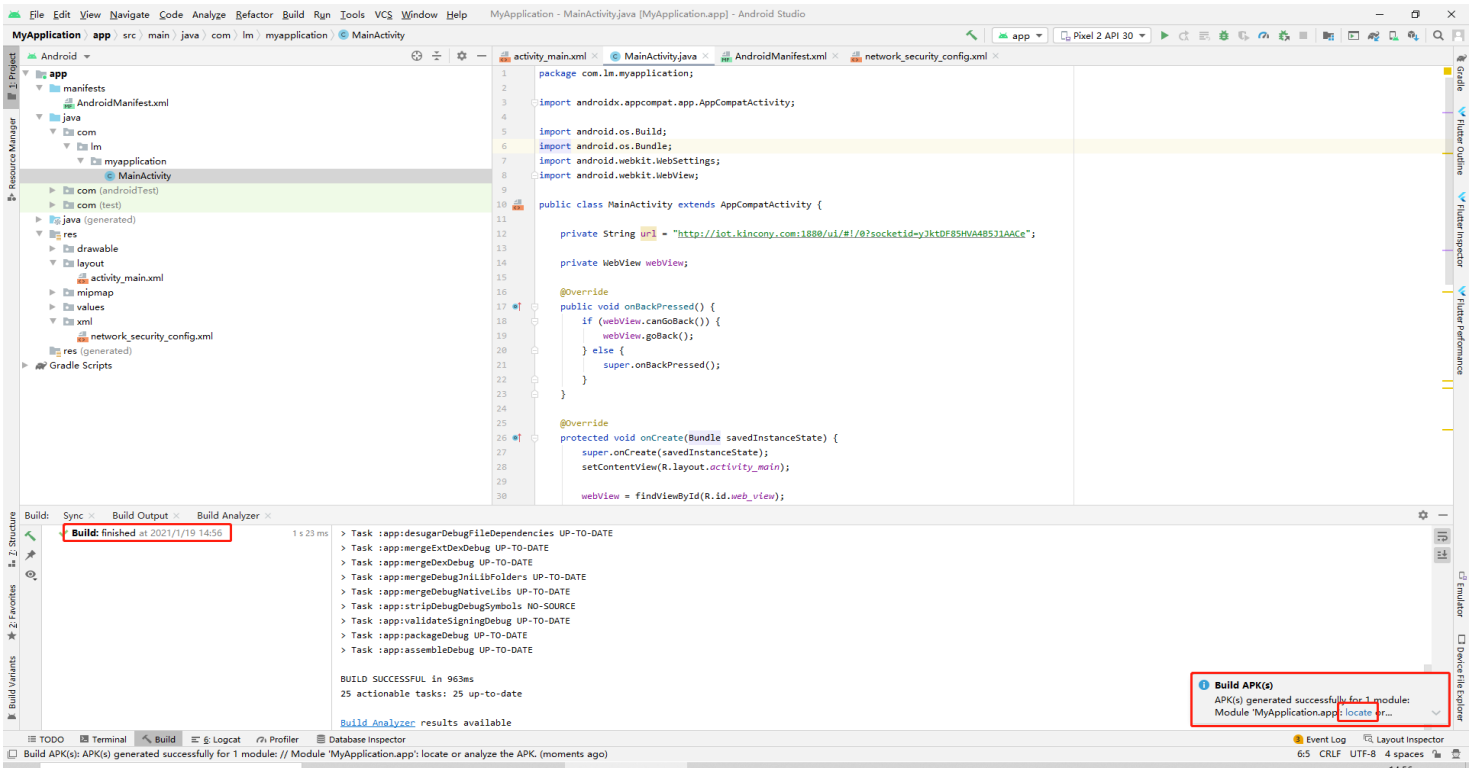
`Click Build-->Build Bunle(s)/APK(s)-->Build APK(s) Compile the project into an APK file`



