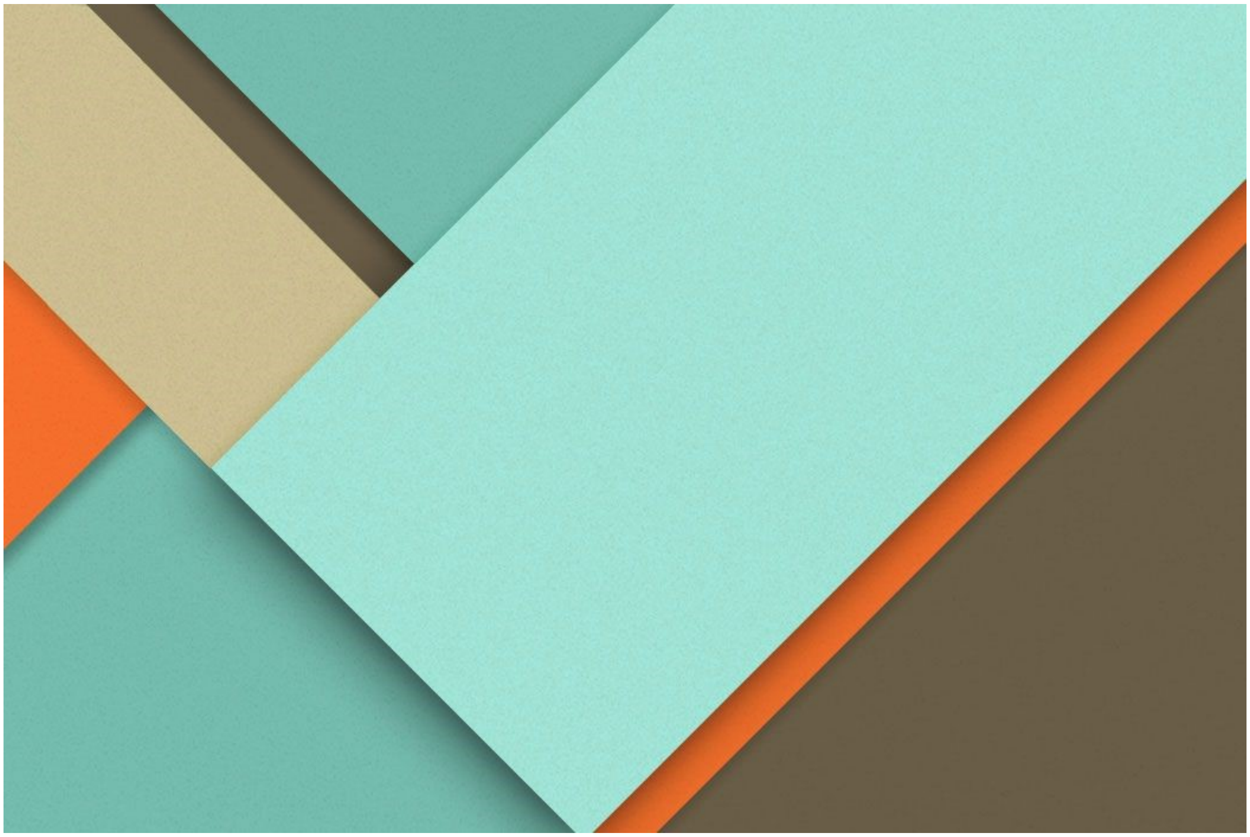


Food Box

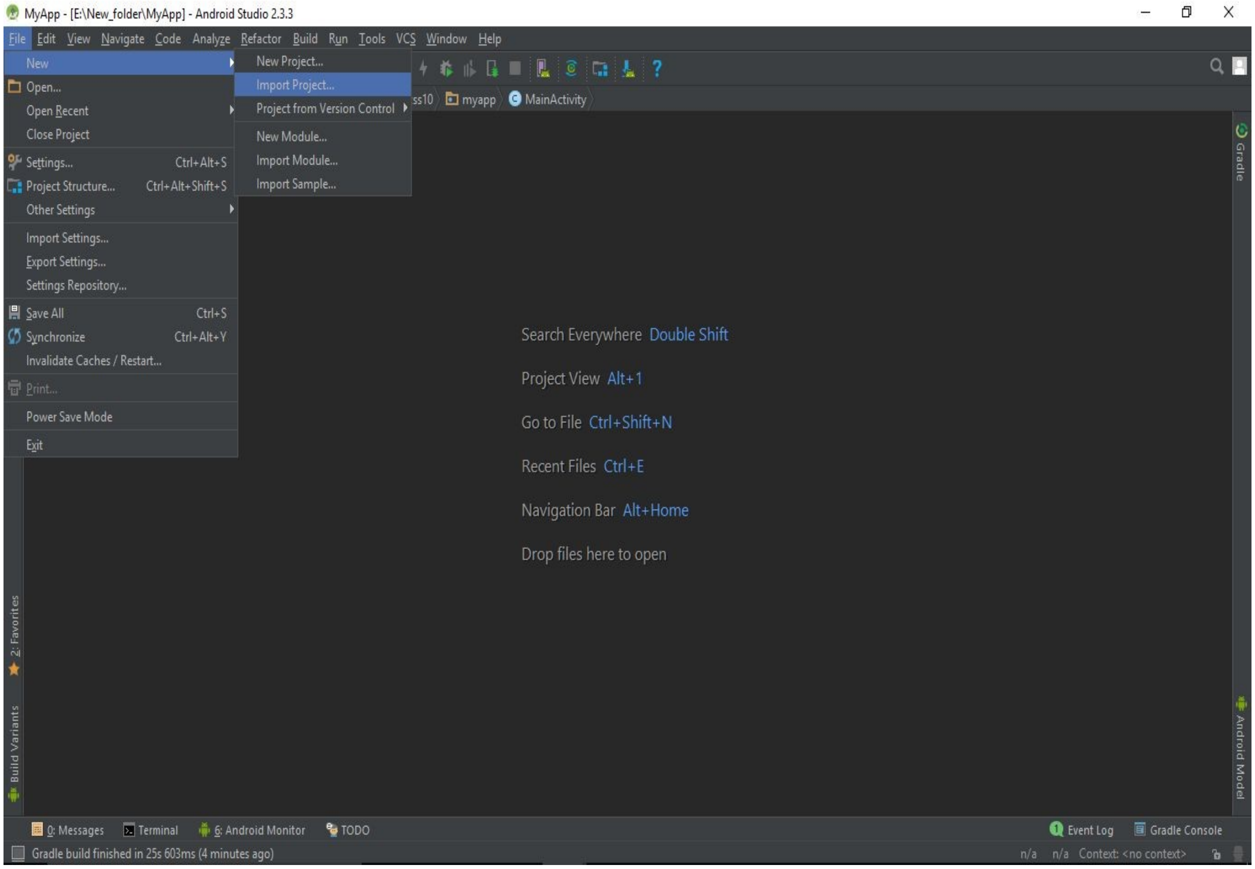
USER

Step 1:

Download [or] Clone the latest and updated source code of Food Box User Android application Source Code from zip to your project root folder.

Step 2:

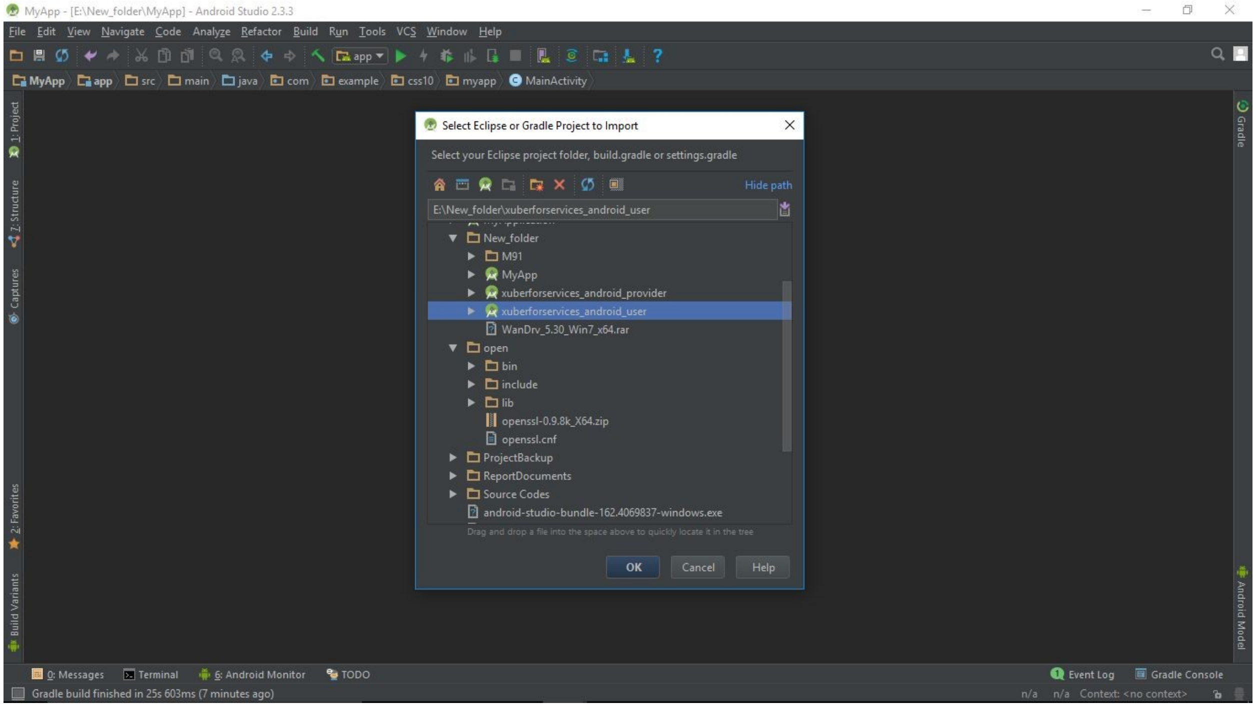
Open your Android Studio. From File options On the top left corner, Choose Import ---> Import Project option.



