

## Introduction

Since the prototype has been developed as a rapid prototype, the ability of configuring the player object and the server connection is not implemented. This means that it will require some “coding” to get the game up and running. The application also require some external libraries which is not exportable and thus must be installed before the client can run. We will start stepwise and explain all the necessary steps needed to be able to run the client.

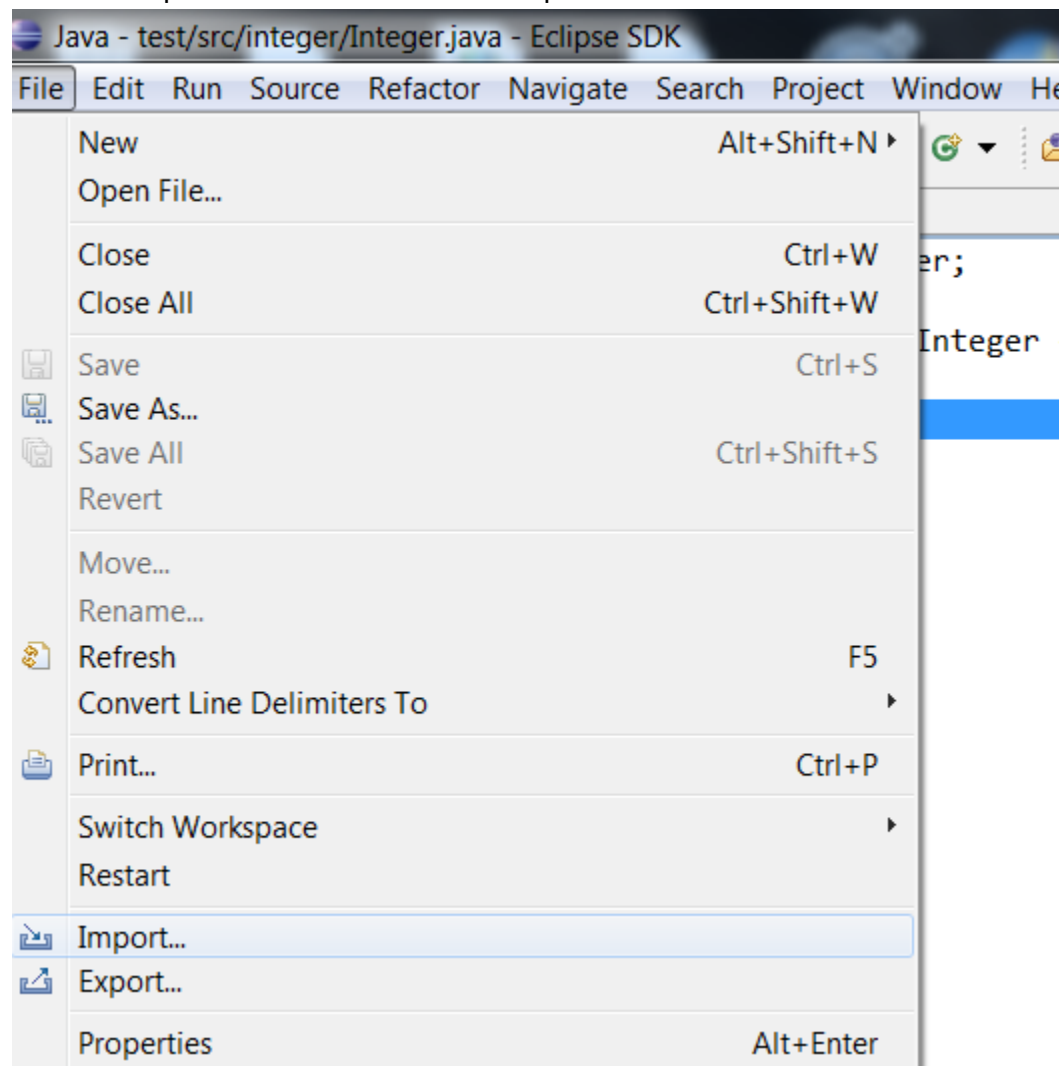
## Step one: Installing Eclipse IDE with android SDK

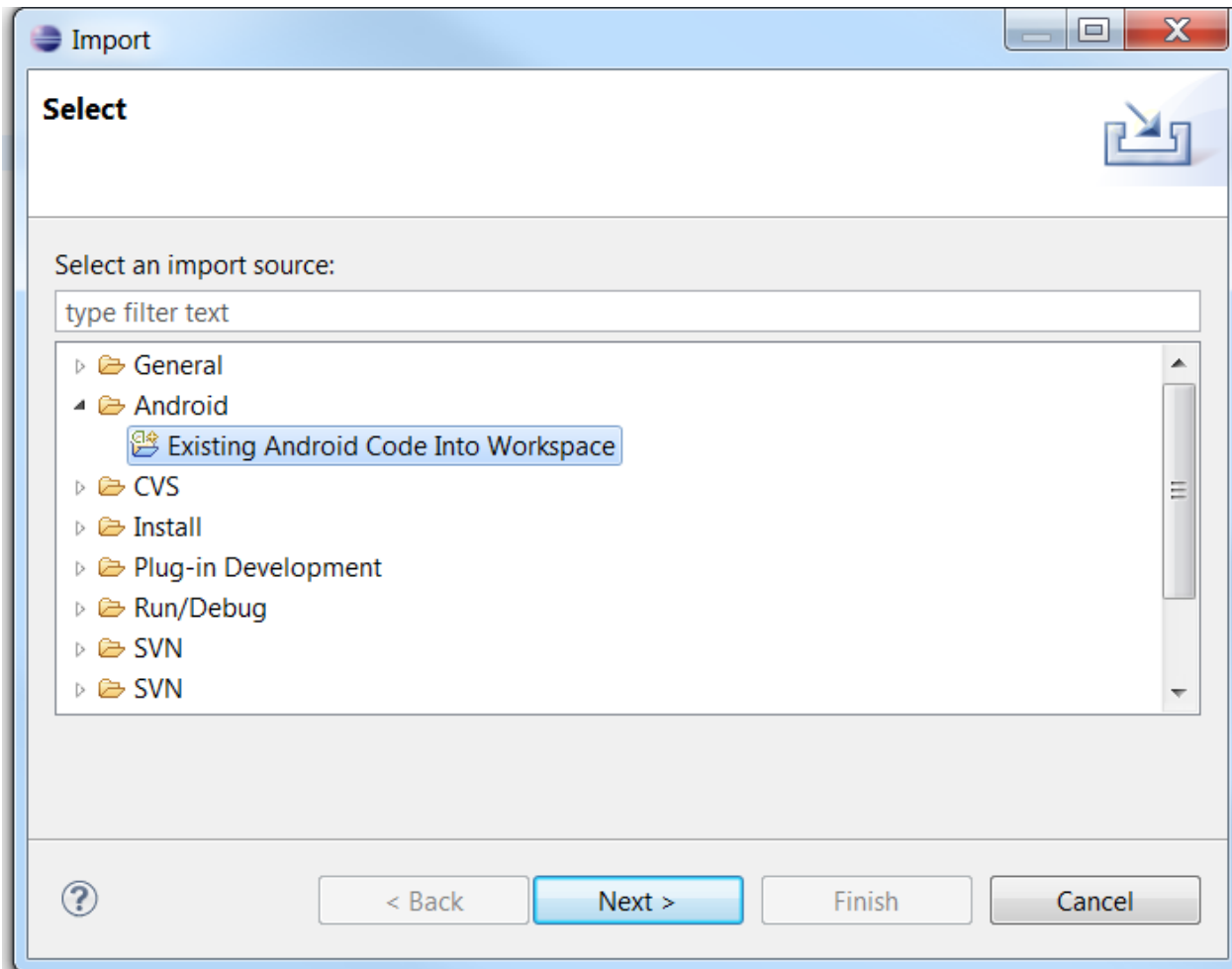
The required file can be found at: <https://developer.android.com/sdk/index.html>.