Waitting Android Studio `Compile, it will takes about 1 minute.`

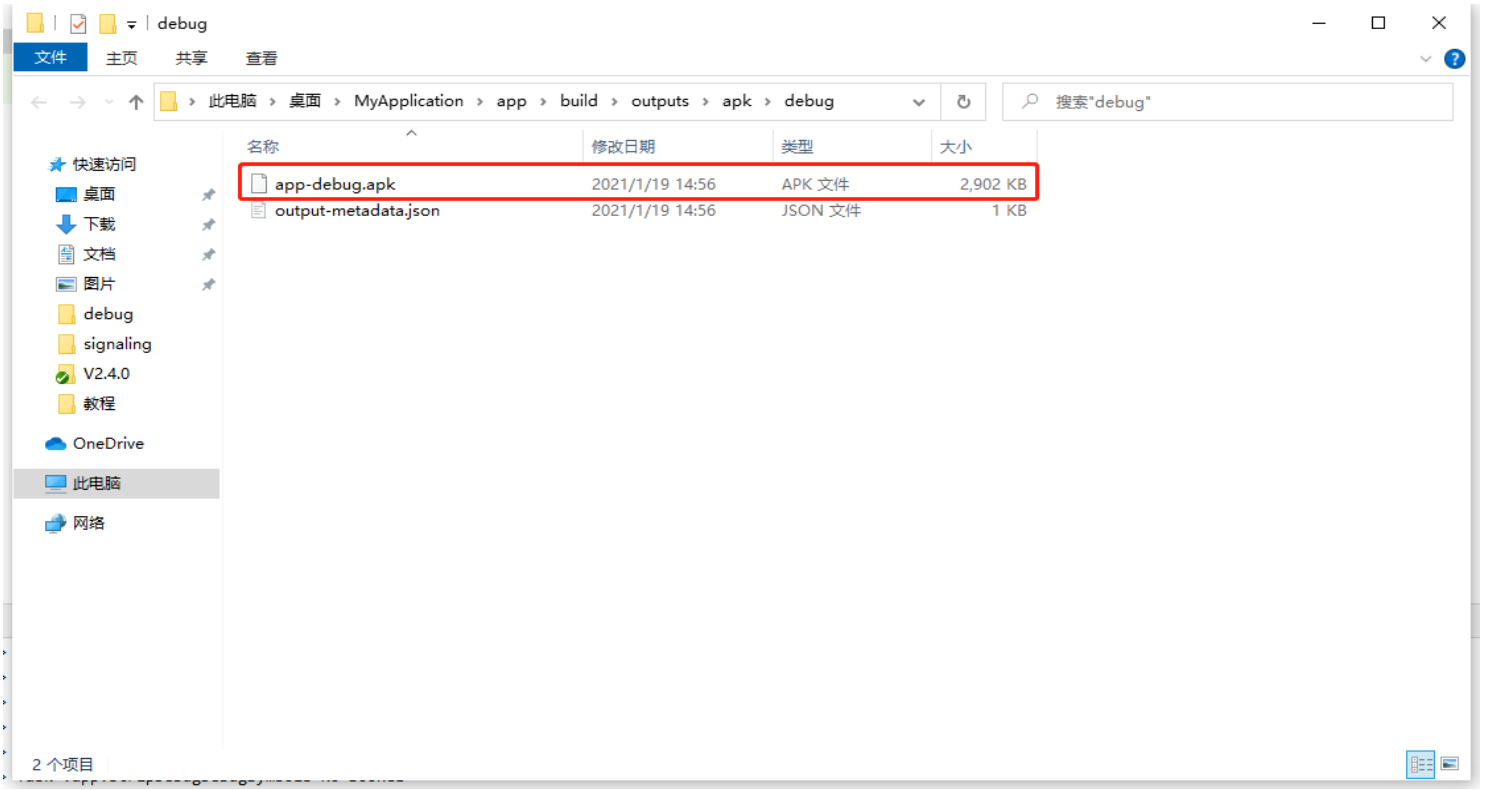


After compiling, a prompt will pop up. Click the “locate” button on the prompt to directly open the APK file directory



