



# Taxi Time: Documentation

## [ How to get Google Map Api Key ]

Google Maps has been used for some of application functionalities. you will need to retrieve an API key from Google to have the app working.

## Maps console

---

Go to **Google Maps Developers Console** and select taxi project which was created in previous section.

### Warning

Having Billing enabled on your Google Account is mandatory for some API's to work. So make sure you have the billing enabled or there might be some parts don't work as expected.

---

# Enabling APIs

You need to have these APIs enabled in library section one by one:

- Maps SDK for Android
- Maps Static API
- Direction API
- Places API
- Distance Matrix API
- Geocoding API
- Maps JavaScript API

## [ How to Setup Backend & Web Panel ]

(Server Requirements: PHP Version -- 7.0)

We Recommend to use VPS Instead of Shared Hosting Even You are free to use Cpanel or any panel at VPS

- 1. Download taxitimeBackend.zip File.
- 2. Upload to your server using FTP or FileManager.
- 3. Extract the zip file.
- 4. If dot files are hidden, First Unhide (Mostly in Cpanel)
- 5. Create A Database in your Hosting then open .env file.
- . In .env file change your Database credentials `DB_DATABASE="your db name"`

```
DB_USERNAME="your db user"  
DB_PASSWORD="your db password"
```

- 7. Go to cabuser/includes/Constants.php file
- . In Constants.php file also change your Database credentials

```
$dbName = 'your fb name' ;  
$dbUser = 'your db user' ;  
$dbPass = 'your db password' ;
```

**Note: In Both .env & Constants.php file database credential will be same.**

- 9. Download taxitime.sql from files
  - 10. Upload/ Import SQL file to your database via PhpMyAdmin
-

“

”

*NOTE: (Optional) How to show hidden files (.htaccess/.env) in*

### *cPanel File Manager*

Many users ask us how they can see the .htaccess, .env files for their websites in cPanel File Manager. By default, such files are not shown. To see the hidden files in cPanel File Manager follow these

**Steps:**

- 1. Log into your cPanel account • 2. Go to File Manager (Files Section) • 3. Click the top-right Settings button.
- 4. Then on the Preferences window, check the option Show Hidden Files (dotfiles)
- 5. Confirm the action by clicking the Save button
- . Now you will see all hidden files (including .htaccess , .env files) in File Manager

Then Admin Panel will look Like

**Admin Panel URL:** <http://yourdomain.com/admin/login>

**Default Admin ID & Password:**

**ID:** admin@taxitime.com

**Password:** 12345

Then you can change it, In Profile tab.