Follow the provided guideline within the website and installation file to complete the installation

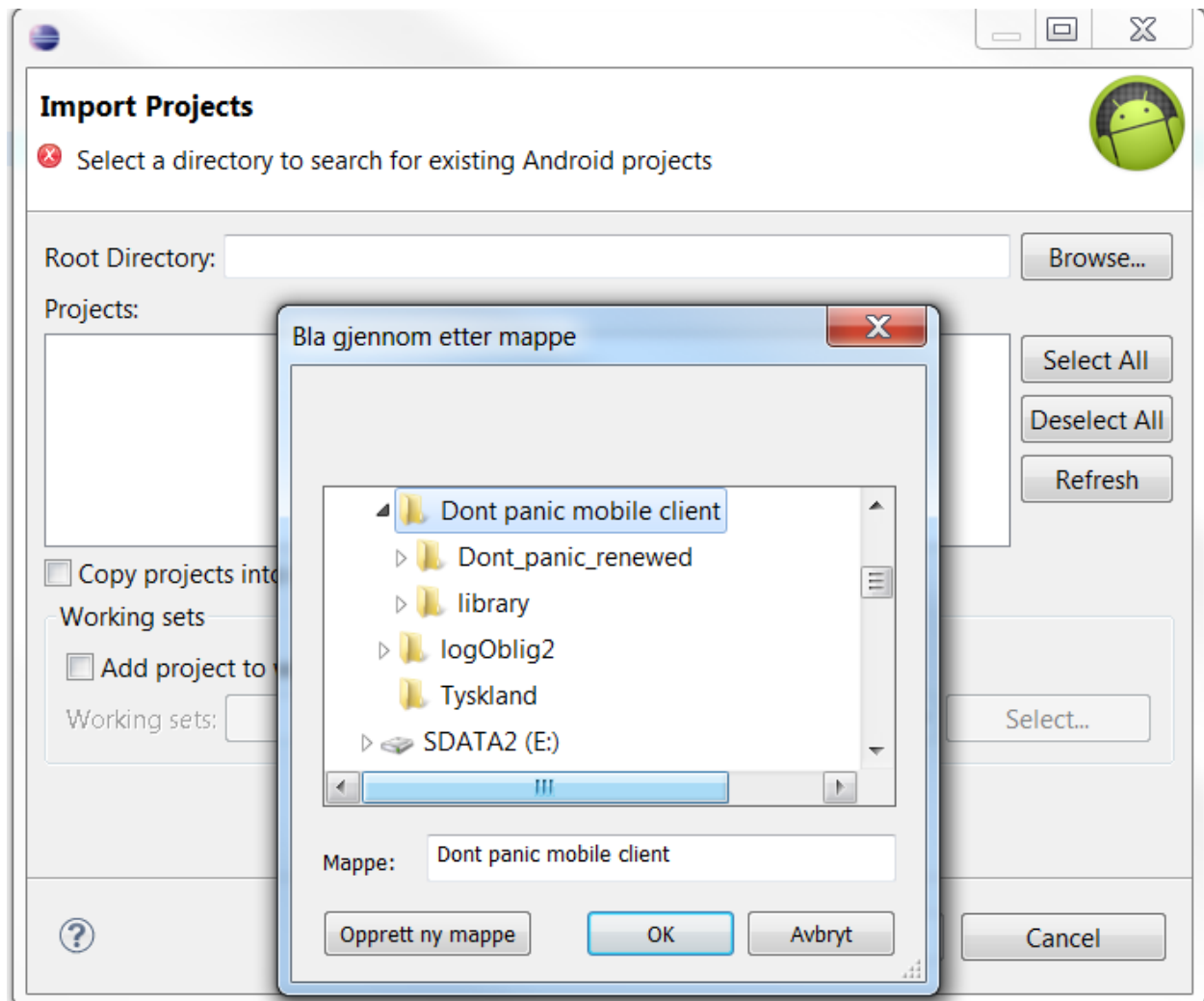
## Step two: Import the project into the eclipse IDE

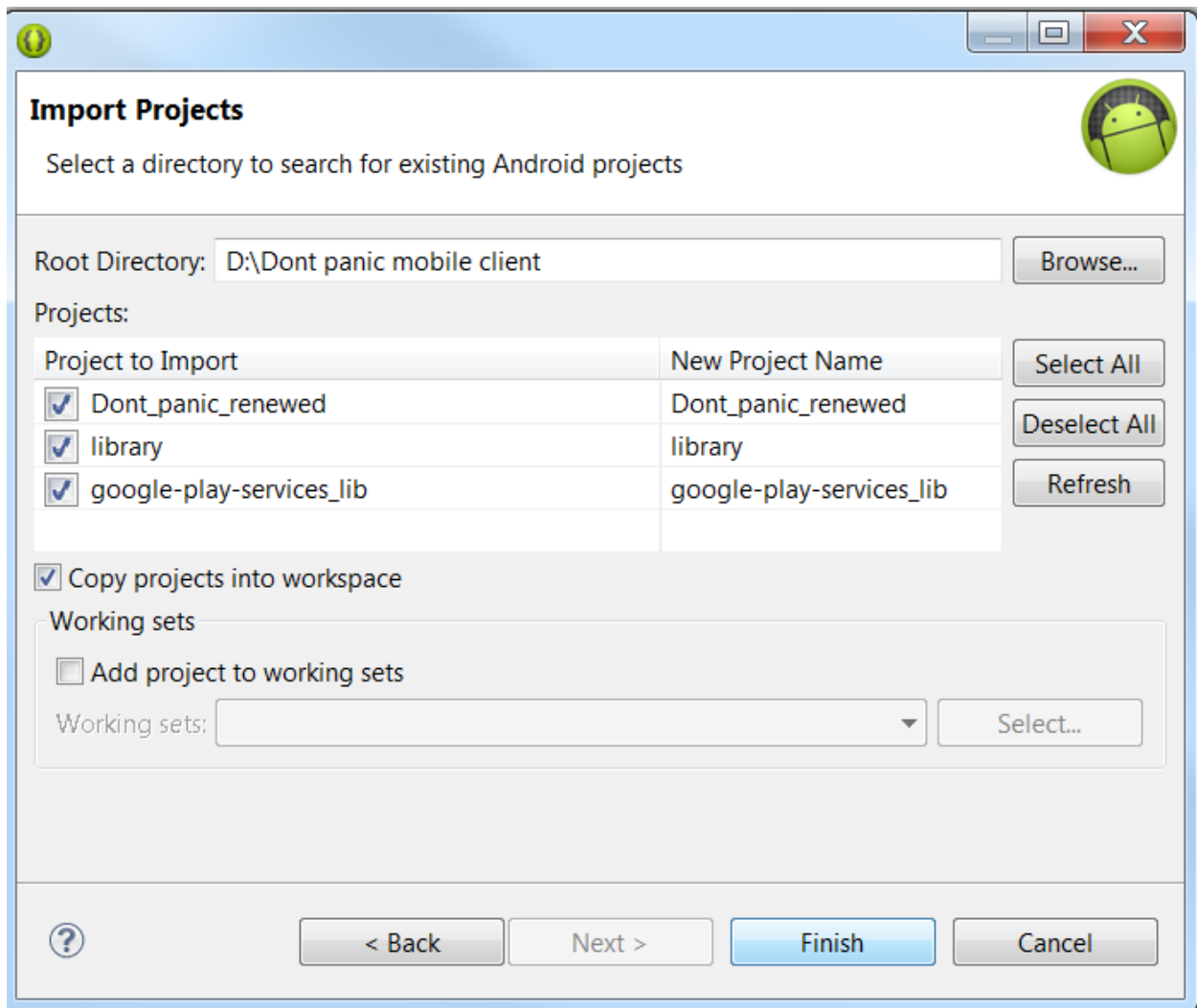
Go to left top corner and select : File->Import