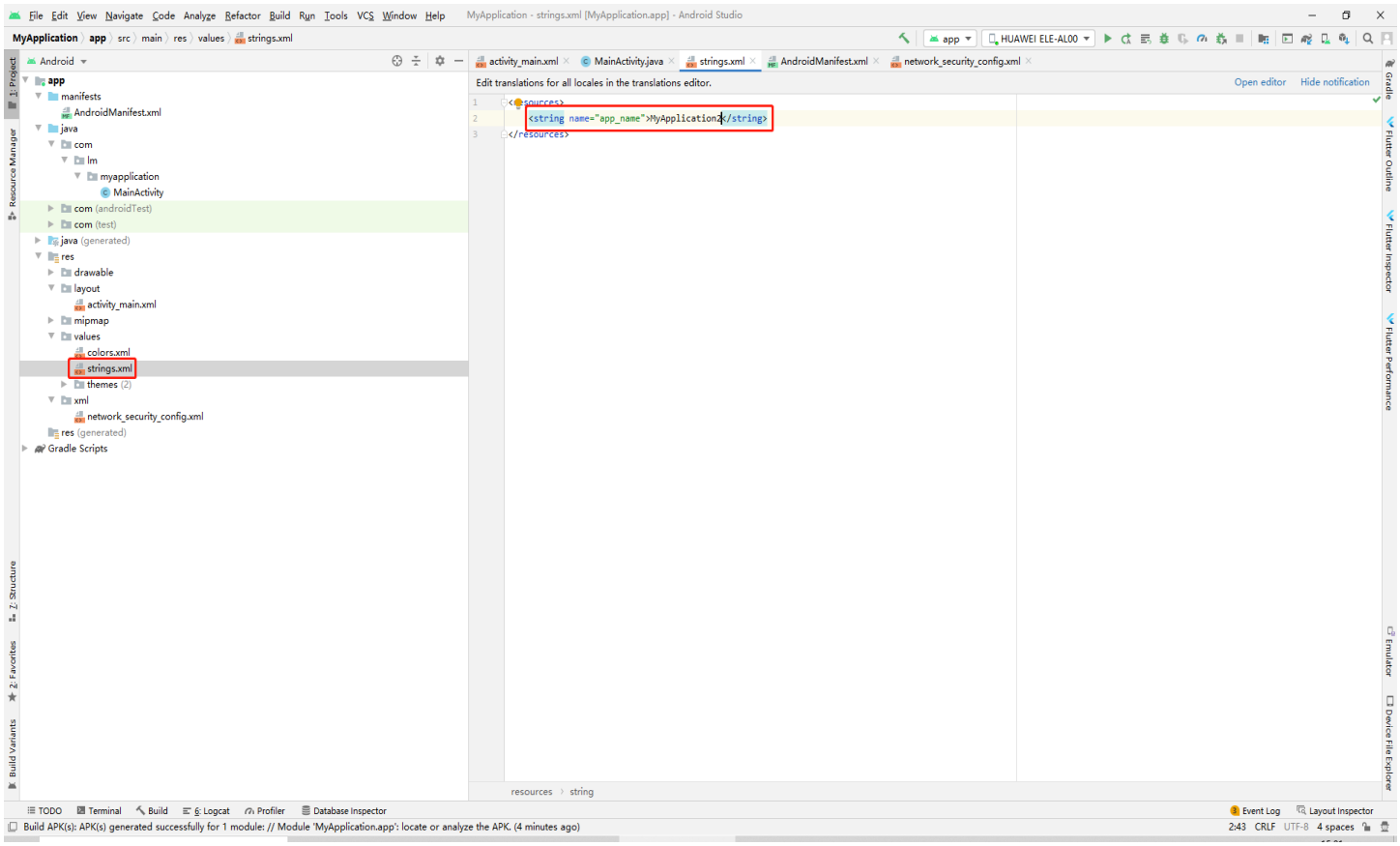
If there is no pop-up prompt, you can also search under the save location path defined before.