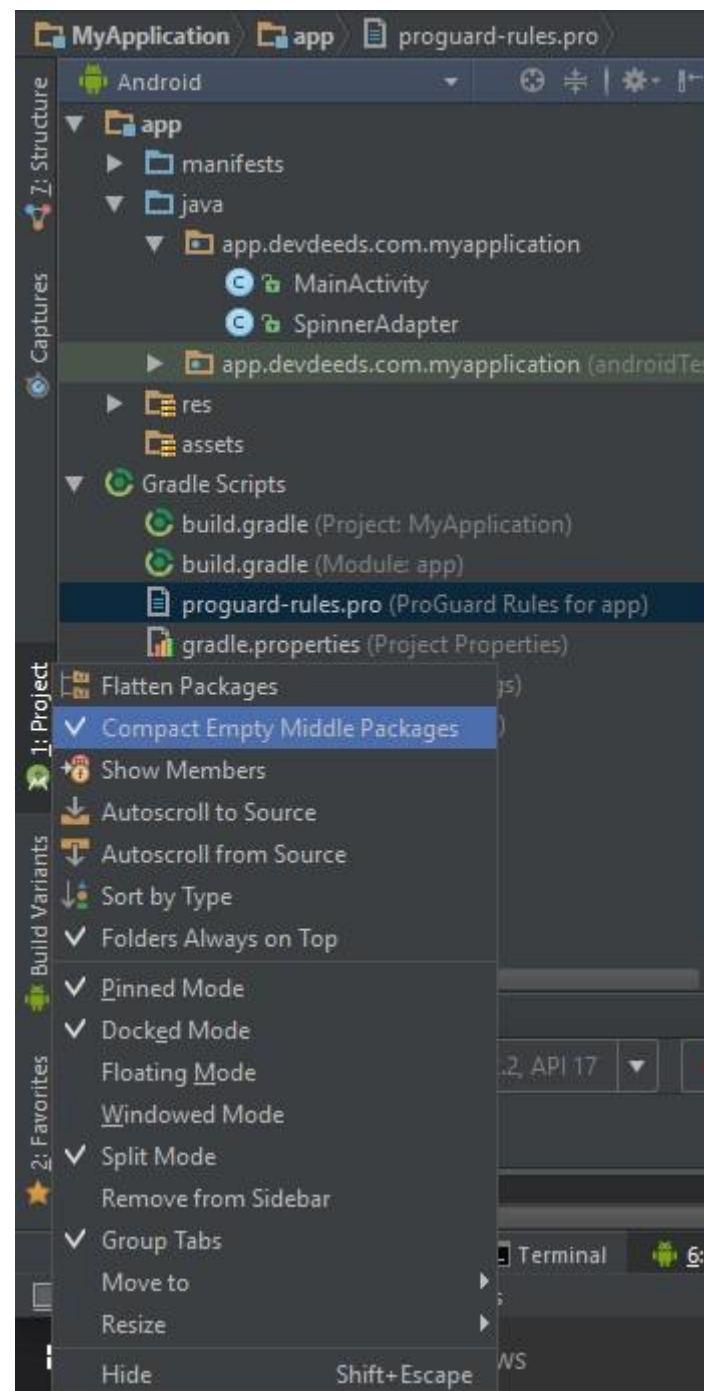
**\*\*\* Backend Setup Done \*\*\***

## [ How to Setup Backend & Web Panel ]

How to change package name In Android App (Same Steps for both user & driver app)

### Step 1: Switch View To Compact Empty Middle Packages

---

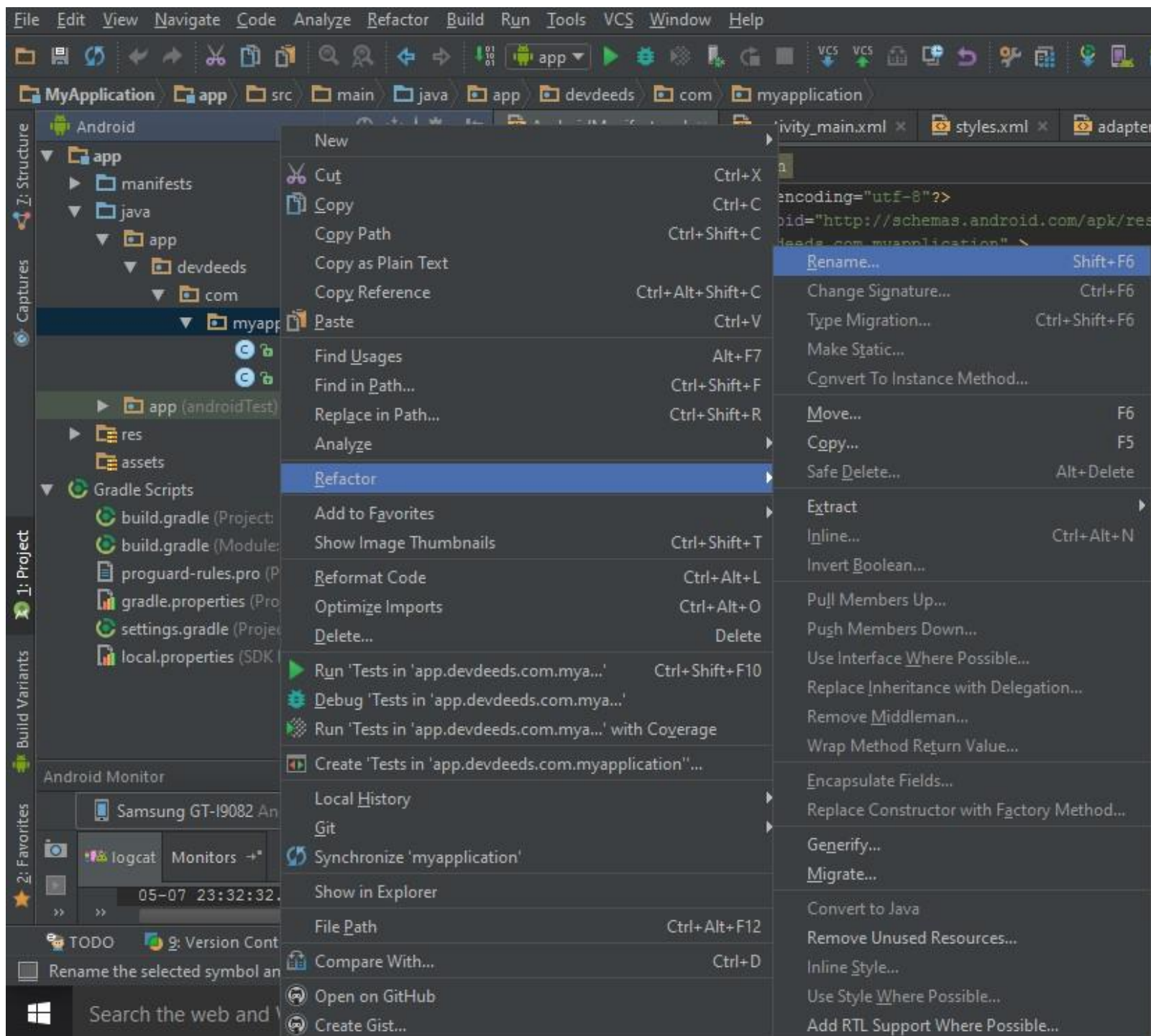


This action will split your packages into sub packages.