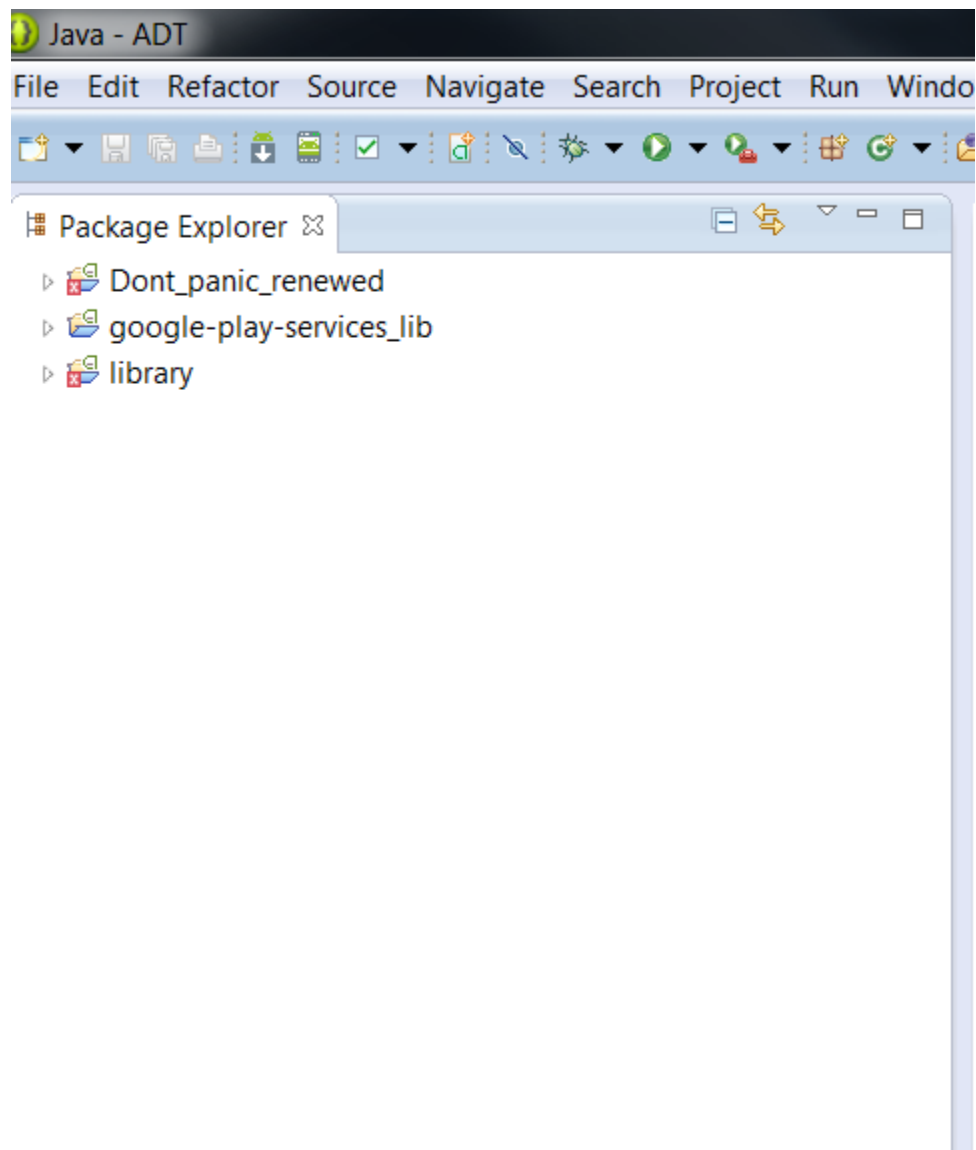


->Android project, Browse to the location of the project file and select the "Dont panic mobile client" folder.

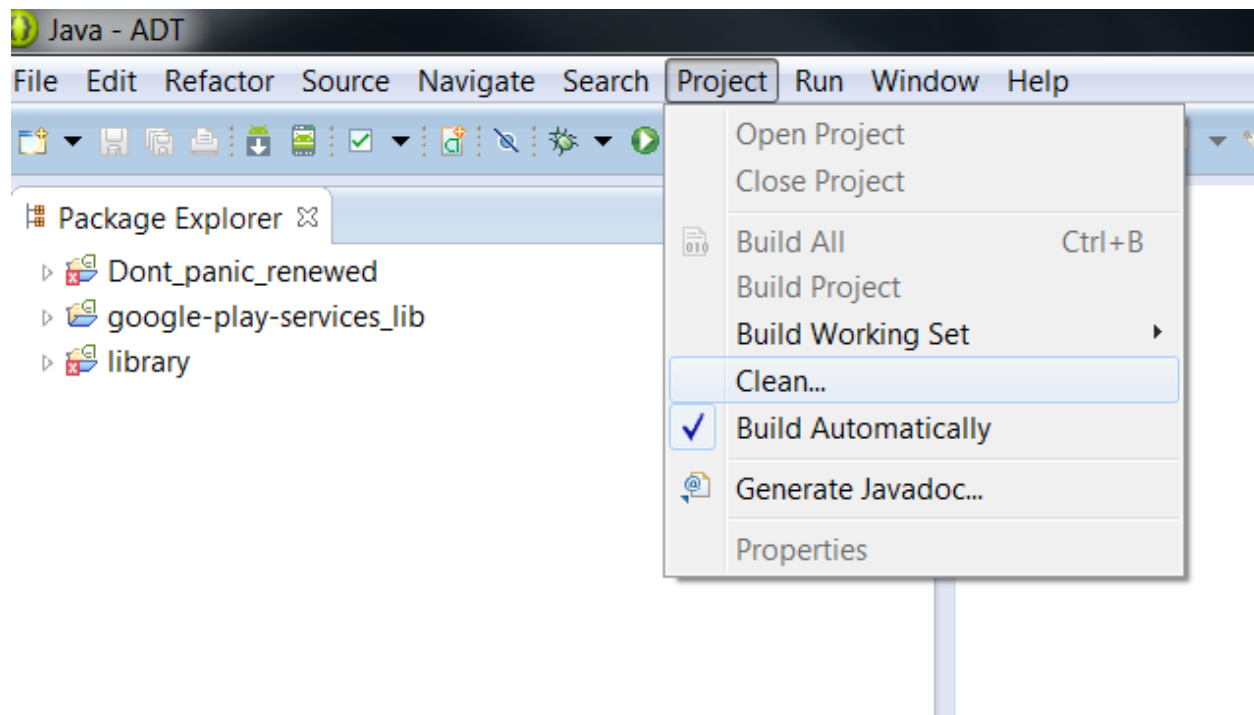




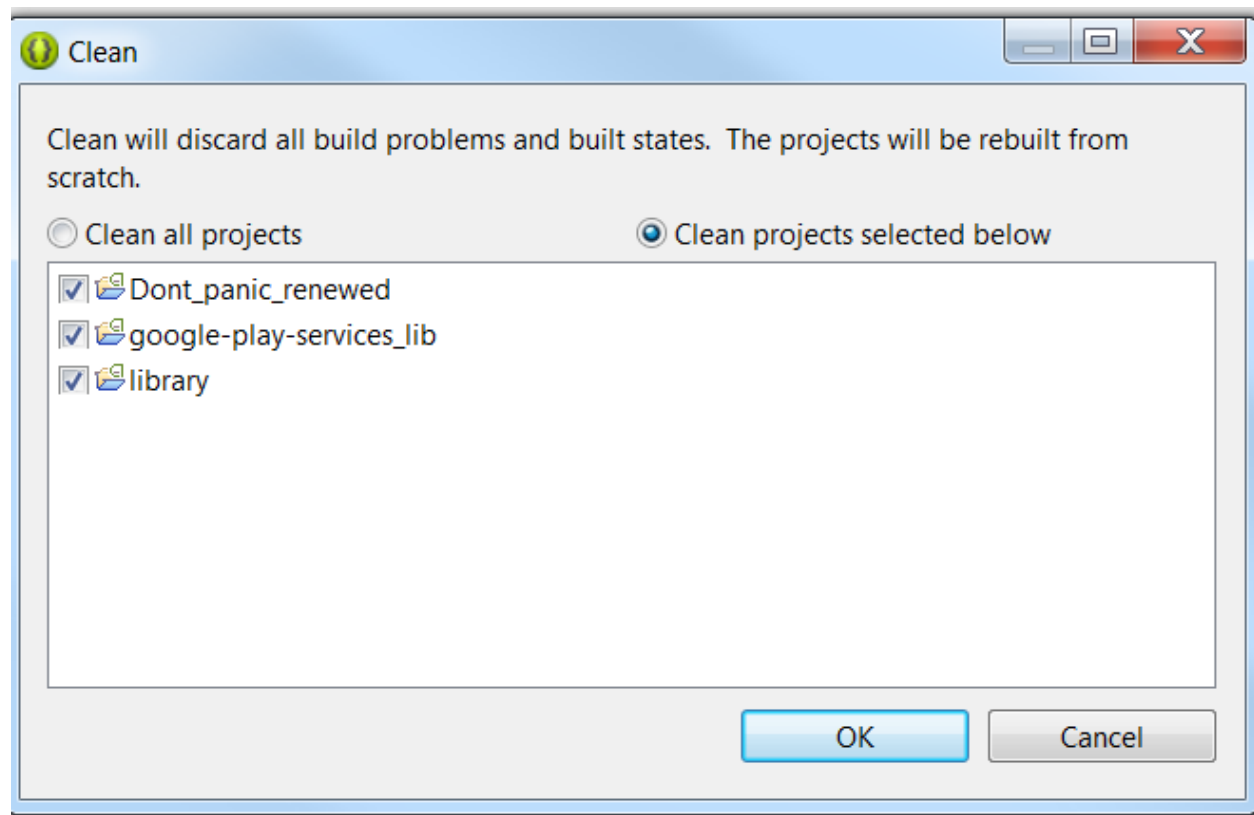
Make sure it looks like this, and press finish