"Save location" path: app/build/outputs/apk/debug



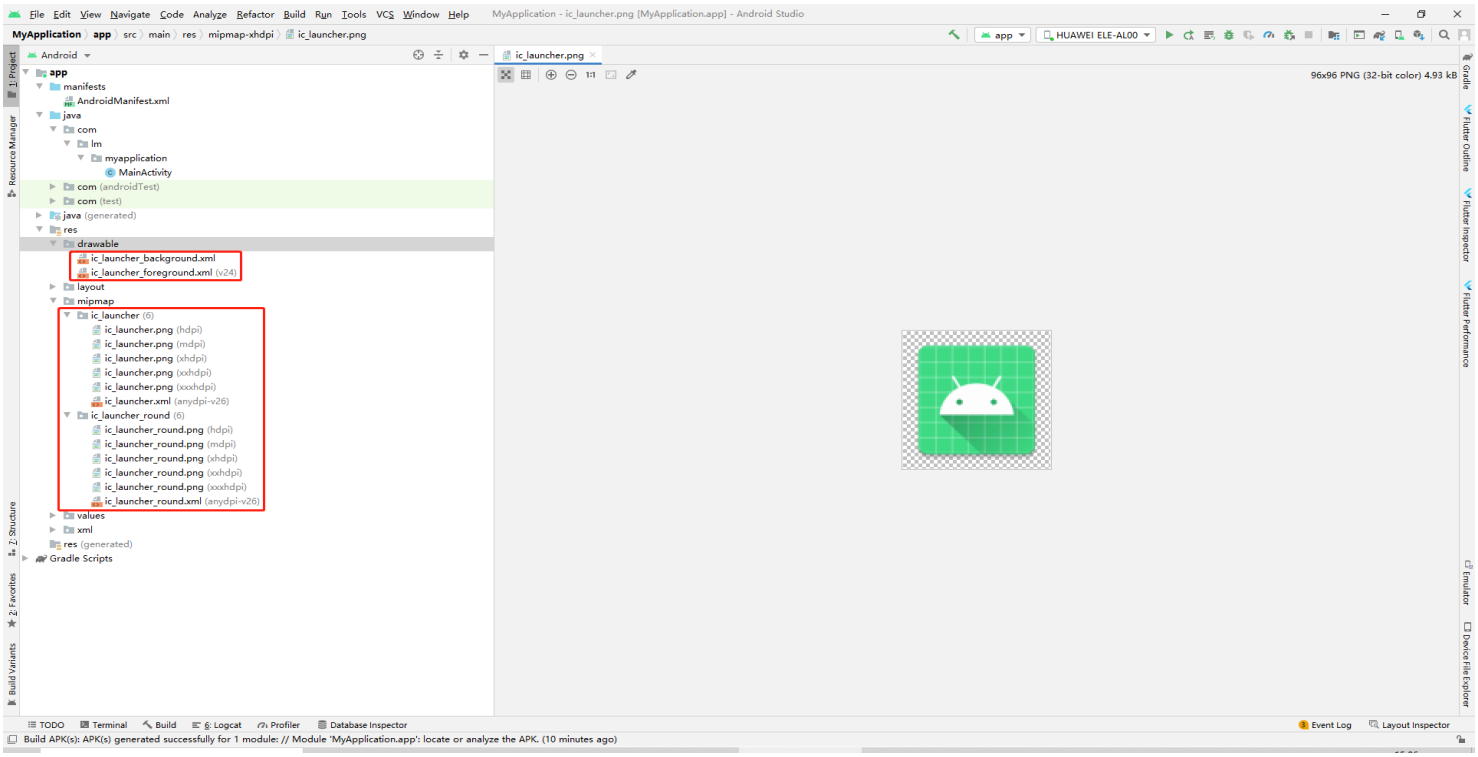
Modify application name

Open `res/values/string.xml,modify app_name`。Now our application name is “`MyApplication2`”