Gradle build finished in 25s 603ms (4 minutes ago)

Step 3:

A Selection dialog will open. Choose the path from where the project is located.


MyApp - [E:\New_folder\MyApp] - Android Studio 2.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
MyApp app src main java com example css10 myapp MainActivity
1: Project
7: Structure
Captures
2: Favorites
Build Variants
Gradle
Android Model
Hide path
New_folder
M91
MyApp
xuberforservices_android_provider
xuberforservices_android_user
WanDrv_5.30_Win7_x64.rar
open
bin
include
lib
openssl-0.9.8k_X64.zip
openssl.cnf
ProjectBackup
ReportDocuments
Source Codes
android-studio-bundle-162.4069837-windows.exe
OK
Cancel
Help
0: Messages
Terminal
6: Android Monitor
TODO
Event Log
Gradle Console
n/a n/a Context: <no context>

Select Eclipse or Gradle Project to Import

Select your Eclipse project folder, build.gradle or settings.gradle

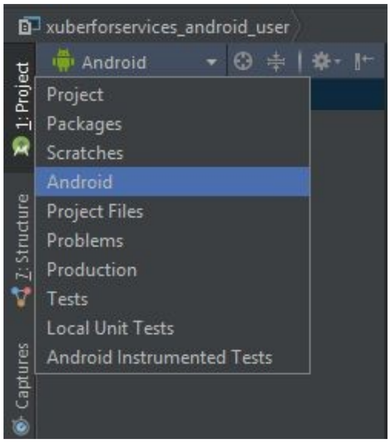
E:\New_folder\xuberforservices_android_user

Drag and drop a file into the space above to quickly locate it in the tree

Gradle build finished in 25s 603ms (7 minutes ago)

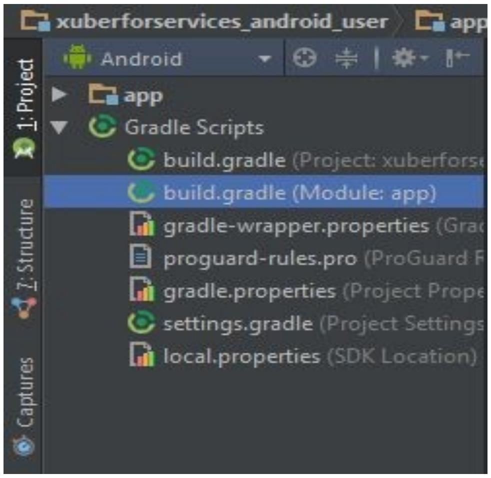
Step 4:

After the project got imported, In the project window, you can customize the files what you viewed. Select Android from the dropdown menu on the left corner, to view all necessary files that you need in your project.



Step 5:

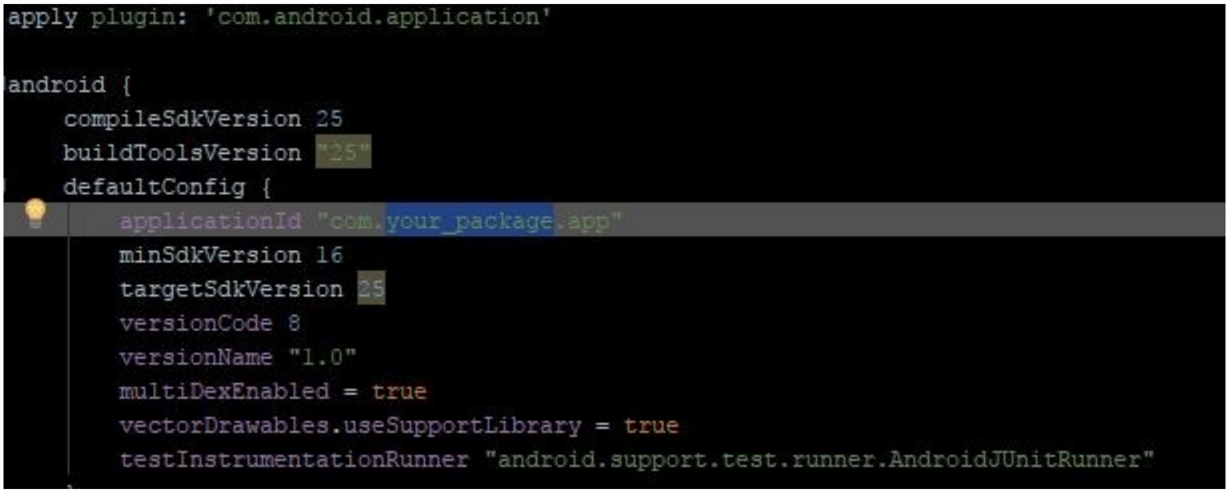
Under Gradle Scripts ---> Select the app's build.gradle file.


xuberforservices_android_user
app
1: Project
7: Structure
Captures
Android
app
Gradle Scripts
build.gradle (Project: xuberforse
build.gradle (Module: app)
gradle-wrapper.properties (Grad
proguard-rules.pro (ProGuard F
gradle.properties (Project Prope
settings.gradle (Project Settings
local.properties (SDK Location)

Step 6:

On app level gradle file replace the applicationId with your package name and sync the project.