The “library” and “MainActivity” project should be imported and visible in the

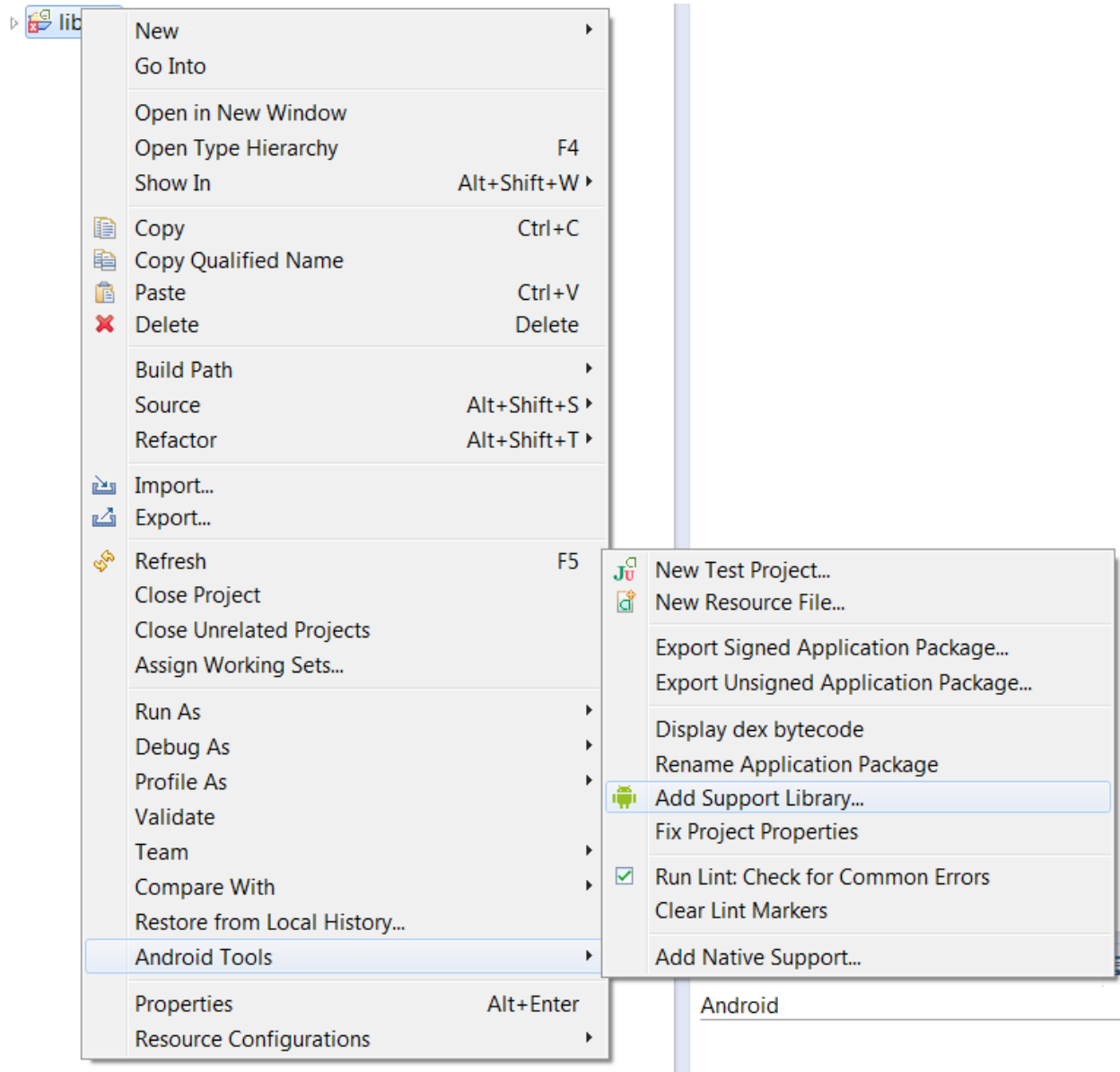


Go to project and select clean.

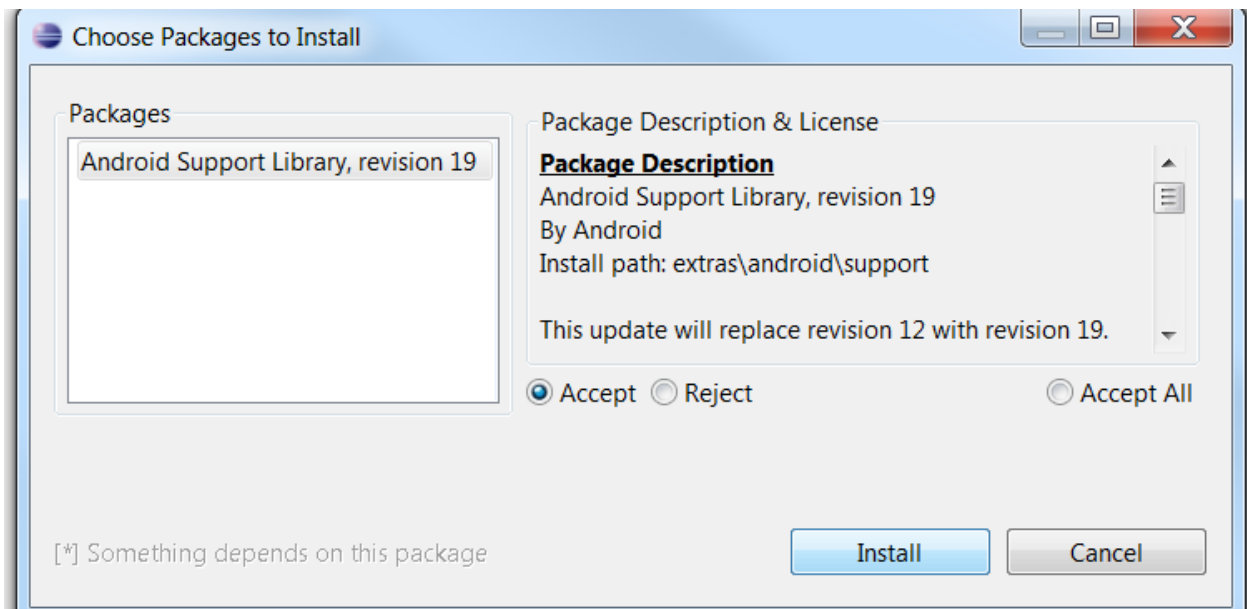


Make sure “library”, “Dont\_panic\_renewed” and “google-play-services\_lib” is selected and press OK

### Step three: Import the support library



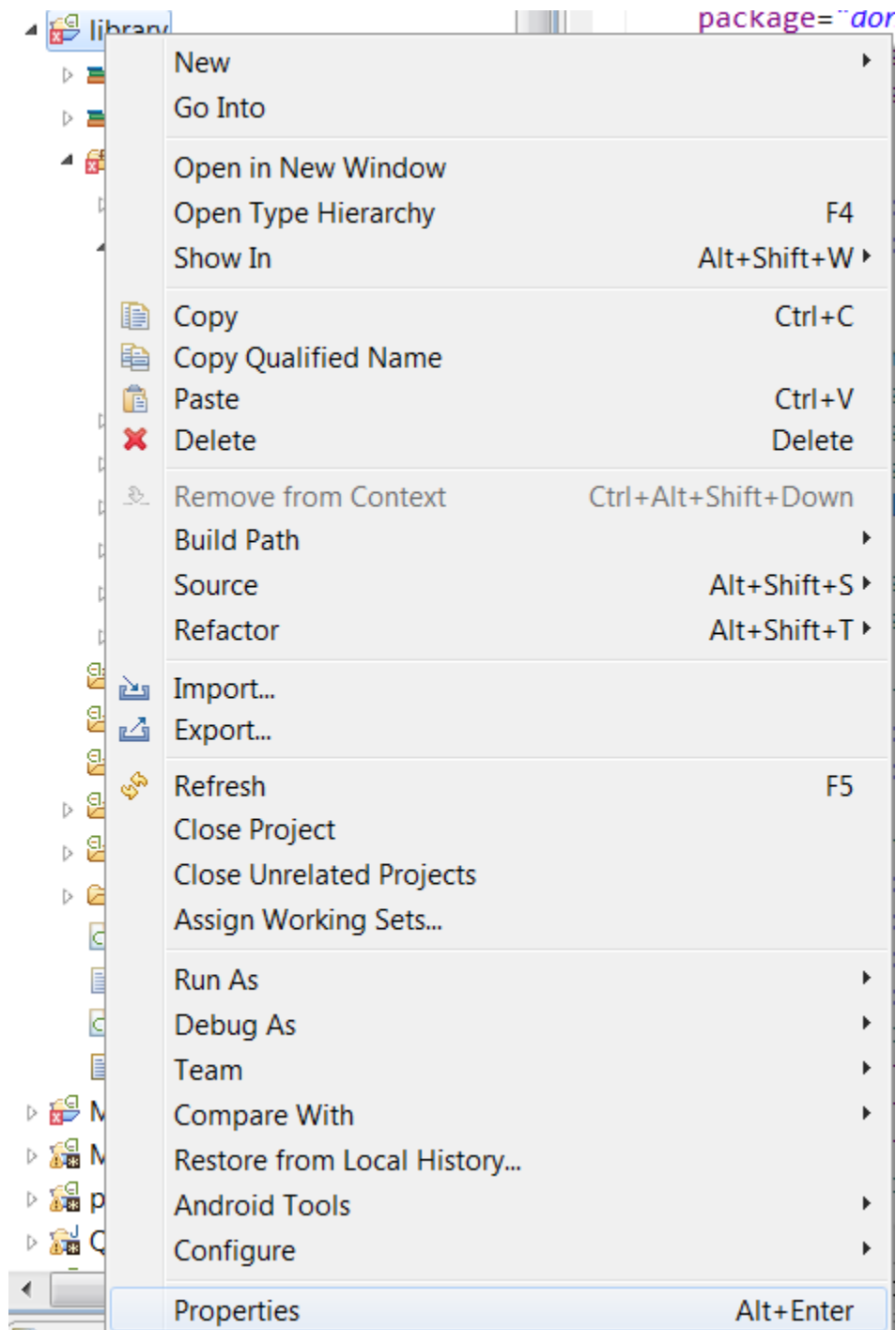
Right click the “library” project, and go to “Android Tools”->”Add Support Library...”



