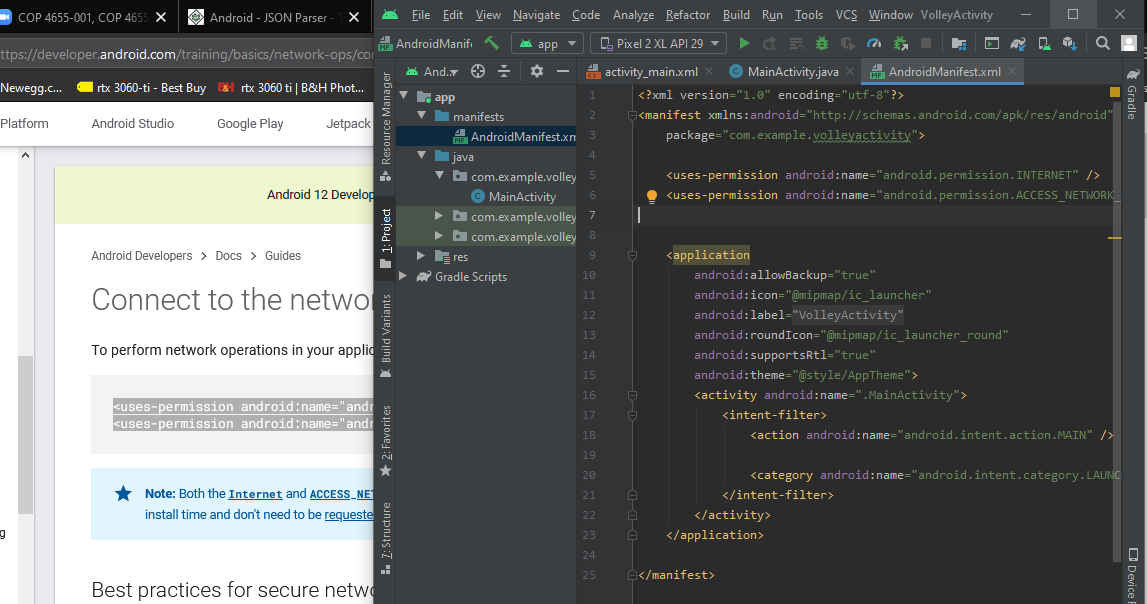
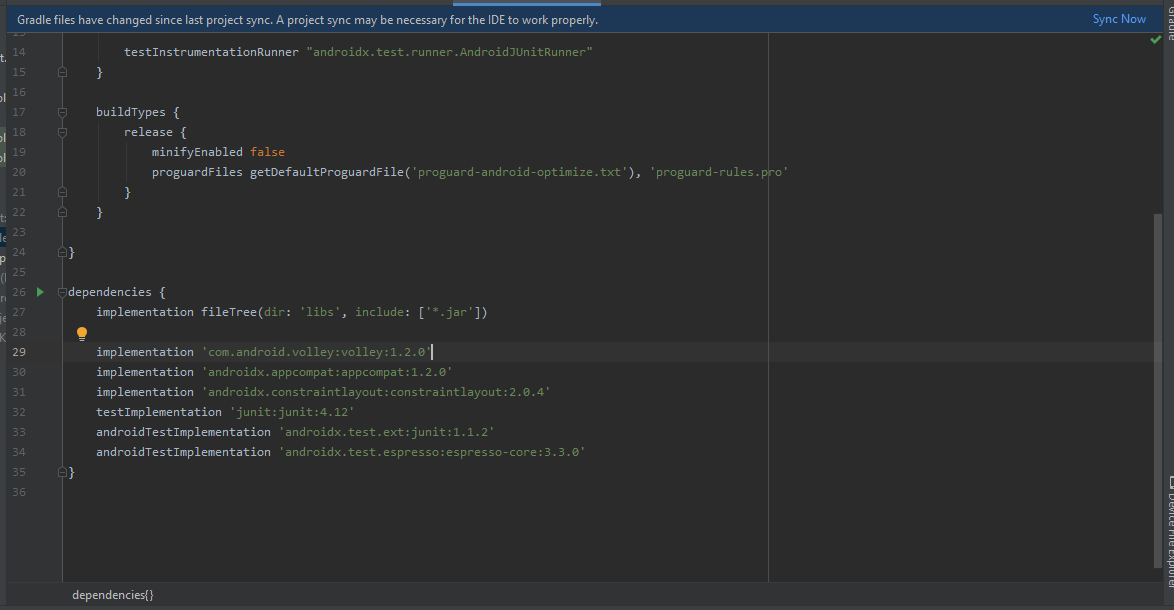
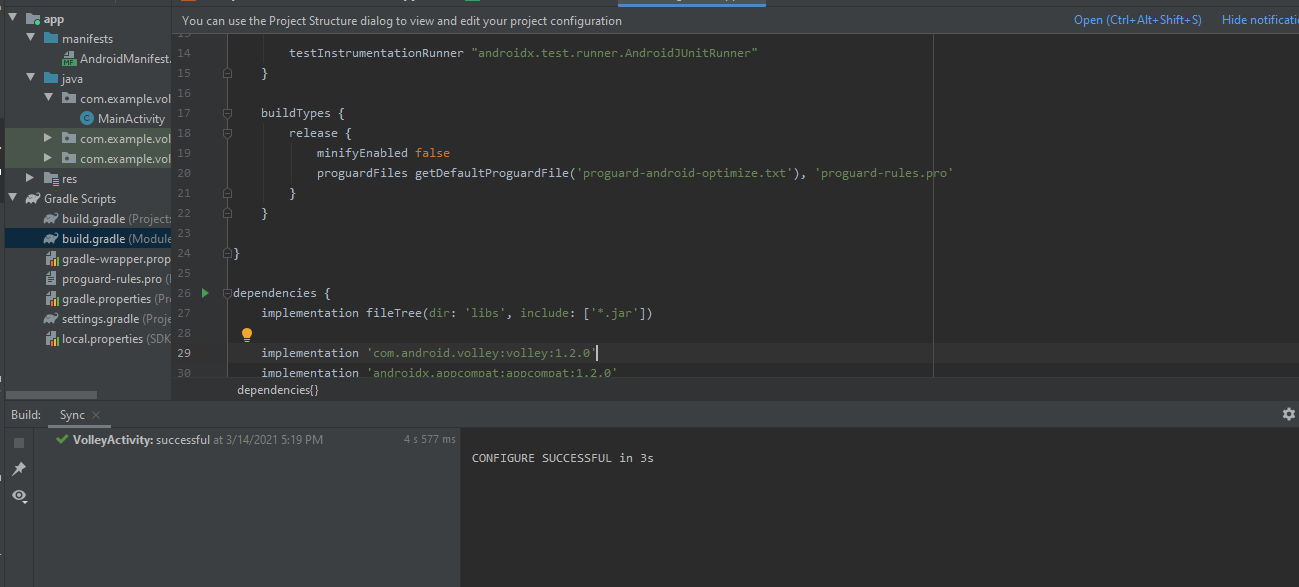
HW 7 Mobile Apps Maaz Malik Z23385841