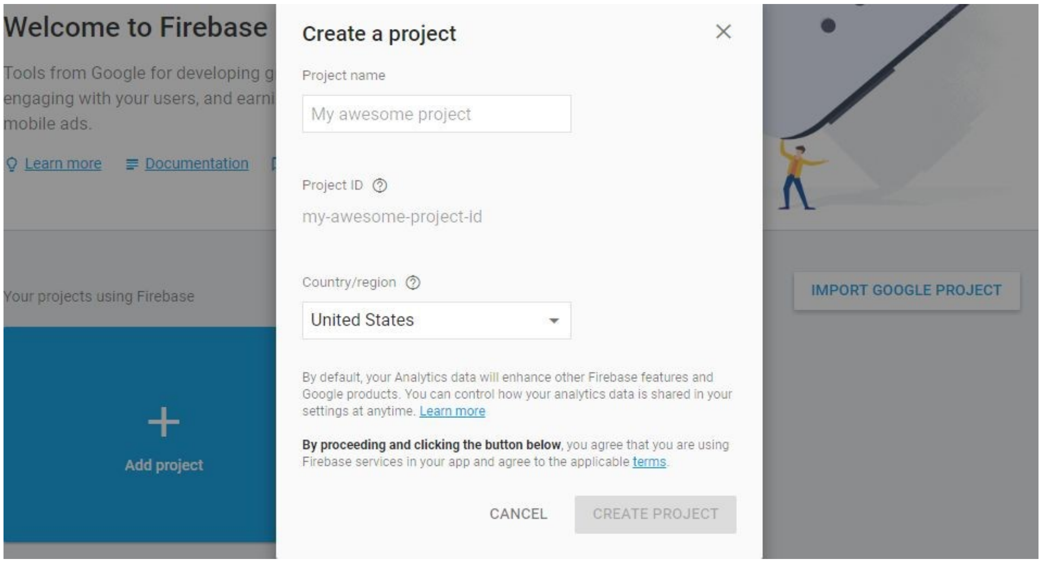
Step



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25"
    defaultConfig {
        applicationId "com.your_package.app"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 8
        versionName "1.0"
        multiDexEnabled = true
        vectorDrawables.useSupportLibrary = true
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
```

7:

To get google_services.json file Sign in to your *account on* https://console.firebase.google.com/ and select Add Project with your Application Name(If Not Exist) and select your country.



Welcome to Firebase
Tools from Google for developing g
engaging with your users, and earni
mobile ads.
Learn more
Documentation
Your projects using Firebase
Add project
IMPORT GOOGLE PROJECT

Create a project

Project name: My awesome project

Project ID: my-awesome-project-id

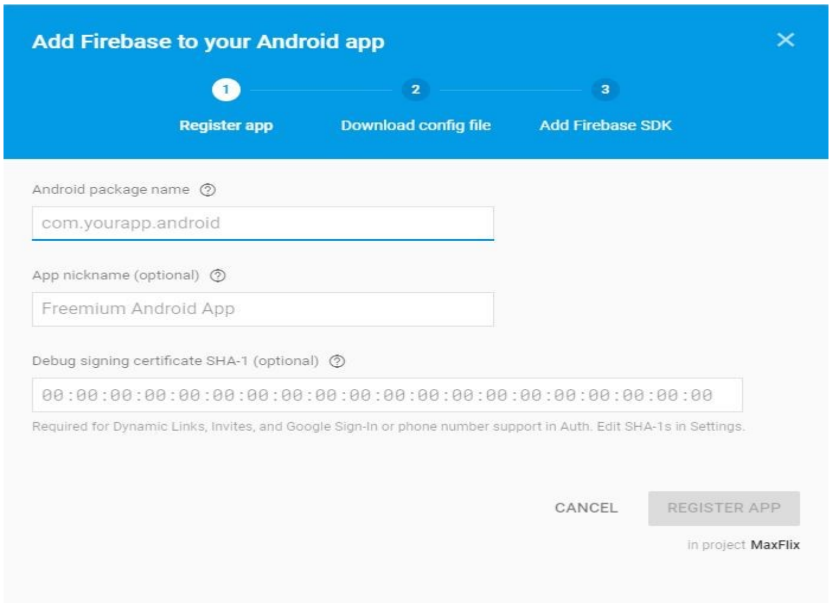
Country/region: United States

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at anytime. Learn more

By proceeding and clicking the button below, you agree that you are using Firebase services in your app and agree to the applicable terms.

Step 8:

Inside your project you can register your android app with your package name, app name(optional) and your own(Android Studio) SHA-1 key.



Add Firebase to your Android app

1 Register app — 2 Download config file — 3 Add Firebase SDK

Android package name: com.yourapp.android

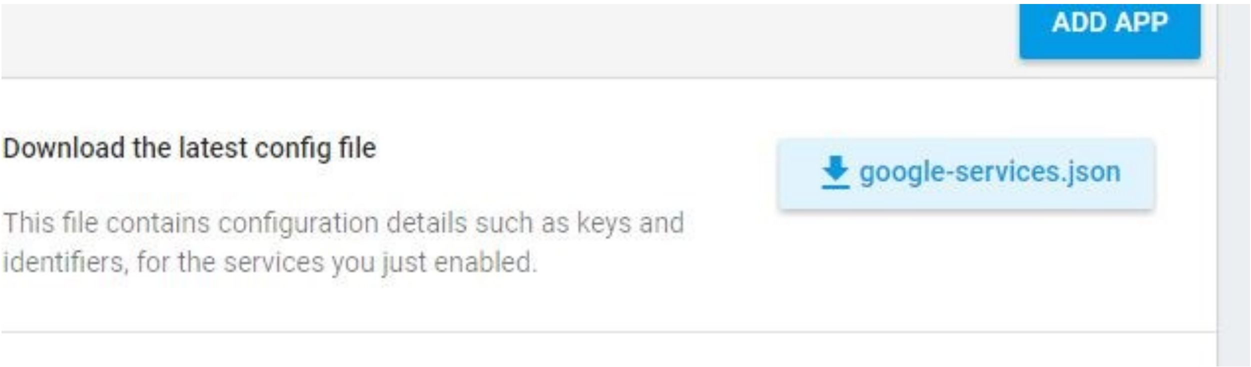
App nickname (optional): Freemium Android App

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Step 9:

After registering App you can see *google-service.json* **file in another popup.**



Download the latest config file

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

Now Download the *google-service.json* **file from your console and finish the steps of app creation and replace that** *google-service.json* **on android studio. Then sync project at android studio.**

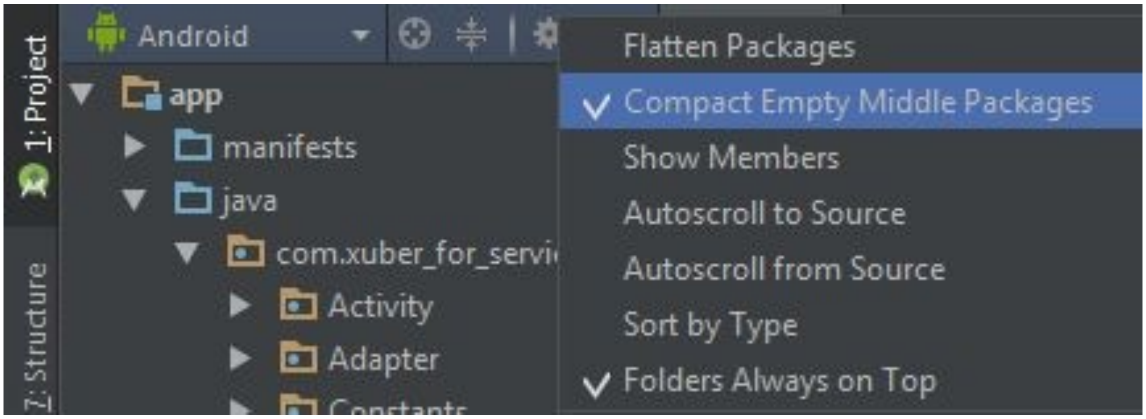
Note : You can download *google-service.json* **file from your console at any time.**

To do that select Settings **option by clicking menu icon of your app(**Which you created already**).**

Step

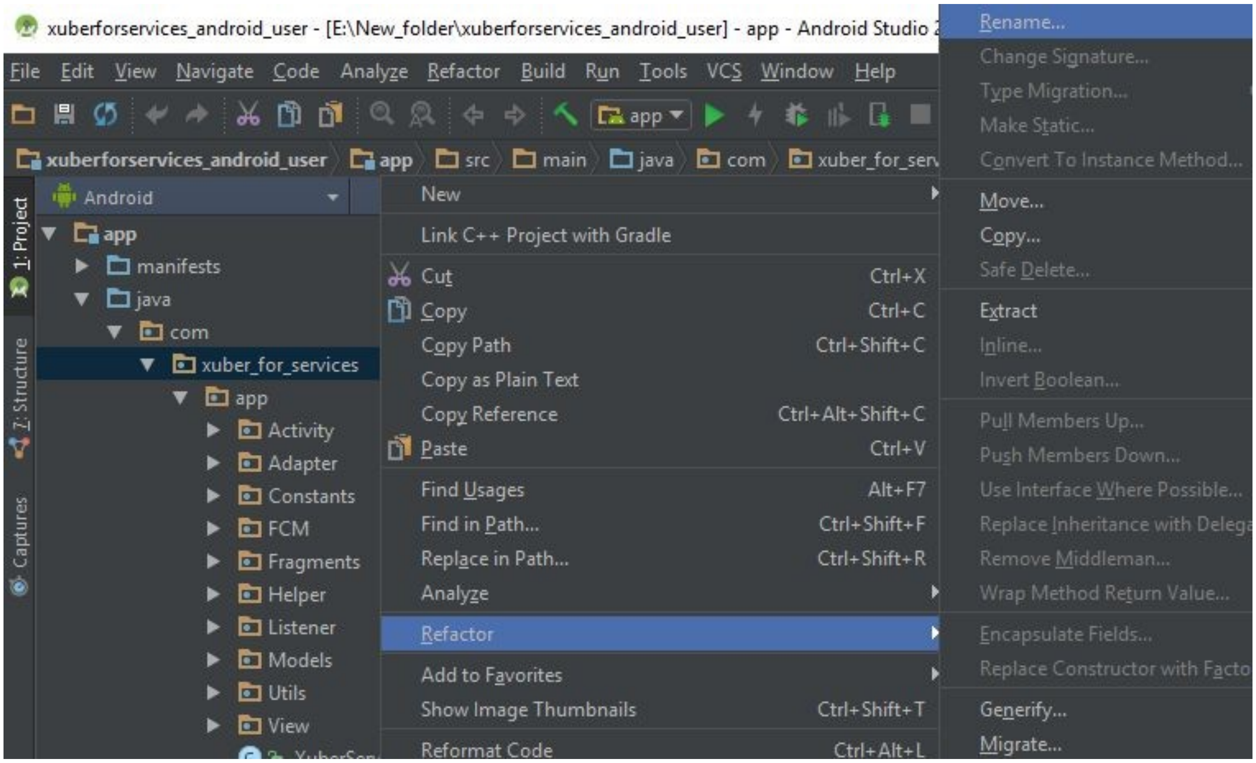
Step 10:

Select Compact Empty middle packages **option by clicking gear icon from android studio's left top corner.**



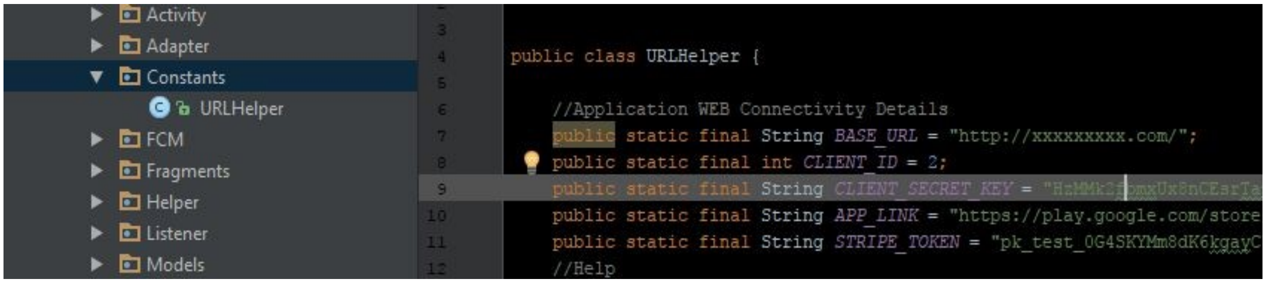
11:

Now you can Rename the package name by right click on the package name -> refactor -> Rename.(Shift+F6)


xuberforservices_android_user - [E:\New_folder\xuberforservices_android_user] - app - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
xuberforservices_android_user app src main java com xuber_for_serv
1: Project
7: Structure
Captures
Android
app
manifests
java
com
xuber_for_services
app
Activity
Adapter
Constants
FCM
Fragments
Helper
Listener
Models
Utils
View
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Rename...
Change Signature...
Type Migration...
Make Static...
Convert To Instance Method...
Move...
Copy...
Safe Delete...
Extract
Inline...
Invert Boolean...
Pull Members Up...
Push Members Down...
Use Interface Where Possible...
Replace Inheritance with Delega
Remove Middleman...
Wrap Method Return Value...
Encapsulate Fields...
Replace Constructor with Facto
Generify...
Migrate...

Step 12:

Open URLHelper java class from Constants package and change BASE_URL, CLIENT_SECRET_KEY and STRIPE_TOKEN with your installation credentials.