## Step 2: Rename Package Using Refactor Tool

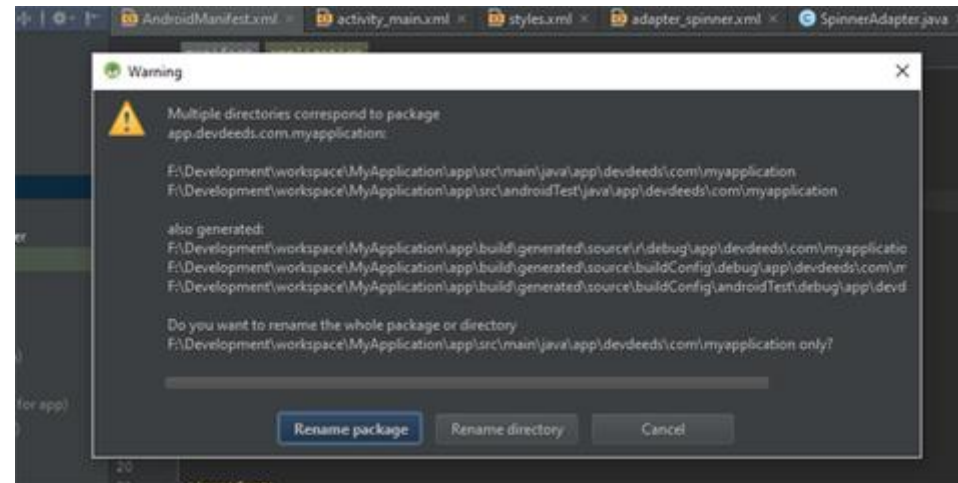
---

Right Click on project explorer menu and you can see Compact Empty Middle Packages. Select this option. After selection the explorer menu will be changed and all the packages are now separated for individual refactor. Refactor tool will change the name of package in source code files.

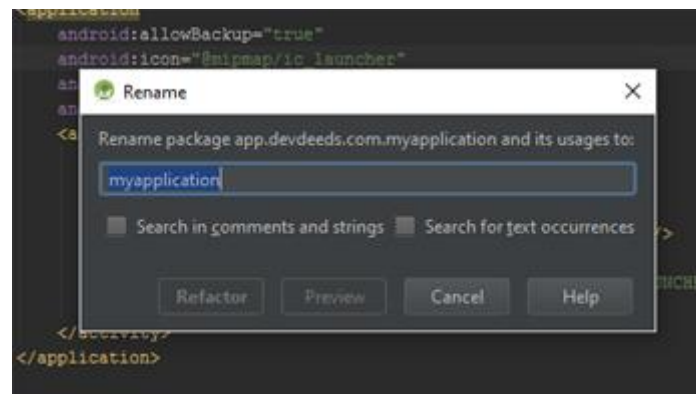




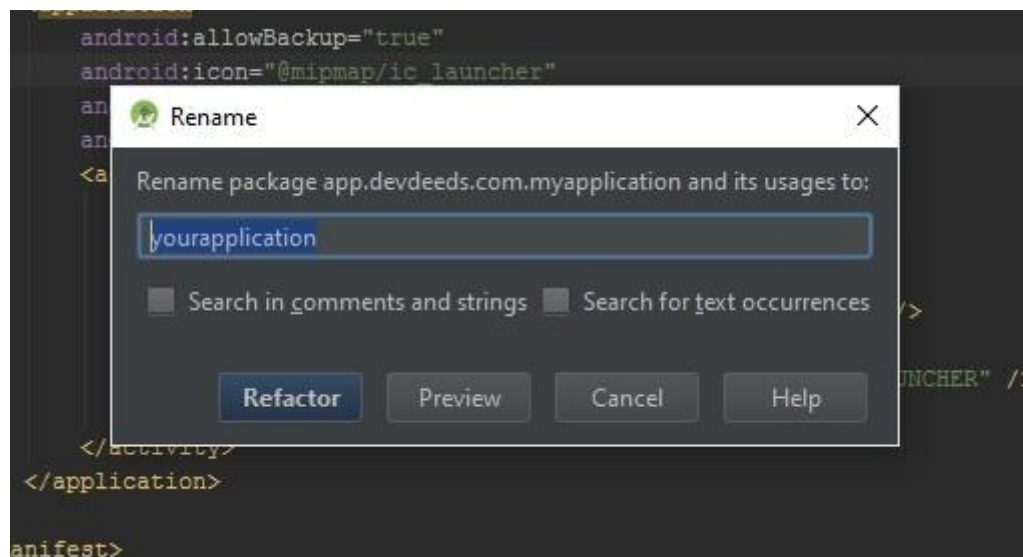
Now you can refactor desired package as you do in the code. For that right click on the package where you can see Refactor option when you select it again opens up a sub options. Please select Rename. Click on it.