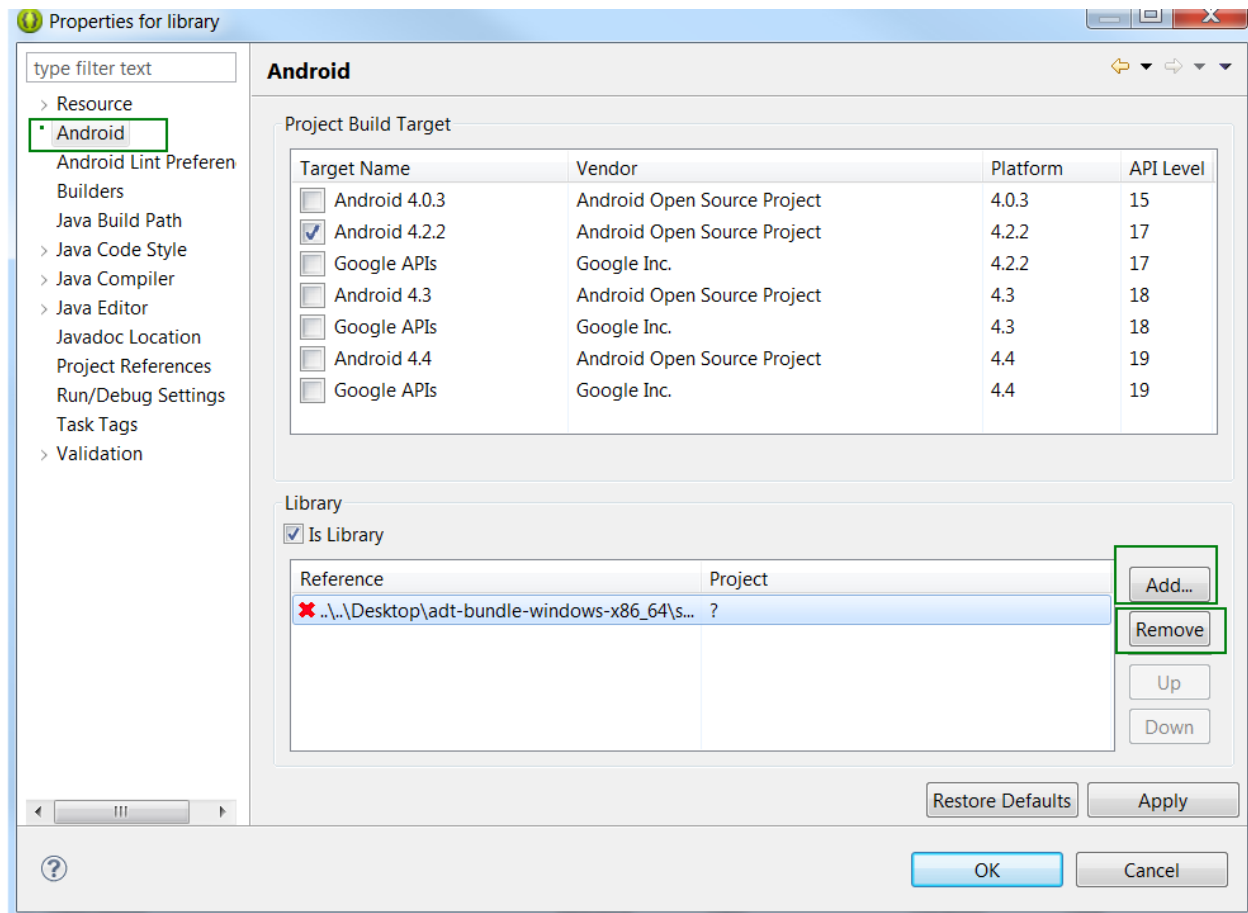
Press Install

Restart Eclipse if asked to

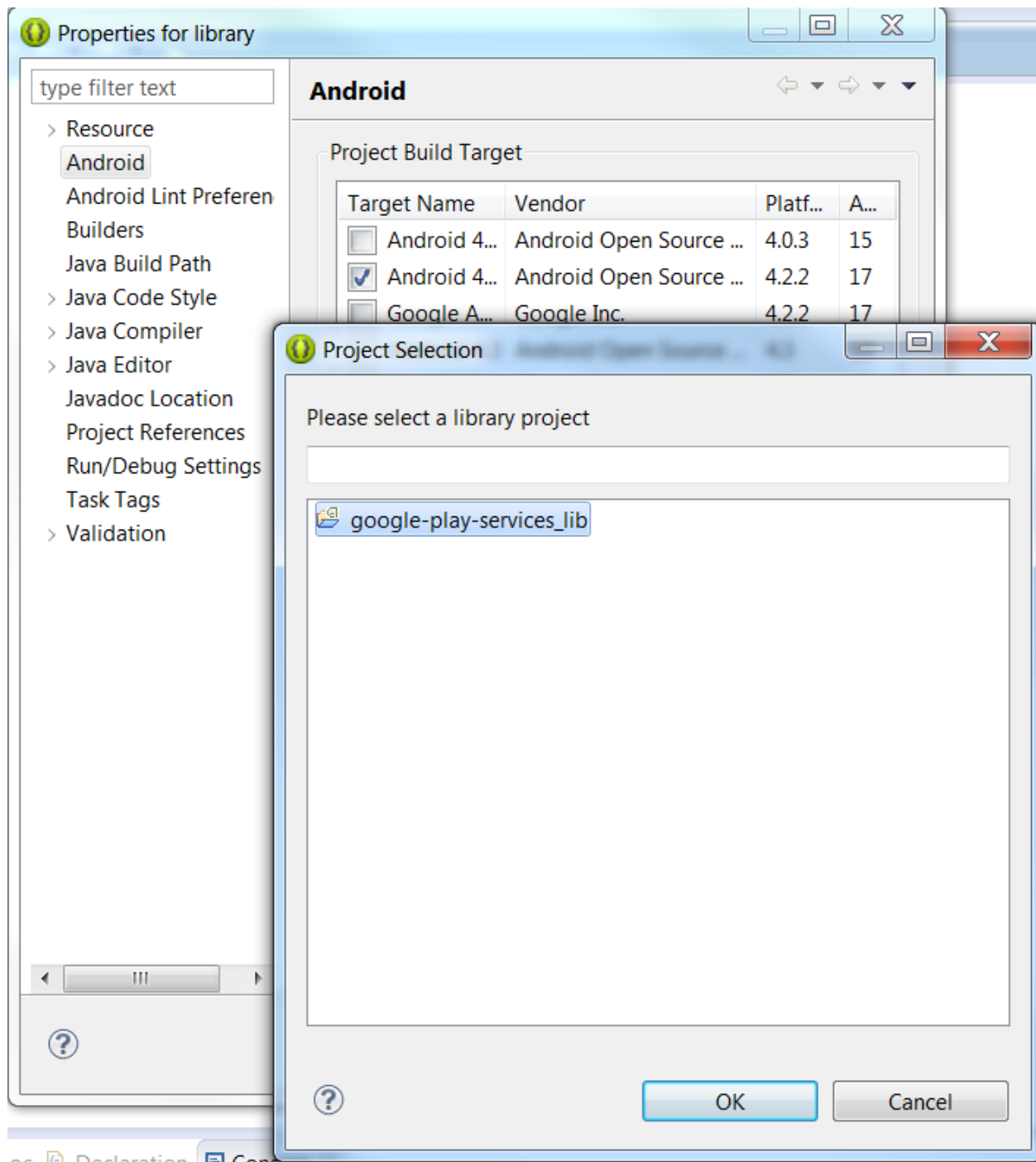




Right click "library" and select Properties

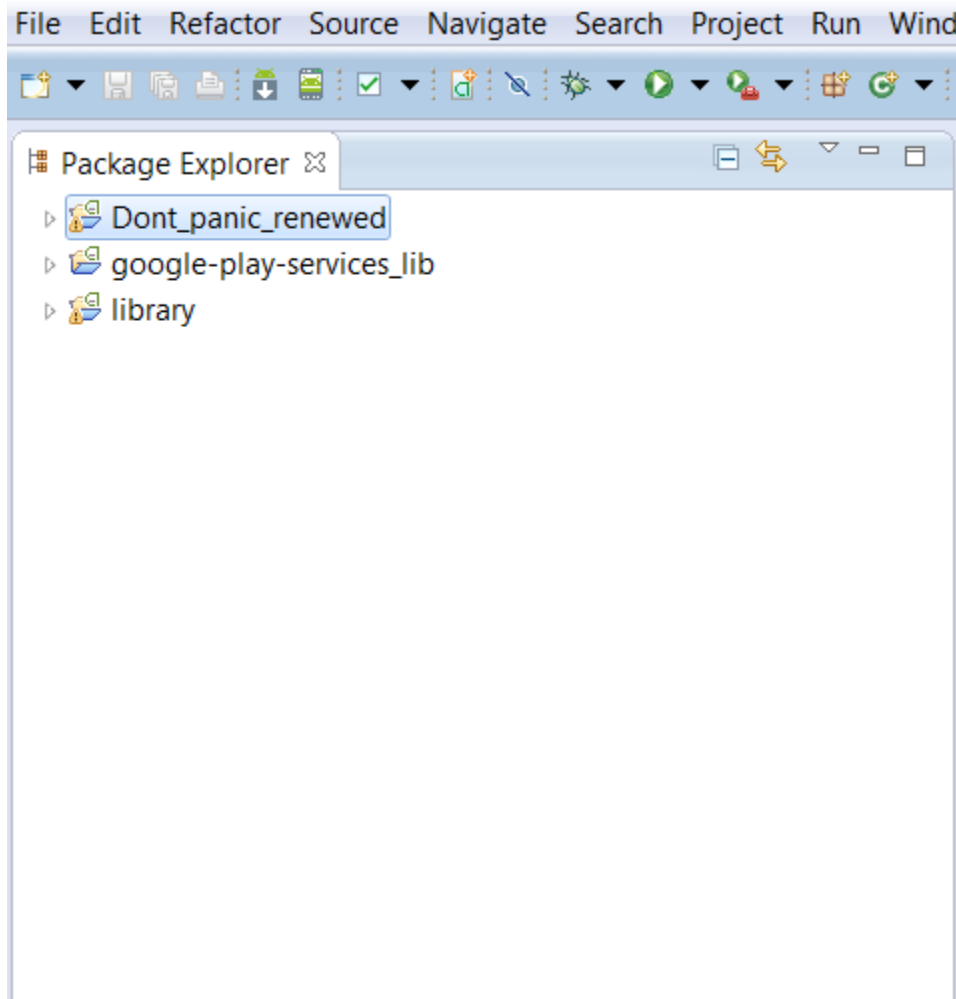


On the left menu select “Android” and then select the item with the red cross and press “Remove”, and then “Add”.



Select google-play-services\_lib and press OK

Do the same for "Dont\_panic\_renewed".



At this point all the errors should have been fixed