```
public class URLHelper {

    //Application WEB Connectivity Details
    public static final String BASE_URL = "http://xxxxxxxxx.com/";
    public static final int CLIENT_ID = 2;
    public static final String CLIENT_SECRET_KEY = "HzMMk2fomxUx8nCEarTa
    public static final String APP_LINK = "https://play.google.com/store
    public static final String STRIPE_TOKEN = "pk_test_0G4SKYMm8dK6kgayC
    //Help
```

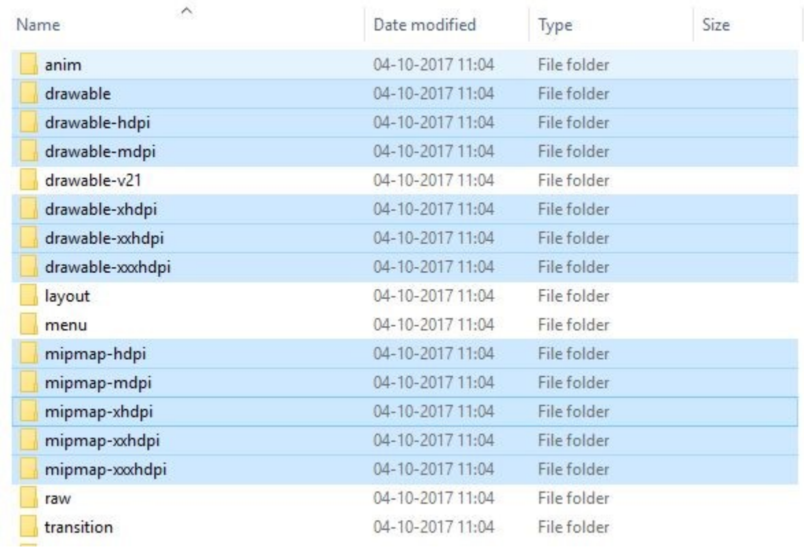
Step 13:

On your root folder Go to app -> src -> main -> res

Step 14:

Change image files from drawable and mipmap folders of res directory to your project's relevant images.

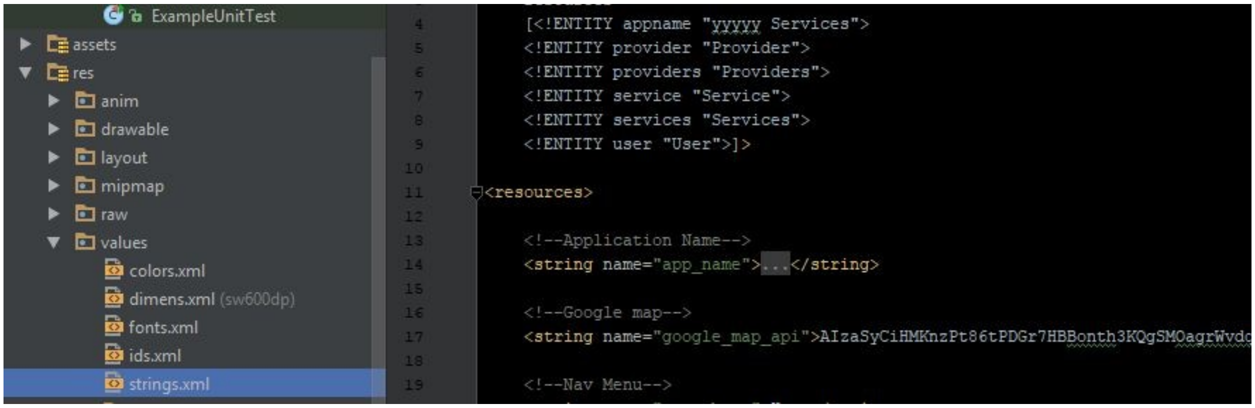
Step



Name	Date modified	Type	Size
anim	04-10-2017 11:04	File folder	
drawable	04-10-2017 11:04	File folder	
drawable-hdpi	04-10-2017 11:04	File folder	
drawable-mdpi	04-10-2017 11:04	File folder	
drawable-v21	04-10-2017 11:04	File folder	
drawable-xhdpi	04-10-2017 11:04	File folder	
drawable-xxhdpi	04-10-2017 11:04	File folder	
drawable-xxxhdpi	04-10-2017 11:04	File folder	
layout	04-10-2017 11:04	File folder	
menu	04-10-2017 11:04	File folder	
mipmap-hdpi	04-10-2017 11:04	File folder	
mipmap-mdpi	04-10-2017 11:04	File folder	
mipmap-xhdpi	04-10-2017 11:04	File folder	
mipmap-xxhdpi	04-10-2017 11:04	File folder	
mipmap-xxxhdpi	04-10-2017 11:04	File folder	
raw	04-10-2017 11:04	File folder	
transition	04-10-2017 11:04	File folder	

Step 15:

At res directory, Change app_name and google_map_api at strings.xml file of values package.