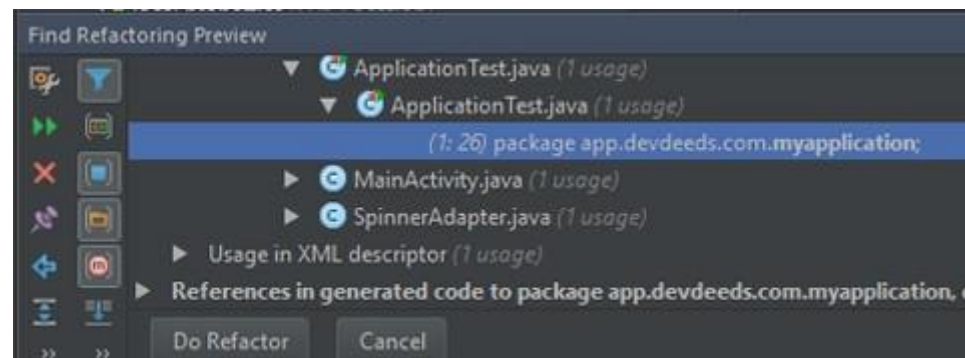
If you are asked for confirmation for renaming then please click Rename Package button.



This dialog box is for editing the existing package name. Here my existing package name is *myapplication*.



I am changing package name to new name called yourapplication.



This is asking for final confirmation to proceed. Because Android Studio has found the usage of old package name used in some places in the project. So the IDE will preview the refactoring code lines. If you press Do Refactor then it will apply the changes to these places. **\*\*How to setup Firebase Phone Auth For Android Apps \*\***

# Get SHA1 from Android Studio

 This way of finding the SHA-1 is pretty old now, as a developer you always want things like these to be on on fingertips. Now I will show you a better and faster way to get SHA-1 fingerprint.

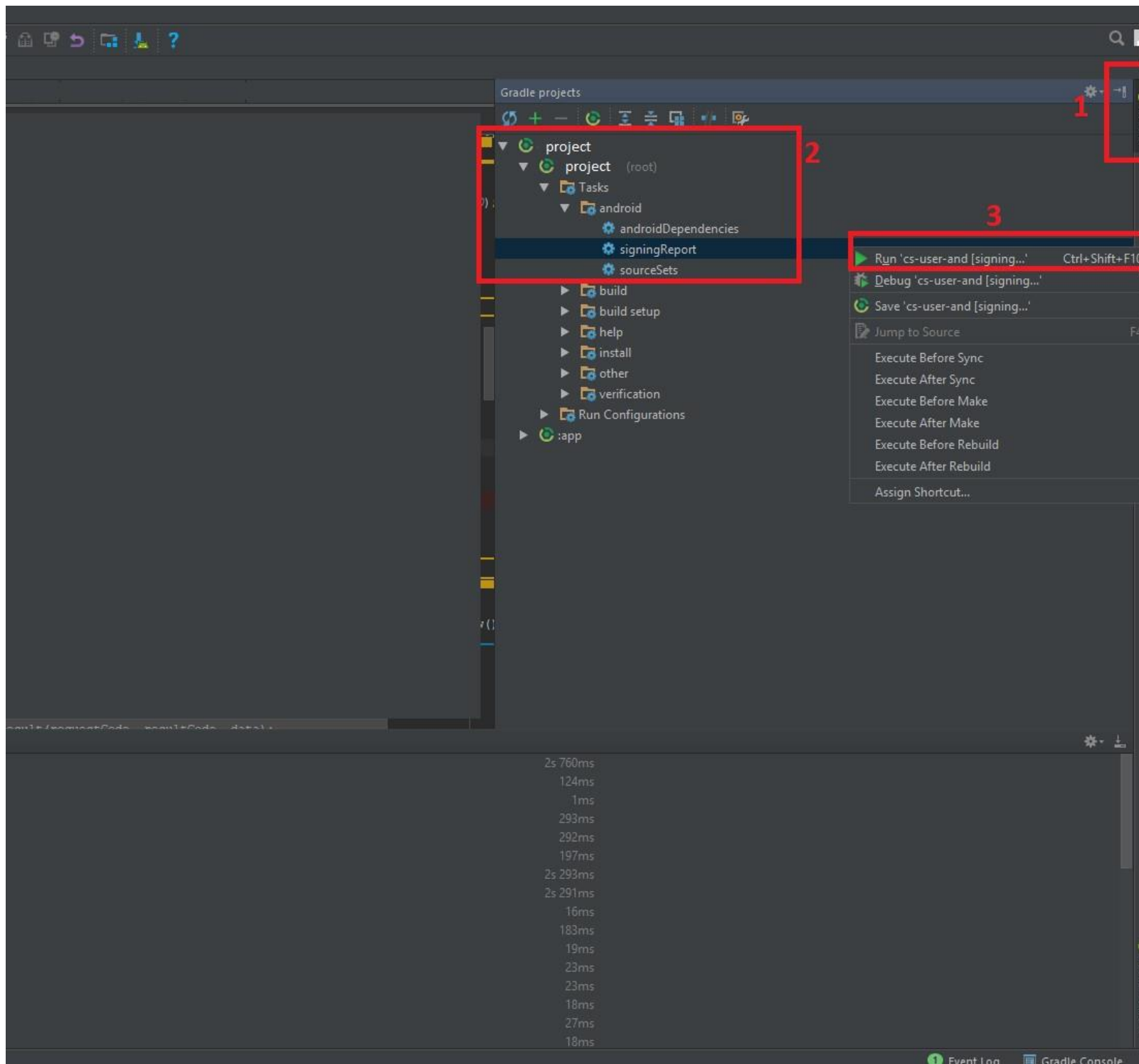
Step 1. Open your project in Android Studio.