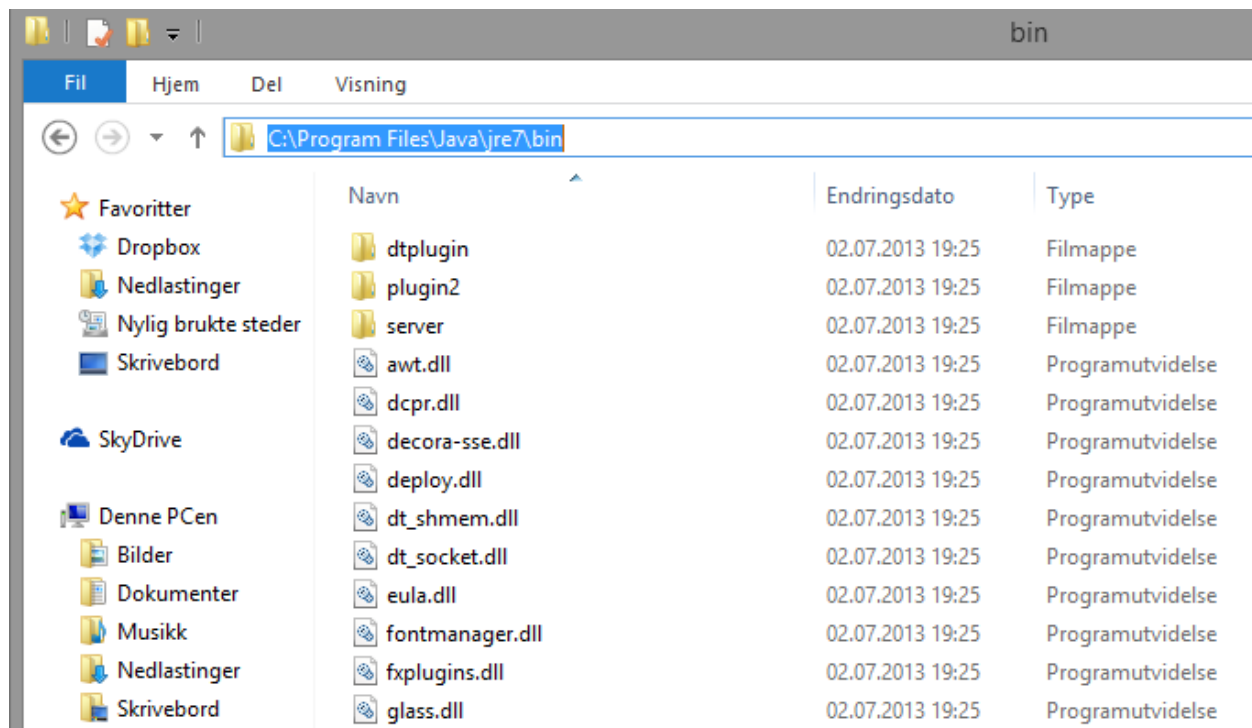
### Step four: Setting up the Google Android map API

Note, because of the restriction of Google, each unique machine is required to submit a request to google before it can use the Google android map API. Its instantly granted, however the step may be a bit tedious.

#### 1: Get the SHA1 key from the machine!

First browse to the directive for the java/bin installation.

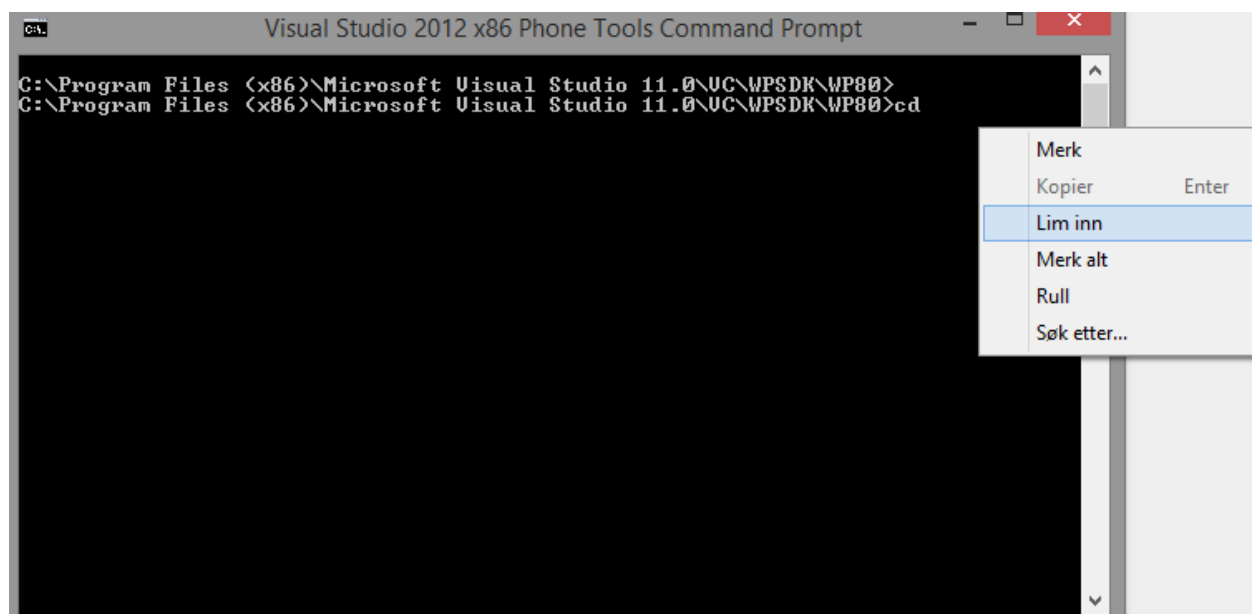
E.g C:\Program Files\Java\jre7\bin



Copy the path!