```
    [<!ENTITY appname "yyyyy Services">
    <!ENTITY provider "Provider">
    <!ENTITY providers "Providers">
    <!ENTITY service "Service">
    <!ENTITY services "Services">
    <!ENTITY user "User">]>

<resources>

    <!--Application Name-->
    <string name="app_name">...</string>

    <!--Google map-->
    <string name="google_map_api">AIzaSyCiHMKnzPt86tPDGr7HBBonth3KQgSM0agrWvdc

    <!--Nav Menu-->
```

16:

To get google_map_api sign into https://console.developers.google.com Google API developer console. Create a project and choose API Key. A new key will be generated. Replace it with your google_map_api in strings.xml.

DELIVERYBOY

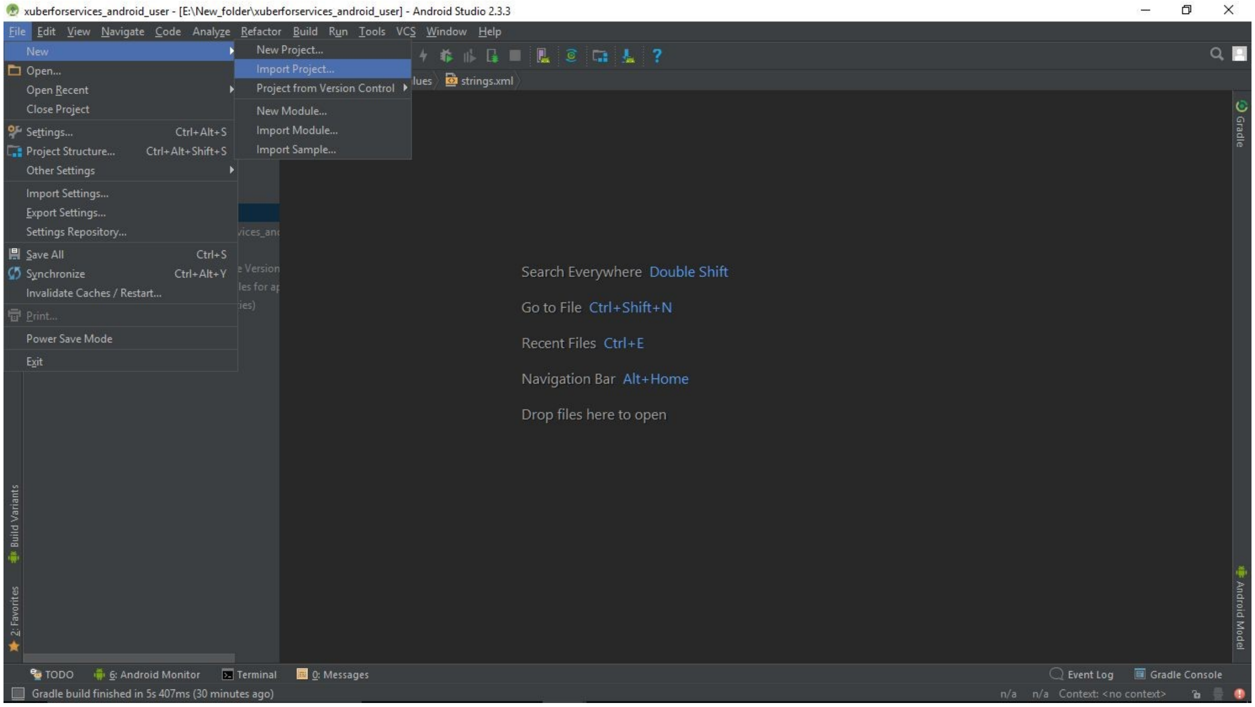
Step 1:

Download [or] Clone the latest and updated source code of Food Box Deliveryboy Android application Source Code from zip to your project root folder.

Step

Step 2:

Open your Android Studio. From File options On the top left corner, Choose Import ---> Import Project option.



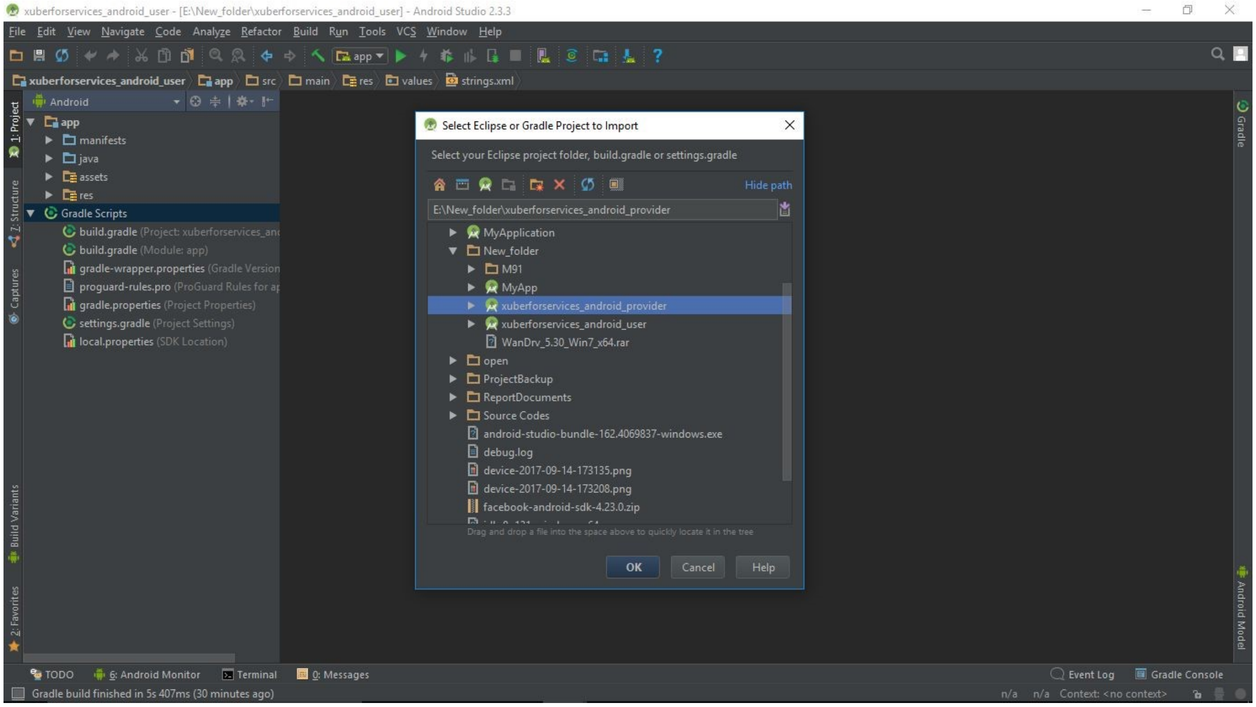
xuberforservices_android_user - [E:\New_folder\xuberforservices_android_user] - Android Studio 2.3.3

Gradle build finished in 5s 407ms (30 minutes ago)

Step

3:

A Selection dialog will open. Choose the path from where the project is located.



xuberforservices_android_user - [E:\New_folder\xuberforservices_android_user] - Android Studio 2.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
xuberforservices_android_user app src main res values strings.xml
Android
app
manifests
java
assets
res
Gradle Scripts
build.gradle (Project: xuberforservices_an
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version
proguard-rules.pro (ProGuard Rules for ap
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model
TODO
6: Android Monitor
Terminal
0: Messages
Event Log
Gradle Console
n/a n/a Context: <no context>

Select Eclipse or Gradle Project to Import

Select your Eclipse project folder, build.gradle or settings.gradle

Hide path

E:\New_folder\xuberforservices_android_provider