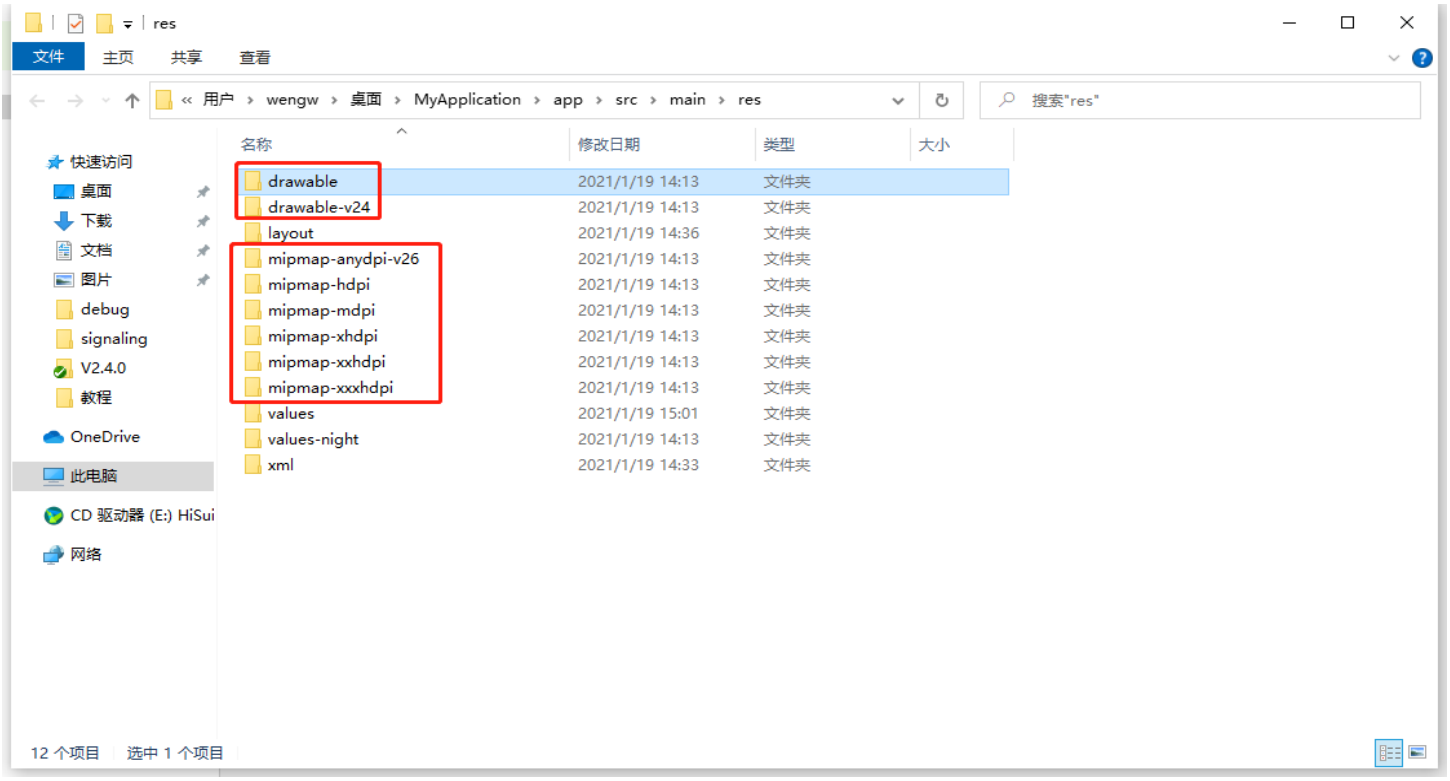


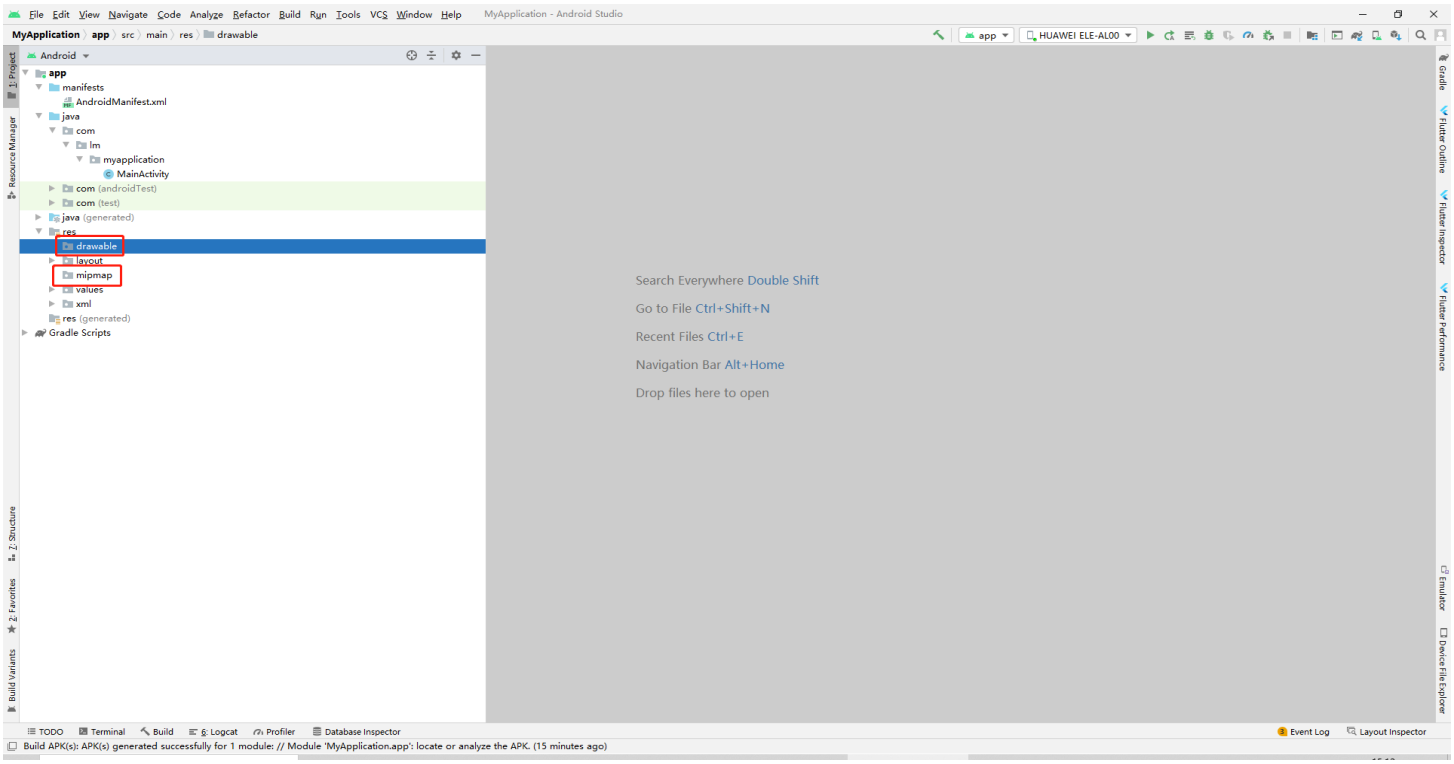
Modify application ico

“res/drawable” and “res/mipmap” folder, Here is the icon generated by Android studio for us by default. We can replace it to make the icon our own. IC_ launcher.png Represents a square icon, IC_ launcher_ round.png Represents a circular icon, which use for different mobile phones.