Step 2. Click on Gradle Menu on the right side and expand it.

Step 3. Click on android and then signing report.

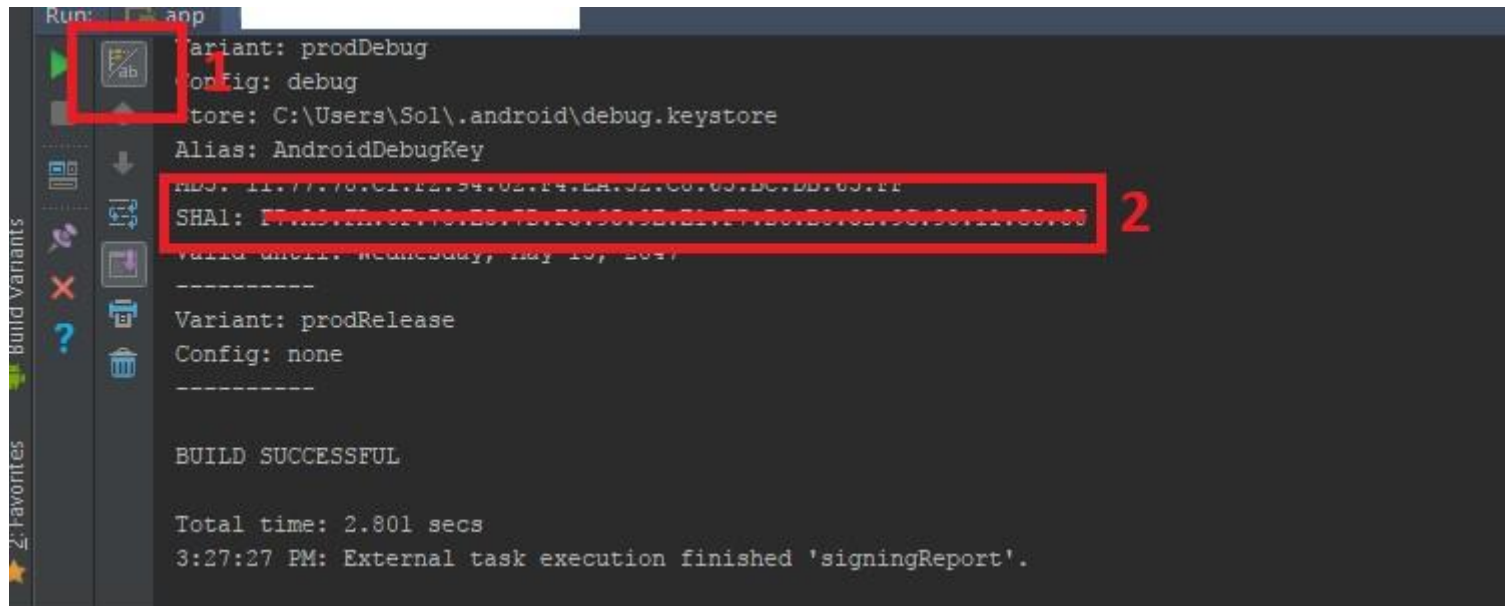
*(Follow the numbers on image)*

Magic happens here:



After clicking on the Run command, go to the Run Tab at the bottom.

Click on the first icon on the left as shown.



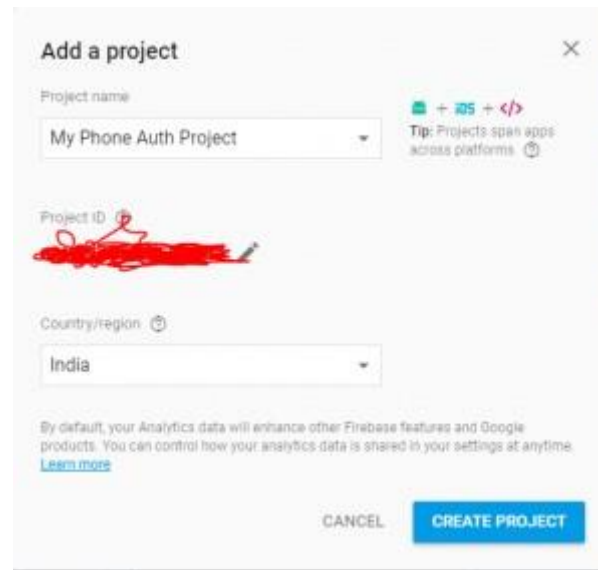
Now We have SHA1 Key which will we use in Firebase Project Setup

## Setup Firebase project

---

As we'll use Firebase Authentication, we first need to setup a project in the firebase console – Use your gmail id to login to **Firebase Console**

In the console, you will see an “Add Project” button, let’s click on this and a new popup will open up. I have given the project name as “My Phone Auth Project”. It’ll automatically generate a project id. You can keep the default. In my case also, I have kept the default.