```
MyApplication
New_folder
    M91
    MyApp
    xuberforservices_android_provider
    xuberforservices_android_user
    WanDrv_5.30_Win7_x64.rar
open
ProjectBackup
ReportDocuments
Source Codes
android-studio-bundle-162.4069837-windows.exe
debug.log
device-2017-09-14-173135.png
device-2017-09-14-173208.png
facebook-android-sdk-4.23.0.zip
```

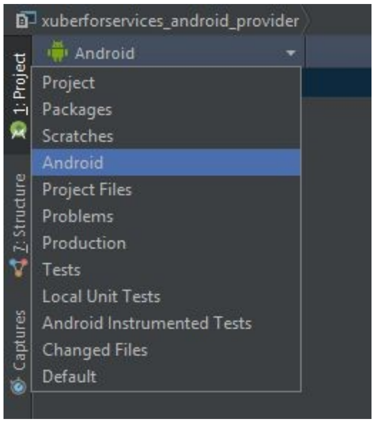
Drag and drop a file into the space above to quickly locate it in the tree

Gradle build finished in 5s 407ms (30 minutes ago)

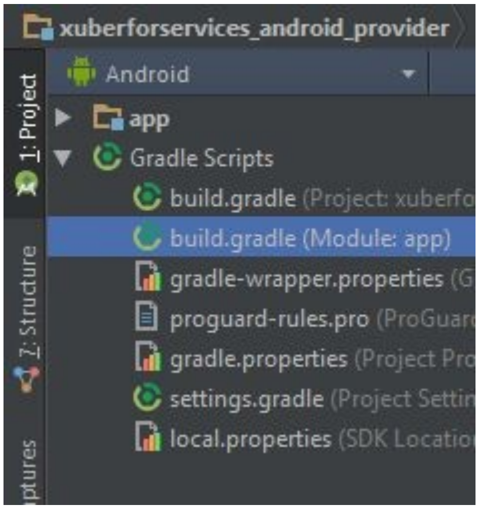
Step 4:

After the project got imported, In the project window, you can customize the files what you viewed. Select Android from the dropdown menu on the left corner, to view all necessary files that you need in your project.

Step 5:



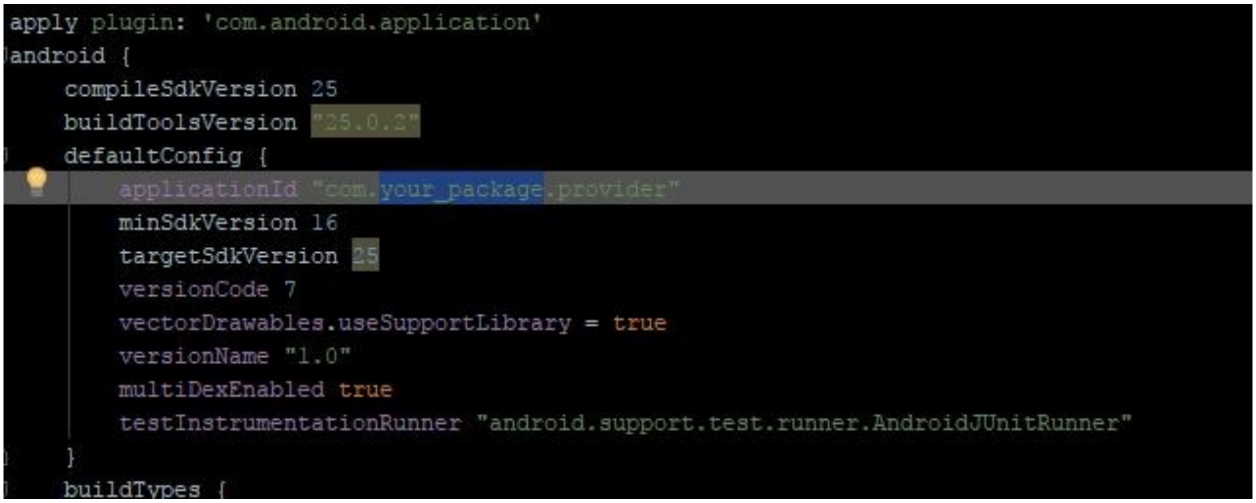
Under Gradle Scripts ---> Select the app's build.gradle file.


xuberforservices_android_provider
Android
app
Gradle Scripts
build.gradle (Project: xuberfo
build.gradle (Module: app)
gradle-wrapper.properties (G
proguard-rules.pro (ProGuard
gradle.properties (Project Pro
settings.gradle (Project Settin
local.properties (SDK Locatio
1: Project
7: Structure
ptures

Step 6:

On app level gradle file replace the applicationId with your package name and sync the project.

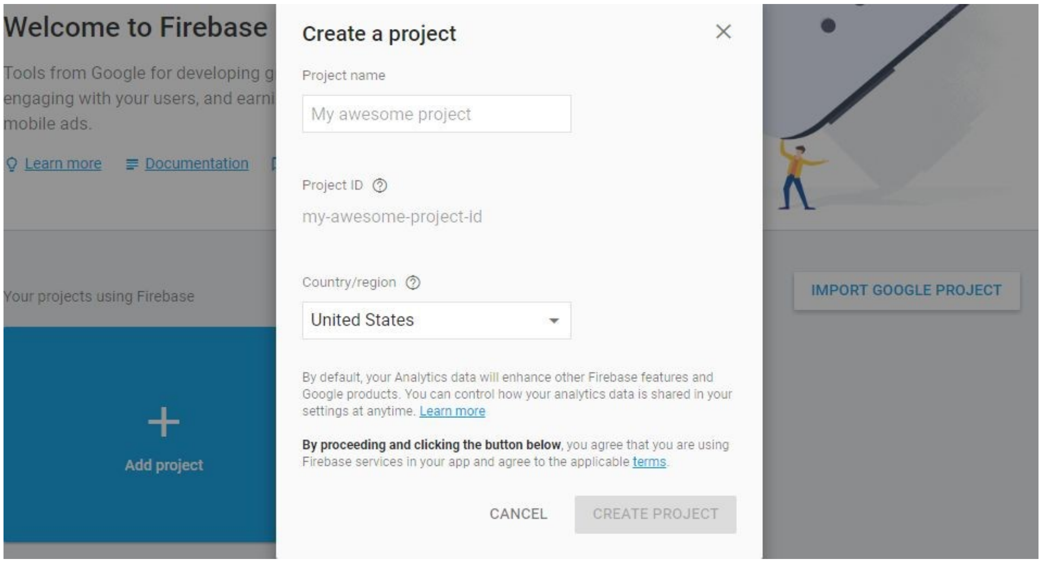
Step