3/14/2021 Dr.Jaramillo

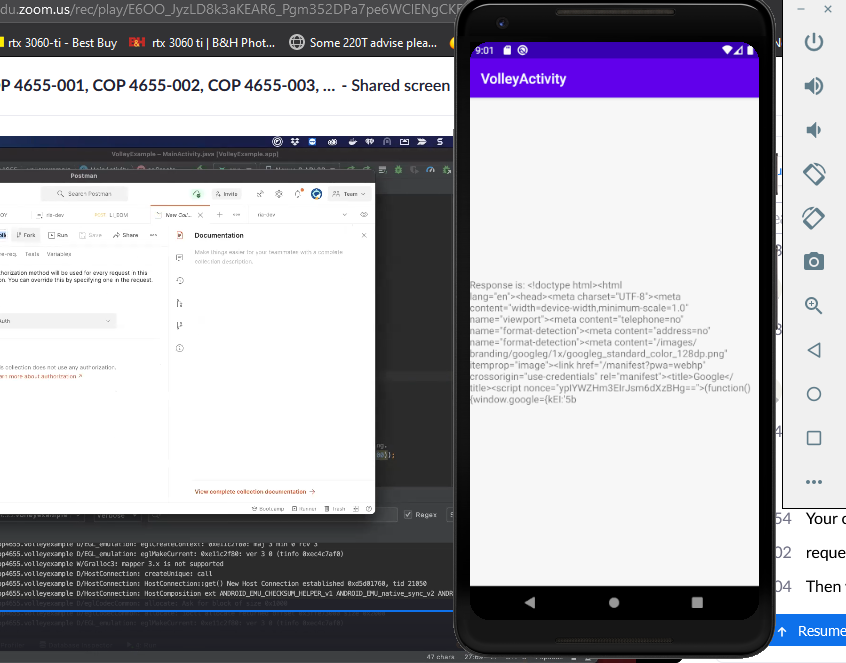


I added the Internet Android Manifest lines in order to Access the internet.