The screenshot shows the 'Add a project' dialog box in the Firebase console. The dialog has a title bar with a close button (X). Below the title, there are three main sections: 'Project name', 'Project ID', and 'Country/region'. The 'Project name' section has a text input field containing 'My Phone Auth Project'. To the right of this field is a tip: 'Tip: Projects span apps across platforms' with a help icon. The 'Project ID' section has a text input field containing a long, alphanumeric string, which is partially obscured by a red scribble. The 'Country/region' section has a dropdown menu with 'India' selected. At the bottom of the dialog, there are two buttons: 'CANCEL' and 'CREATE PROJECT'. A small note at the bottom 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](#)'.

### Firebase-phone-auth-project-creation

Change the country as per your choice and click on “CREATE PROJECT” button. It’ll take a few moments and then firebase makes the project ready for you.

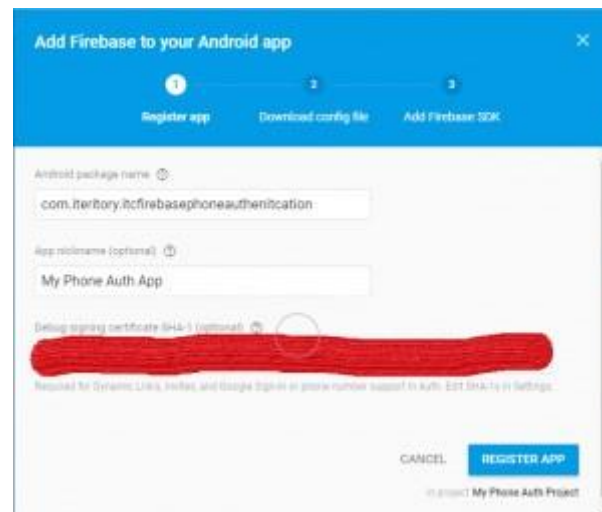
A new window will open up, there will be several options. As we are here to develop an android app, let’s select “Add Firebase to your Android app” option.



A new window will pop-up. We'll provide exactly the same package name that we used while creating the app in Android Studio. In my case, it is `com.iteritory.itcfirebasephoneauthenitcation`.

Let's Provide an app nick name as well.

Paste the SHA-1 fingerprint in the last field. I have masked my app's SHA-1 fingerprint in below image. After populating these info, my window looks like below –



Click on register app button. A new window will open up.






On the new window click on the Download google-services.json button and a .json file will be downloaded in your local system.



As described in the above screen, switch to project view if you have not already done. Copy the downloaded googl-services.json file and paste inside app folder. If you notice, it will be a sibling of build.gradle file inside app folder.

## Enable Firebase Phone Authentication

- To do this go to Firebase Console and open the project that you are using.
- Then go to Sign In Method and enable Phone Authentication.

USERS	SIGN-IN METHOD	TEMPLATES	USAGE
Sign-in providers			
Provider		Status	
 Email/Password		Disabled	
 Phone		Enabled	
 Google		Enabled	

Click on the Continue.