```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.your_package.provider"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 7
        vectorDrawables.useSupportLibrary = true
        versionName "1.0"
        multiDexEnabled true
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
```

7:

To get google_services.json file Sign in to your account on https://console.firebase.google.com/ and select Add Project with your Application Name(If Not Exist) and select your country.



Welcome to Firebase

Tools from Google for developing g
engaging with your users, and earni
mobile ads.

Your projects using Firebase

Create a project

Project name: My awesome project

Project ID: my-awesome-project-id

Country/region: United States

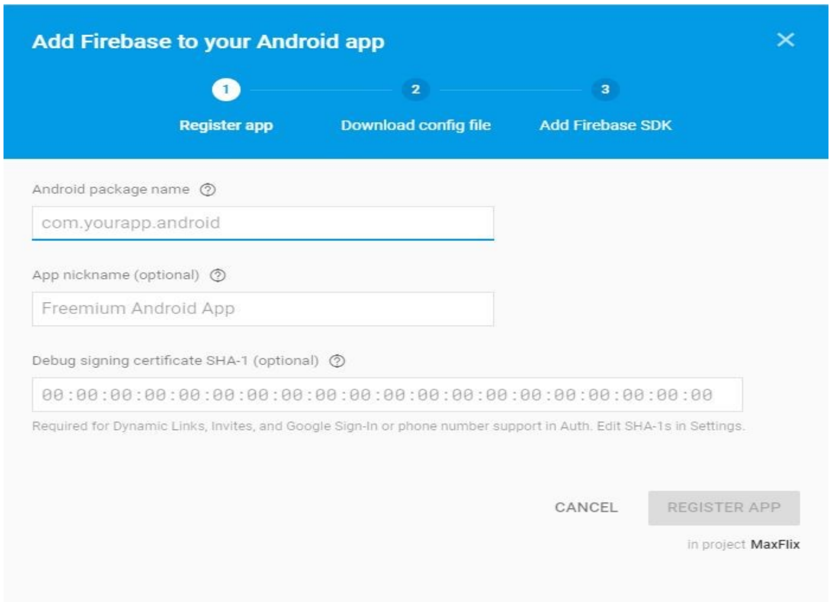
By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at anytime. Learn more

By proceeding and clicking the button below, you agree that you are using Firebase services in your app and agree to the applicable terms.

Step

Step 8:

Inside your project you can register your android app with your package name, app name(optional) and your own(Android Studio) SHA-1 key.



Android package name
com.yourapp.android

App nickname (optional)
Freemium Android App

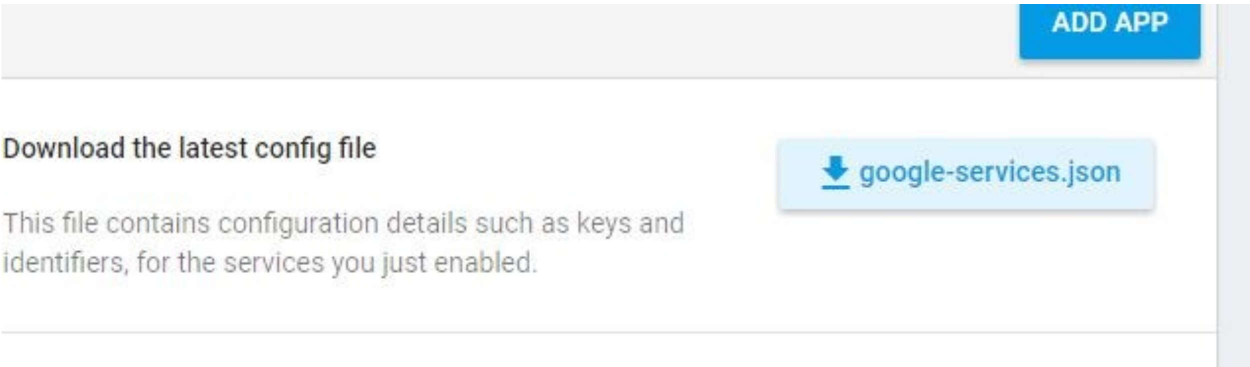
Debug signing certificate SHA-1 (optional)
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

in project MaxFlix

9:

After registering App you can see *google-service.json* file in another popup.



Download the latest config file

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

Now Download the *google-service.json* file from your console and finish the steps of app creation and replace that *google-service.json* on android studio. Then sync project at android studio.

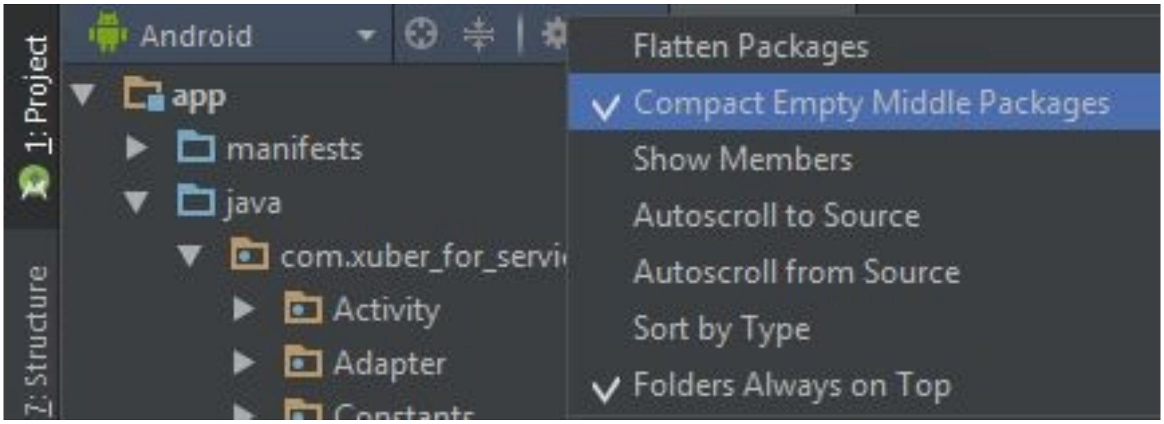
Note : You can download *google-service.json* file from your console at any time.

Step

To do that select Settings **option by clicking menu icon of your app(**Which you created already**).**

Step 10:

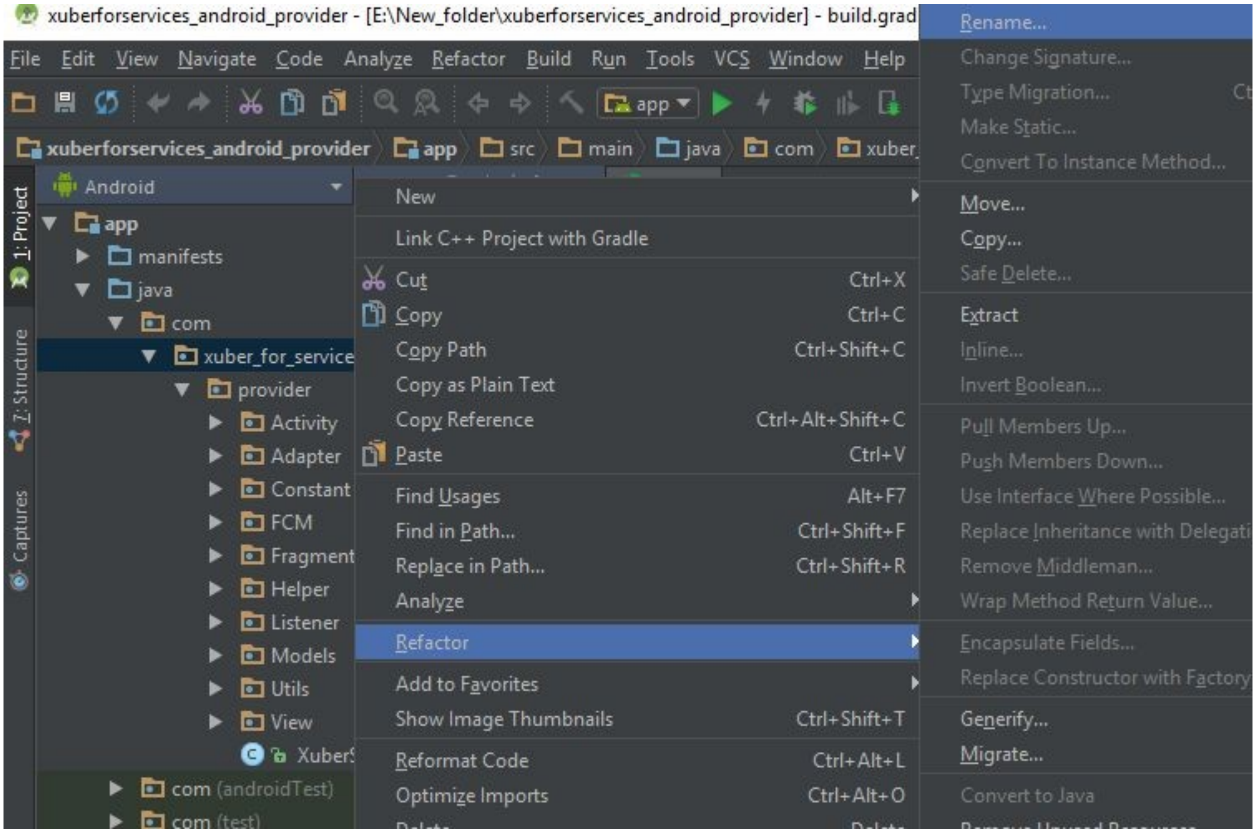
Select Compact Empty middle packages option by clicking gear icon from android studio's left top corner.



11:

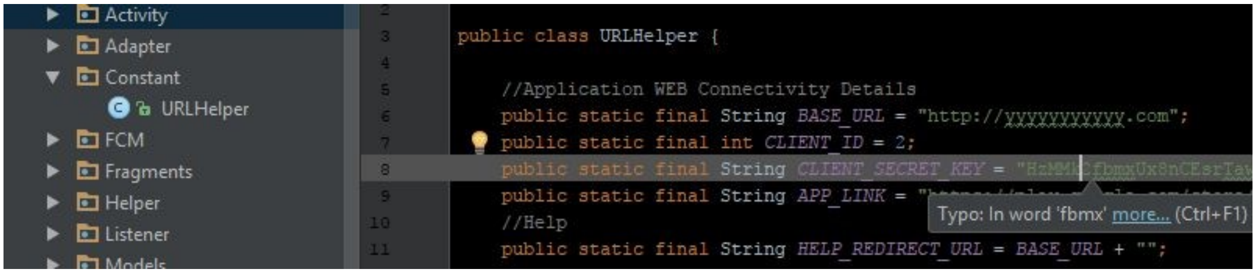
Now you can Rename the package name by right click on the package name -> refactor -> Rename.(Shift+F6)

Step


xuberforservices_android_provider - [E:\New_folder\xuberforservices_android_provider] - build.grad
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
xuberforservices_android_provider app src main java com xuber
1: Project
7: Structure
Captures
Android
app
manifests
java
com
xuber_for_service
provider
Activity
Adapter
Constant
FCM
Fragment
Helper
Listener
Models
Utils
View
XuberS
com (androidTest)
com (test)
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Rename...
Change Signature...
Type Migration... Ct
Make Static...
Convert To Instance Method...
Move...
Copy...
Safe Delete...
Extract
Inline...
Invert Boolean...
Pull Members Up...
Push Members Down...
Use Interface Where Possible...
Replace Inheritance with Delegati
Remove Middleman...
Wrap Method Return Value...
Encapsulate Fields...
Replace Constructor with Factory
Generify...
Migrate...
Convert to Java

Step 12:

Open URLHelper java class from Constants package and change BASE_URL and CLIENT_SECRET_KEY with your installation credentials.



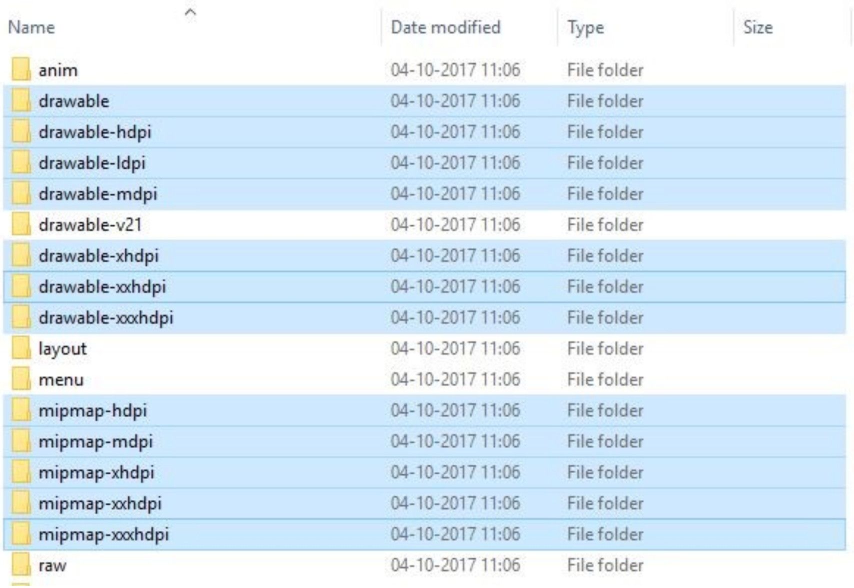
```
public class URLHelper {