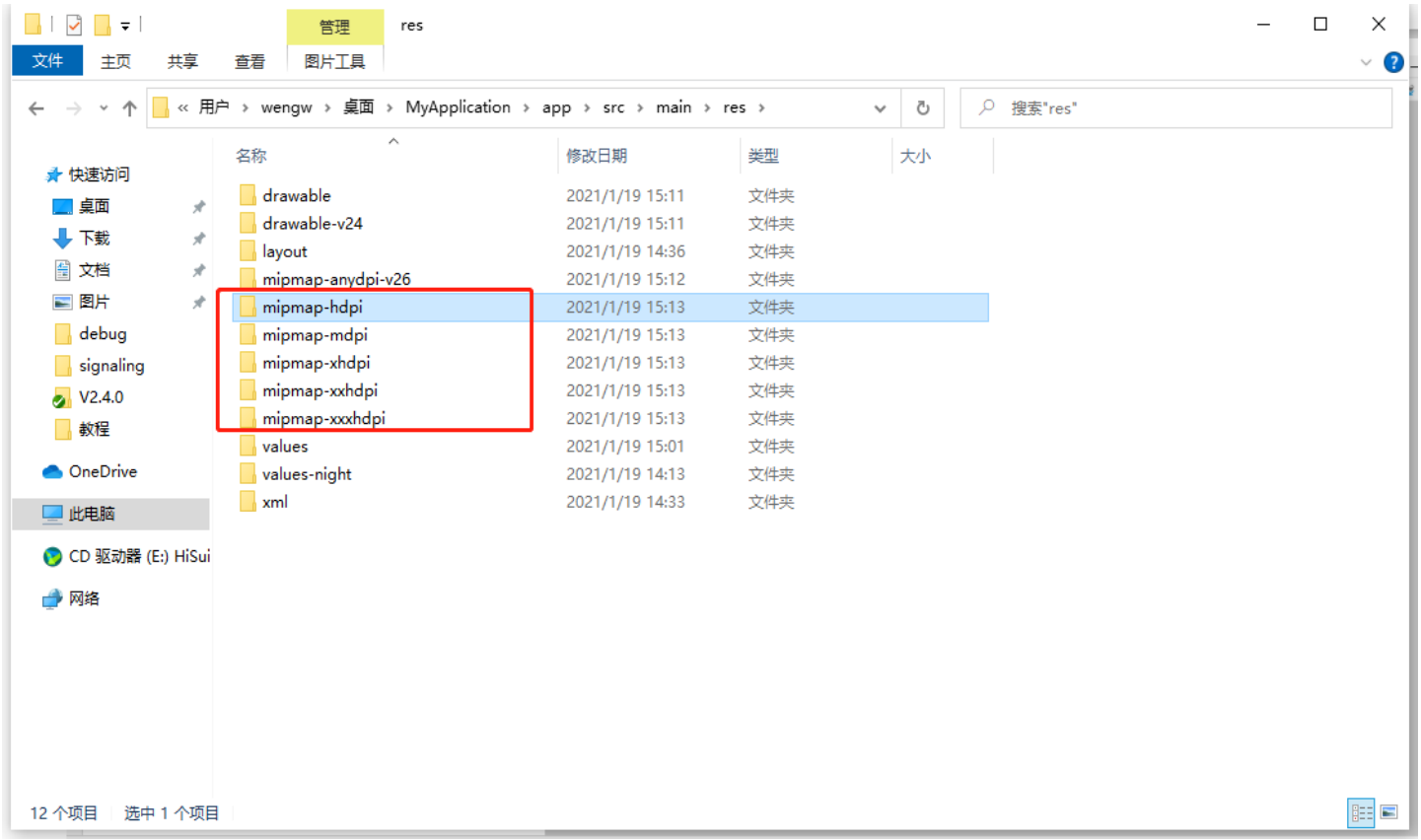


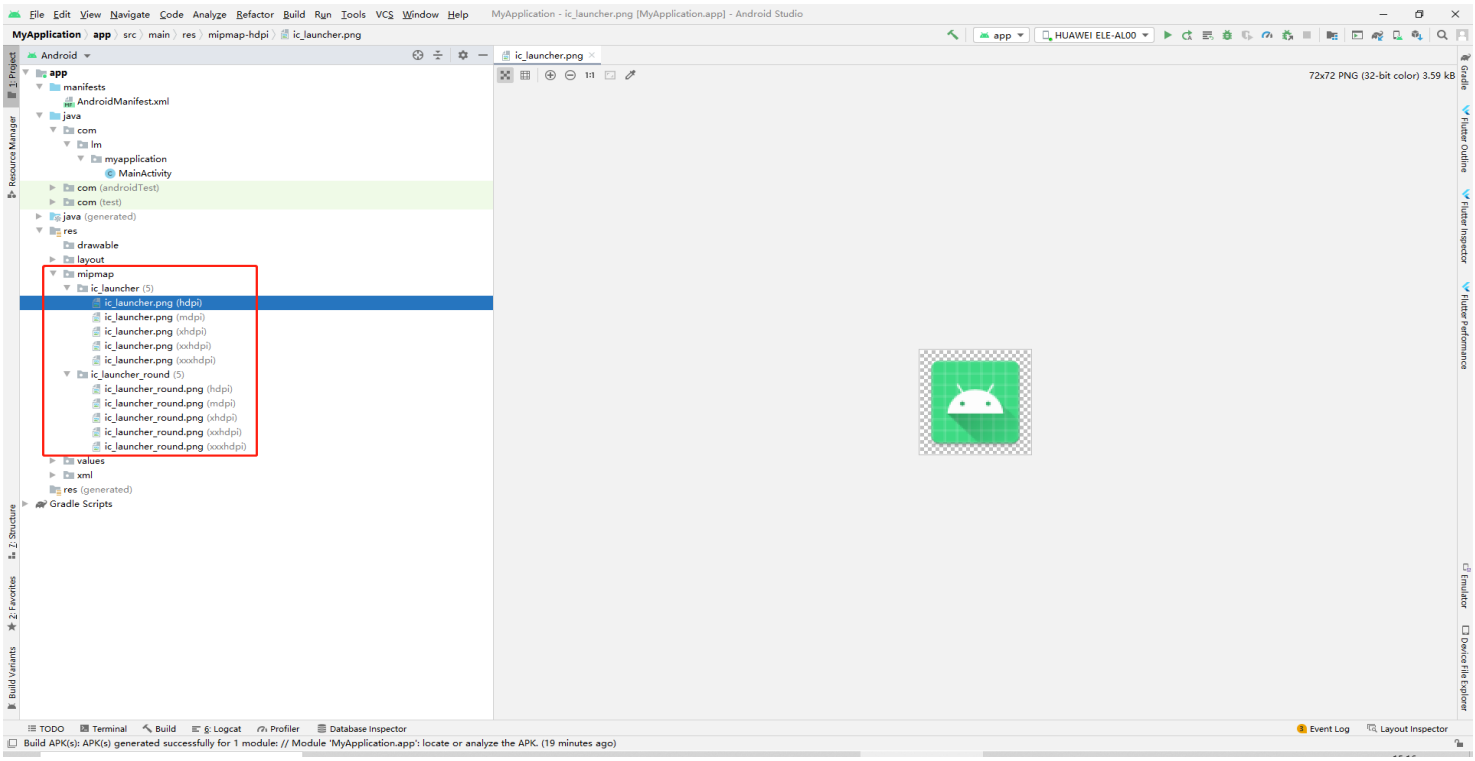
Open Save location: app/src/main/res folder, delete default folder.