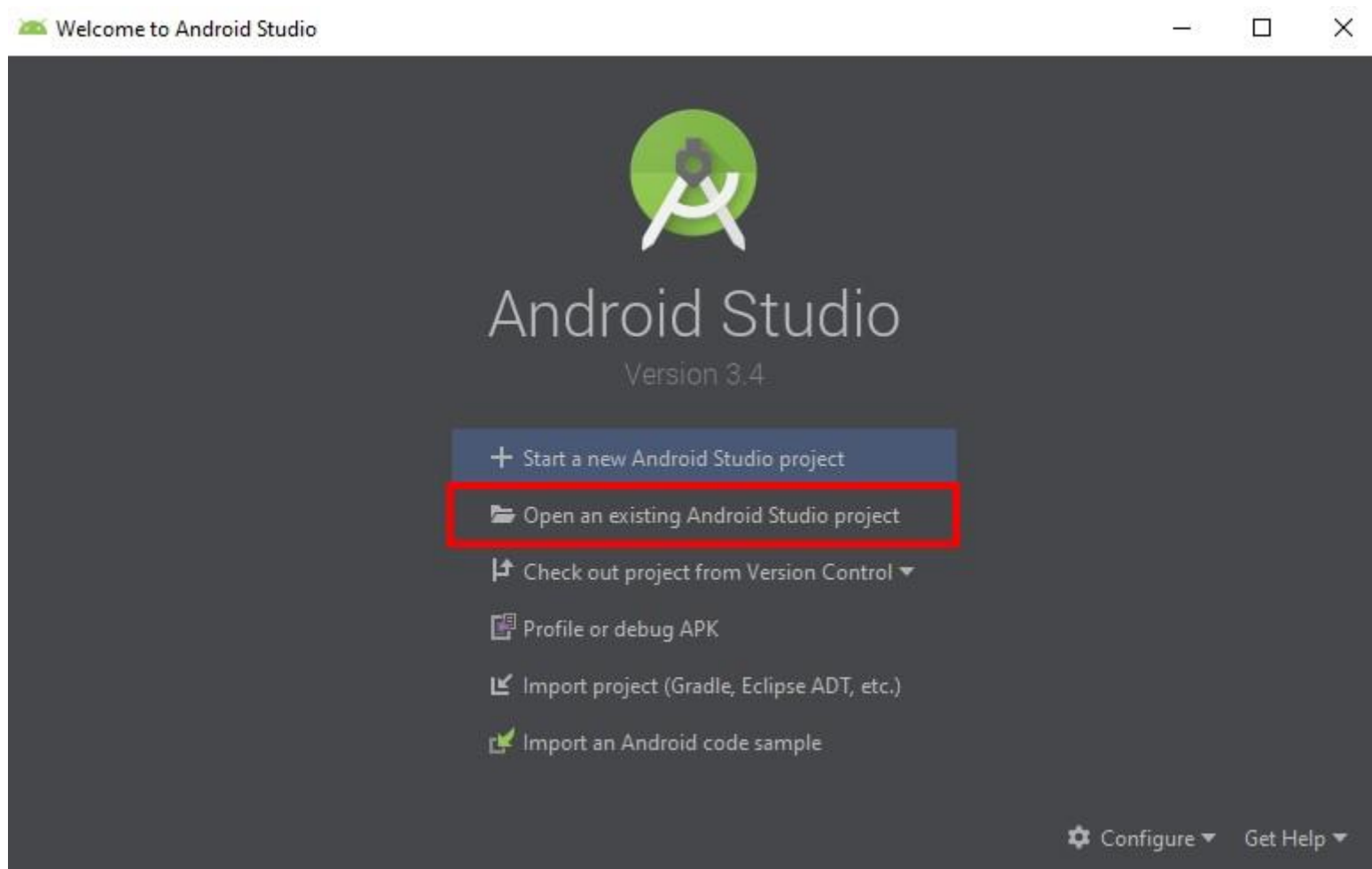
Click on Finish button.

**We Recommend to use latest version of Android Studio**

**Extract Both Zip File MeTaxiUser.zip & MeTaxiDriver.zip**

**Importing the TaxiCabOnDemandTaxi project into Android Studio**

- 1. Launch Android Studio**
- 2. Click on Open an existing Android Studio Project**



1. Select the MeTaxiUser project and open it
2. Change Package Name of your app (Steps are given above)

**\*\* Change App name, Google Map API key from Stings.xml \*\***

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE resources [
    <!ENTITY appName "MeRide">
    <!ENTITY appNameCaps "MeRide">
    <!ENTITY Provider "MeRide Driver">
    <!ENTITY provider "MeRide Driver">
    <!ENTITY Service "Ride">
    <!ENTITY service "ride">
]>
<resources>