    //Application WEB Connectivity Details
    public static final String BASE_URL = "http://yyyyyyyyyyy.com";
    public static final int CLIENT_ID = 2;
    public static final String CLIENT_SECRET_KEY = "HzMMkCfbmxUx8nCEsrIaw
    public static final String APP_LINK = "
    //Help
    public static final String HELP_REDIRECT_URL = BASE_URL + "";
```

13:   Step 14:

On your root folder Go to app -> src -> main -> res

Step

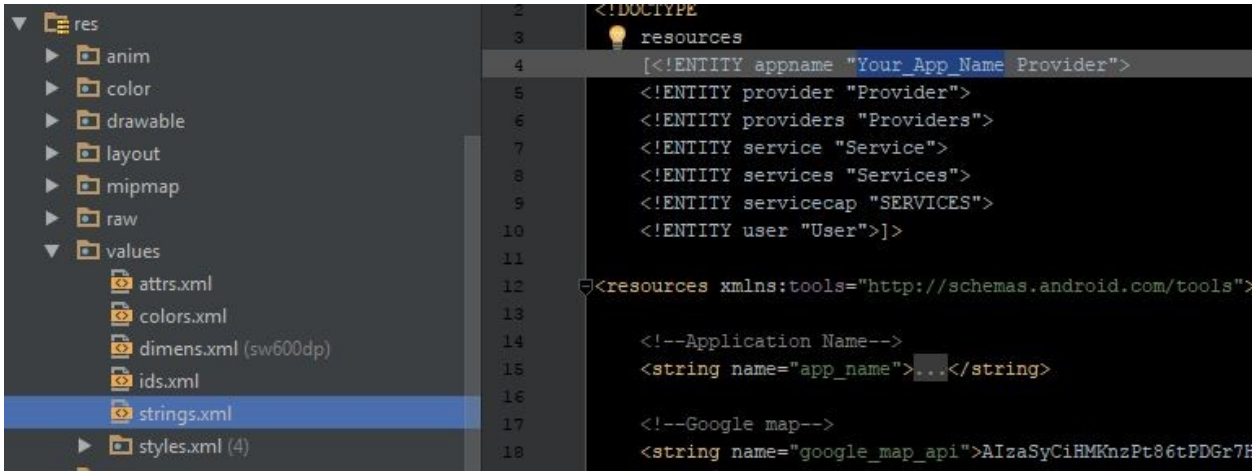
Change image files from drawable and mipmap folders of res directory to your project's relevant images.



Name	Date modified	Type	Size
anim	04-10-2017 11:06	File folder	
drawable	04-10-2017 11:06	File folder	
drawable-hdpi	04-10-2017 11:06	File folder	
drawable-ldpi	04-10-2017 11:06	File folder	
drawable-mdpi	04-10-2017 11:06	File folder	
drawable-v21	04-10-2017 11:06	File folder	
drawable-xhdpi	04-10-2017 11:06	File folder	
drawable-xxhdpi	04-10-2017 11:06	File folder	
drawable-xxxhdpi	04-10-2017 11:06	File folder	
layout	04-10-2017 11:06	File folder	
menu	04-10-2017 11:06	File folder	
mipmap-hdpi	04-10-2017 11:06	File folder	
mipmap-mdpi	04-10-2017 11:06	File folder	
mipmap-xhdpi	04-10-2017 11:06	File folder	
mipmap-xxhdpi	04-10-2017 11:06	File folder	
mipmap-xxxhdpi	04-10-2017 11:06	File folder	
raw	04-10-2017 11:06	File folder	

Step 15:

At res directory, Change appname, ACCOUNT_KIT_CLIENT_TOKEN and google_map_api at string.xml file of values package.



```
<!DOCTYPE
    resources
    [<!ENTITY appname "Your_App_Name Provider">
    <!ENTITY provider "Provider">
    <!ENTITY providers "Providers">
    <!ENTITY service "Service">
    <!ENTITY services "Services">
    <!ENTITY servicecap "SERVICES">
    <!ENTITY user "User">]>

<resources xmlns:tools="http://schemas.android.com/tools">

    <!--Application Name-->
    <string name="app_name">...</string>

    <!--Google map-->
    <string name="google_map_api">AIzaSyCiHMKnzPt86tPDGr7H
```

Step 16:

To get google_map_api sign into https://console.developers.google.com Google API developer console. Create a project and choose API Key. A new key will be generated. Replace it with your google_map_api in strings.xml.

WEB PANEL

Step 1:

Download [or] Clone the latest and updated source code of Food Box Deliveryboy Android application Source Code from zip to your server root folder.

Step 2:

Before setting up Thinkin Cab, we need the server to have the following prerequisite software's or frameworks in your server to build a conceivable environment.

LAMP Stack:

The server environment should have Linux Operating System with Apache Server and MySQL database, PHP server scripting language.

The following are the compatible versions of the LAMP stack

a. **Linux - Ubuntu - 14.04 or 16.04 (LTS is Preferred) or Equivalent**
b. **Apache >= 2.4.25**
c. **MySQL >= 5.7**
d. **PHP >= 7**

Additional PHP Modules required.

1. OpenSSL PHP Extension
2. PDO PHP Extension
3. Mbstring PHP Extension
4. Tokenizer PHP Extension
5. XML PHP Extension Apache modules

1. Rewrite Module

PubNub :

Source to Setup PubNub: https://www.pubnub.com/docs

Domain Name:

It is recommended that you get a domain name and an SSL certificate for the same for our application to work with full functionality.

SSL Certificate:

To maintain compatibility of the application across the web app along with live tracking you need to have SSL certificate to fetch the user location from the browser.

Composer:

Composer is required to download the dependencies for the application.

INSTALLATION

1. **Unzip the code in the server.**
2. **Set the following permissions**
    a. **sudo chgrp -R www-data storage bootstrap/cache public**
    b. **sudo chmod -R ug+rwx storage bootstrap/cache public**

    To the following folders

    c. **./bootstrap**
    d. **./storage**
    e. **./public**
3. **Now edit the .env file in server folder**

```
DB_HOST = localhost (Provide Database host URL here)
DB_PORT = 3306 (Provide Database port here)
DB_DATABASE = xuber (Provide the Database name from step-5)
DB_USERNAME = root (Database username)
DB_PASSWORD = (Database password)
```

For mail delivery integration using gmail, If you use other mail deliveryboys, please get the details below by contacting them

```
MAIL_DRIVER = smtp

MAIL_HOST = smtp.gmail.com

MAIL_PORT = 587

MAIL_USERNAME = (Provide your gmail username) MAIL_PASSWORD = (Provide
your gmail password)

MAIL_ENCRYPTION = tls
```

If for Paypal payment integration to receive payments from users,

```
PAYPAL_CLIENT_ID = your_paypal_client_id

PAYPAL_SECRET = your_paypal_secret
```

To integrate Google Maps in the app you'll have to create an API key

```
GOOGLE_API_KEY = your_google_maps_api_key
```

4. **Run the following commands to complete setup**
    a. **composer install**
    b. **php artisan key:generate**

c. **php artisan migrate --seed**
d. **php artisan storage:link**

5. **Configure virtual hosts to direct to the subdirectory public inside the application Reference: https://www.cloudways.com/en/?id=315139**
6. **Configure SSL certificate for getting location in browser.**
   Reference: http://support.cloudways.com/how-can-i-getdeploy-an-ssl-certificate-for-my-application

7. **By now the application should be live and ready for testing.**
8. **You can reach the admin panel from this URL yourdomain.com/admin/login**
   a. **Default admin credentials are**

      Username: admin@foodbox.com

      Password: 123456

9. **You'll also have demo accounts to access the user and deliveryboy panels, which you might like to disable while moving the application to production**
   a. **Default user and deliveryboy credentials are**

      Username: demo@foodbox.com

      Password: 123456

Thank You..