Copy our new file and folder to this path

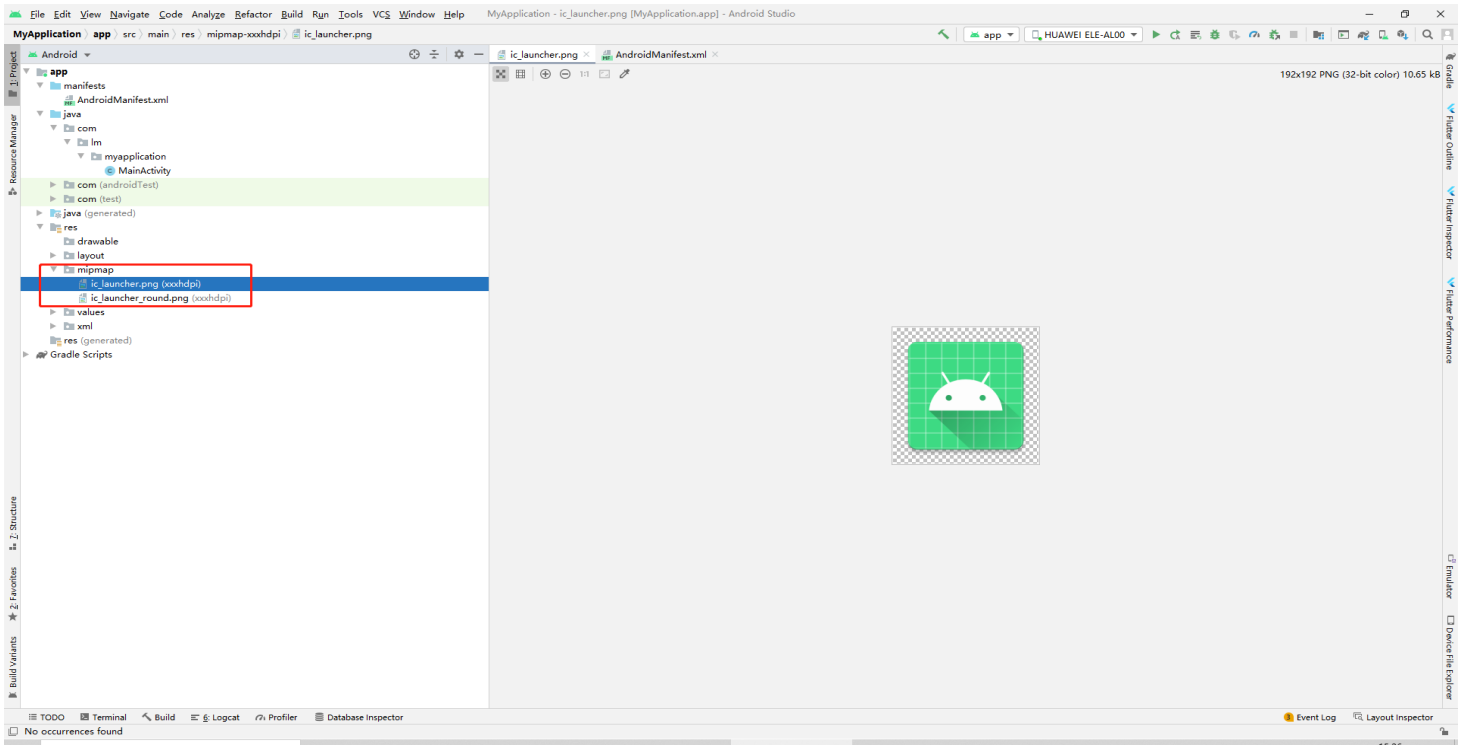




It should be noted that different resolutions are required for different folders.

folder	ico resolutions
mipmap-hdpi	72*72
mipmap-mdpi	48*48
mipmap-xhdpi	96*96
mipmap-xxhdpi	144*144
mipmap-xxxhdpi	192*192

If you not make all types of ico, you only need make ico for “`mipmap-xxhdpi`” or “`mipmap-xxxhdpi`” folder is ok.



After change the ico file, click “`Build-->Build Bunle(s)/APK(s)-->Build APK(s)`” recompile the project to APK file.