    <string name="app_name" translatable="false">appName</string>
    <!--Google map-->
    <string name="google_map_api" translatable="false">Google Map Api</string>

```

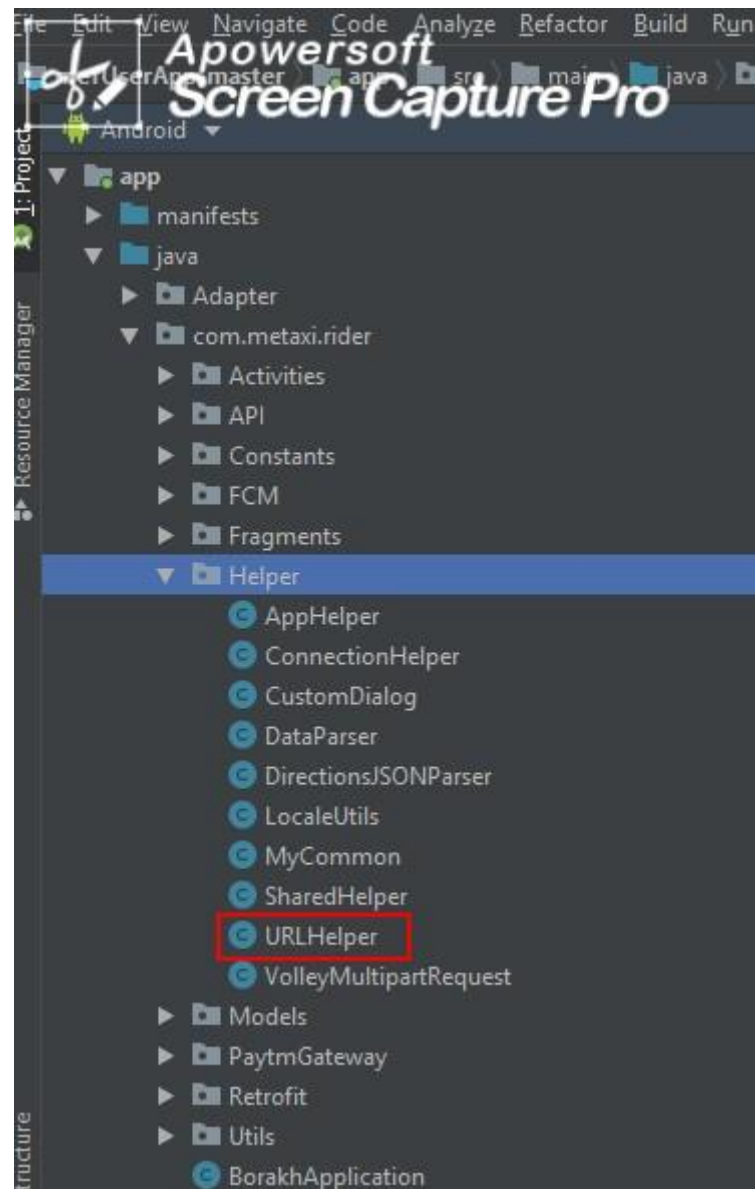
**\*\* Change App Icon \*\***



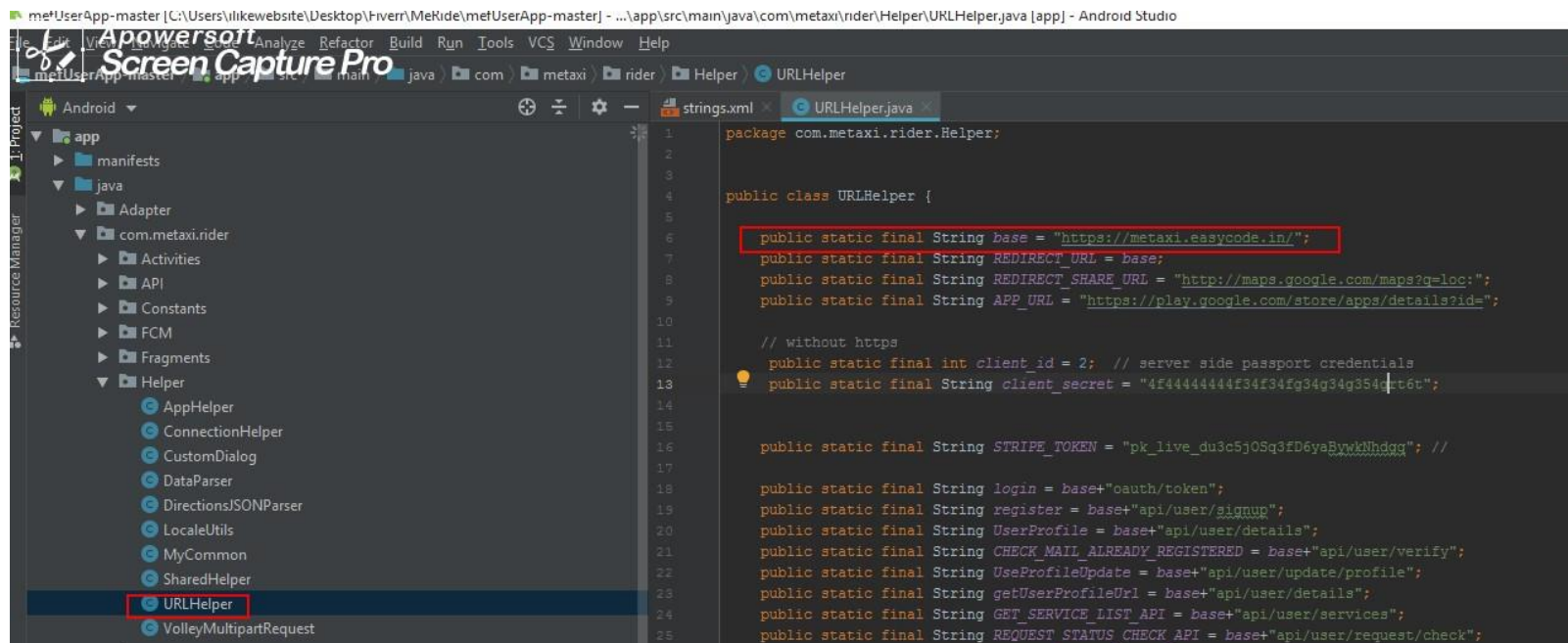
**\*\* connect server to app \*\***

**Go To**

**Packagename->Helper->UrlHelper.java**



Then change base url public static final String *base* = "<http://yoursdomain.com/>";



If you want to setup any online Payment Gateway in your app according to your country or want to setup by us contact us.

Mail ID: [geteasycode@gmail.com](mailto:geteasycode@gmail.com)

Skype: [geteasycode@gmail.com](https://www.skype.com/people/geteasycode@gmail.com)

Thank You