Press the windows key-> type cmd-> press enter:

Right click in the black window and write cd (then rightclick and select paste):

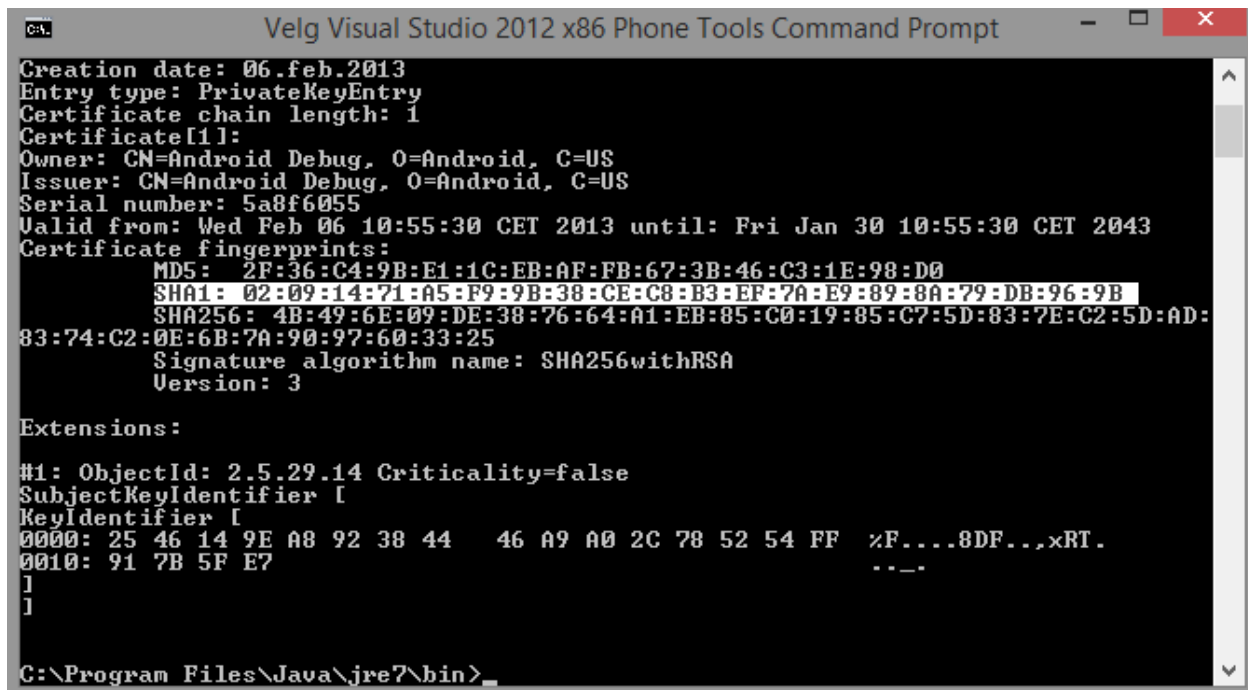


Press enter.

Then copy and paste this following command:

```
keytool -list -v -keystore "%USERPROFILE%\android\debug.keystore" -alias androiddebugkey  
-storepass android -keypass android
```

and press enter, the result should be



```
C:\Program Files\Java\jre7\bin>keytool -list -v -keystore "%USERPROFILE%\android\debug.keystore" -alias androiddebugkey -storepass android -keypass android

Creation date: 06.feb.2013
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 5a8f6055
Valid from: Wed Feb 06 10:55:30 CET 2013 until: Fri Jan 30 10:55:30 CET 2043
Certificate fingerprints:
    MD5: 2F:36:C4:9B:E1:1C:EB:AF:FB:67:3B:46:C3:1E:98:D0
    SHA1: 02:09:14:71:A5:F9:9B:38:CE:C8:B3:EF:7A:E9:89:8A:79:DB:96:9B
    SHA256: 4B:49:6E:09:DE:38:76:64:A1:EB:85:C0:19:85:C7:5D:83:7E:C2:5D:AD:
83:74:C2:0E:6B:7A:90:97:60:33:25
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 25 46 14 9E A8 92 38 44    46 A9 A0 2C 78 52 54 FF    7F....8DF...xRT.
0010: 91 7B 5F E7                    ....
]
]

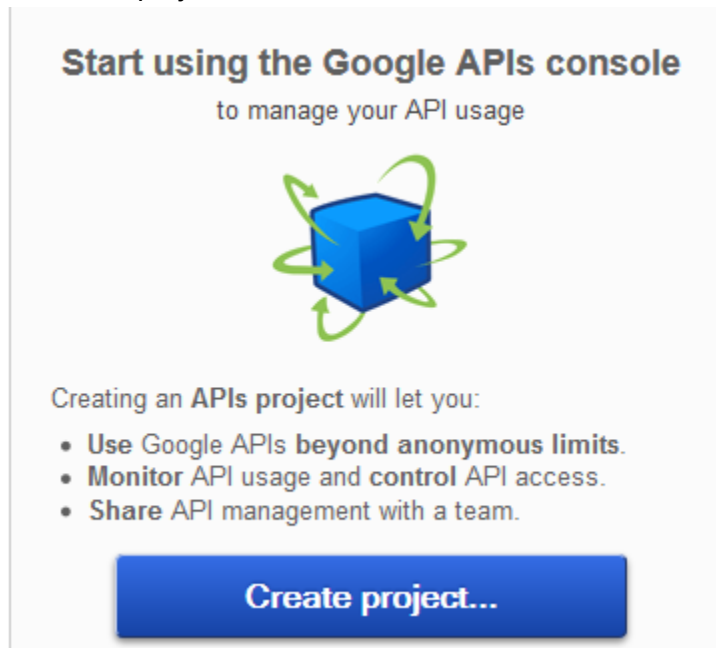
C:\Program Files\Java\jre7\bin>
```

Copy paste the SHA1 address.

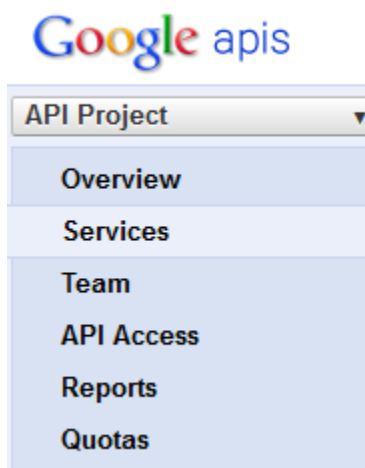
Open the browser and enter : <https://code.google.com/apis/console/?noredirect>

This require a google e-mail account, so if you don't have one, please sign up at gmail.com.

You will be greeted with this message, and simply press create project and it will automatically create an project.



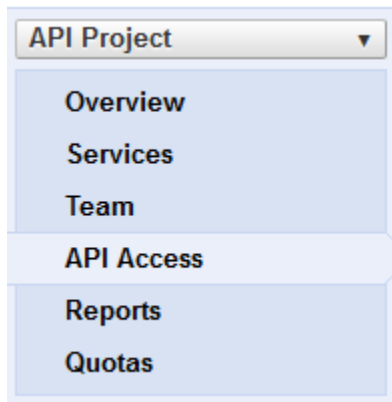
Once the project has been created, select "services" from the left side menu.



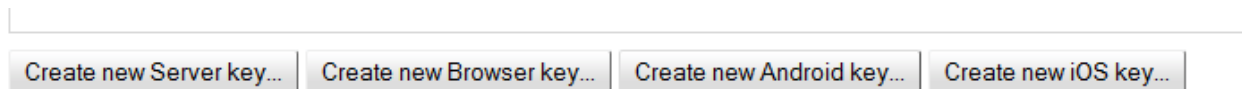
Find “Google Maps Android API v2 and press the button to the right so that its marked as ON

 Google Contacts CardDAV API 	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 <u>Google Maps Android API v2</u> 	<input checked="" type="checkbox"/> ON	
 Google Maps API v3 	<input type="checkbox"/> OFF	Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>

Then go to the menu and select “API Access”



Press the button “Create new Android Key”



### Notification Endpoints



Paste in the SHA1 which we copied from the black command window and add “;dont.panic.renewed” and hit create.

Configure Android Key for API Project

**This key can be deployed in your Android applications.**

API requests are sent directly to Google from your clients' Android devices. Google verifies that each request originates from an Android application that matches one of the certificate SHA1 fingerprints and package names listed below. You can discover the SHA1 fingerprint of your developer certificate using the following command:

```
keytool -list -v -keystore mystore.keystore
```

[Learn more](#)

**Accept requests from an Android application with one of the certificate fingerprints and package names listed below:**

02:09:14:71:A5:F9:9B:38:CE:C8:B3:EF:7A:E9:89:8A:79:DB:96:9B;dont.panic.renewed

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:  
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

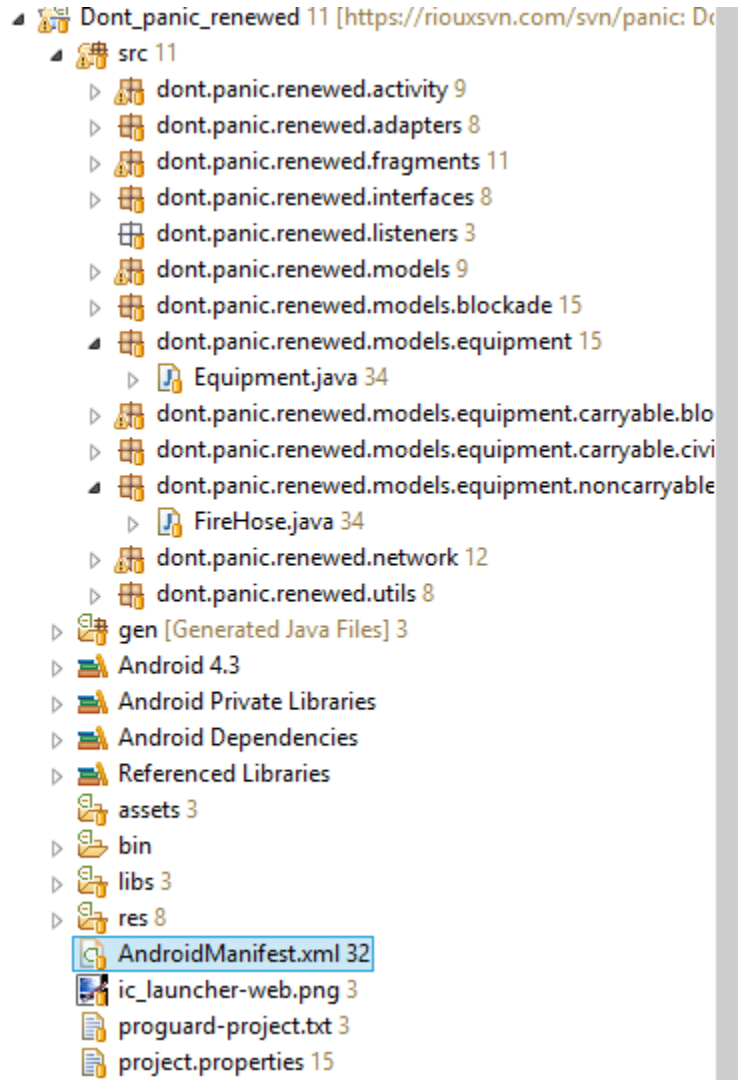
Create

Cancel

Copy the API key

Key for Android apps (with certificates)	
API key:	AIzaSyDarWr15BZMiGCXWf7OYn4N1NAQI05EqsY
Android apps:	02:09:14:71:A5:F9:9B:38:CE:C8:B3:EF:7A:E9:89:8A:79:DB:96:9B;dont.panic.renewed
Activated on:	Oct 1, 2013 7:08 AM
Activated by:	lap.q.to@gmail.com – you

Enter the Android Manifest file for the Dont\_panic\_renewed project



Paste in the API key in the marked section.

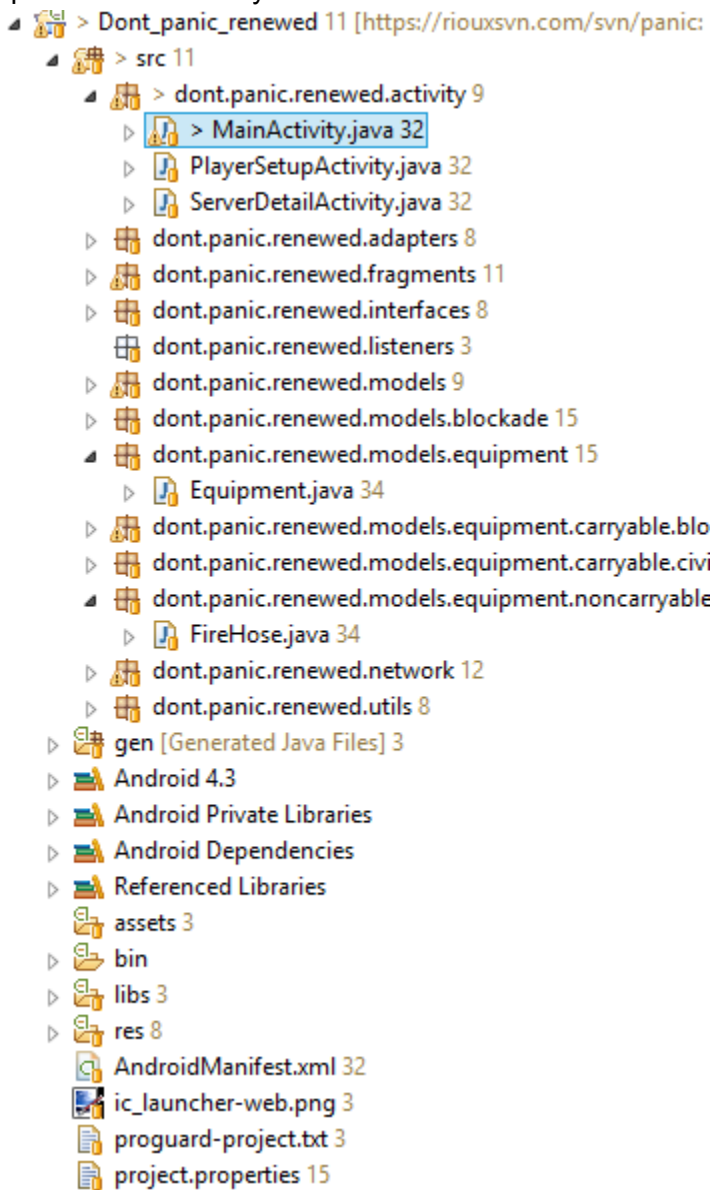
```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyDarWrL5BZMiGCXvf70Yn4NLNAQI05EqsY" />

    <meta-data android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
</application>
```

Exit Eclipse and restart it. Now all the error markers should be gone.

## Step five: Setting player nickname and role

Open the MainActivity file



The screenshot shows a project explorer with the following structure:

- ▲ > Dont\_panic\_renewed 11 [<https://riouxsvn.com/svn/panic>]
  - ▲ > src 11
    - ▲ > dont.panic.renewed.activity 9
      - ▶ > MainActivity.java 32
      - ▶ PlayerSetupActivity.java 32
      - ▶ ServerDetailActivity.java 32
    - ▶ dont.panic.renewed.adapters 8
    - ▶ dont.panic.renewed.fragments 11
    - ▶ dont.panic.renewed.interfaces 8
    - ▶ dont.panic.renewed.listeners 3
    - ▶ dont.panic.renewed.models 9
    - ▶ dont.panic.renewed.models.blockade 15
    - ▲ dont.panic.renewed.models.equipment 15
      - ▶ Equipment.java 34
    - ▶ dont.panic.renewed.models.equipment.carryable.blo
    - ▶ dont.panic.renewed.models.equipment.carryable.civi
    - ▲ dont.panic.renewed.models.equipment.noncarryable
      - ▶ FireHose.java 34
    - ▶ dont.panic.renewed.network 12
    - ▶ dont.panic.renewed.utils 8
  - ▶ gen [Generated Java Files] 3
  - ▶ Android 4.3
  - ▶ Android Private Libraries
  - ▶ Android Dependencies
  - ▶ Referenced Libraries
  - ▶ assets 3
  - ▶ bin
  - ▶ libs 3
  - ▶ res 8
  - ▶ AndroidManifest.xml 32
  - ▶ ic\_launcher-web.png 3
  - ▶ proguard-project.txt 3
  - ▶ project.properties 15

In the MainActivity, under Oncreate the players nickname and role can be selected.

```

/**
 * Default onCreate function in android application.
 * This sets up the fragment adapter which is needed to use fragments and also setup the player o
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ActionBar actionBar = getSupportActionBar();
    fragmentAdapter = new MapAndActionFragmentAdapter(getSupportFragmentManager());
    actionBar.setNavigationMode(2);

    Player.getInstance().setNickname("Alex");
    Player.getInstance().setRole(RolesNamesAndBenefis.FIREFIGHTER);
    Player.getInstance().setUpTestEquipmentData();

    viewPager = (ViewPager) findViewById(R.id.activity_main_pager);

    viewPager.setAdapter(fragmentAdapter);

    viewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            // When swiping between different app sections, select the corresponding tab.
            // We can also use ActionBar.Tab#select() to do this if we have a reference to the
            // Tab.
            actionBar.setSelectedNavigationItem(position);
        }
    });
}

```

To change the nick, simply replace whats in the “ “ which whats desired.

To set the role to fire fighter, replace the setRole line to:

```
Player.getInstance().setRole(RolesNamesAndBenefis.FIREFIGHTER);
```

To set the role to medic, replace the setRole line to:

```
Player.getInstance().setRole(RolesNamesAndBenefis.MEDIC);
```

To set the role to volunteer, replace the setRole line to:

```
Player.getInstance().setRole(RolesNamesAndBenefis.VOLUNTEER);
```

## Step six: Configuring the server IP

To configure the Ip to the server, go to the MainAcitvty file and scroll down to

```

17-  /*
18-   * this method establish the connection with the server.
19-   */
20-  public void setUpClient(){
21-      client = new GameClient("129.241.103.121", this);
22-      t = new Thread(new Runnable() {
23-
24-          @Override
25-          public void run() {
26-              client.run();
27-
28-          }
29-      });
30-      t.start();
31-  }

```

Replace the IP adress.

## Step Seven: Run.

To run the project simply select