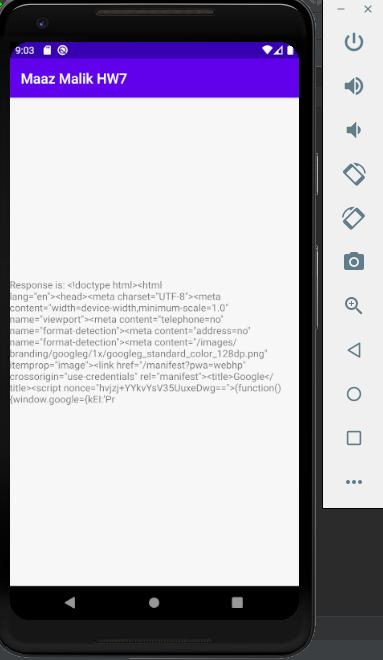


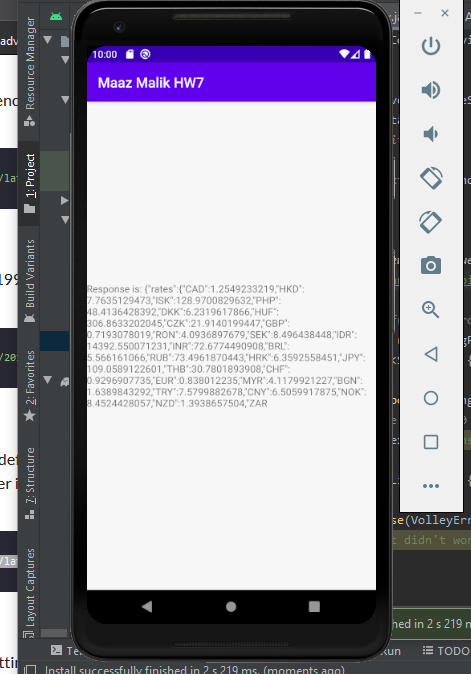


I added the Gradle Files and Pressed the Sync Now button. My Decencies look slightly different from what the Professor had but its ok because I added what I needed to.



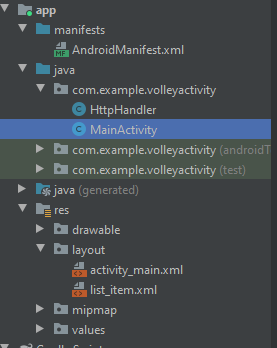
I successfully Did the Send Request portion.



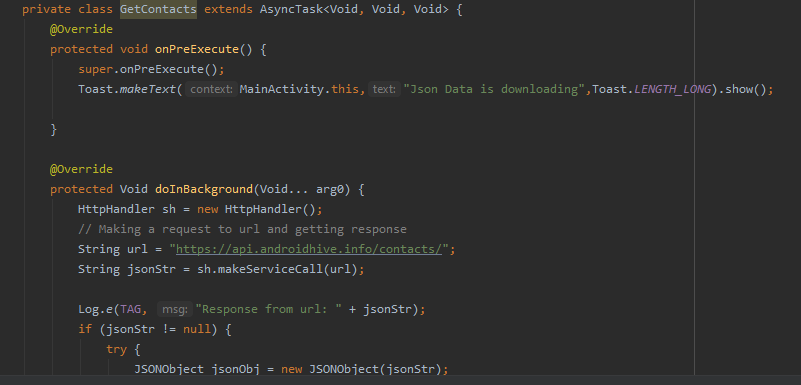


I went on the Exchange rate website and did and Exchange rate API call.

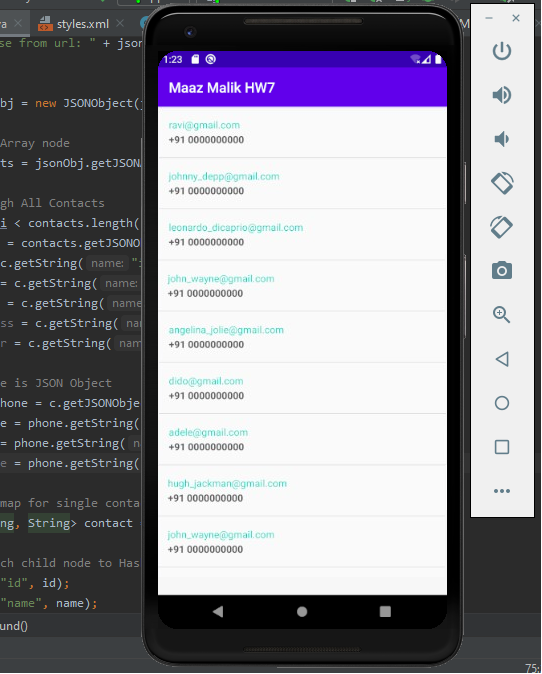
**JSON Parse**



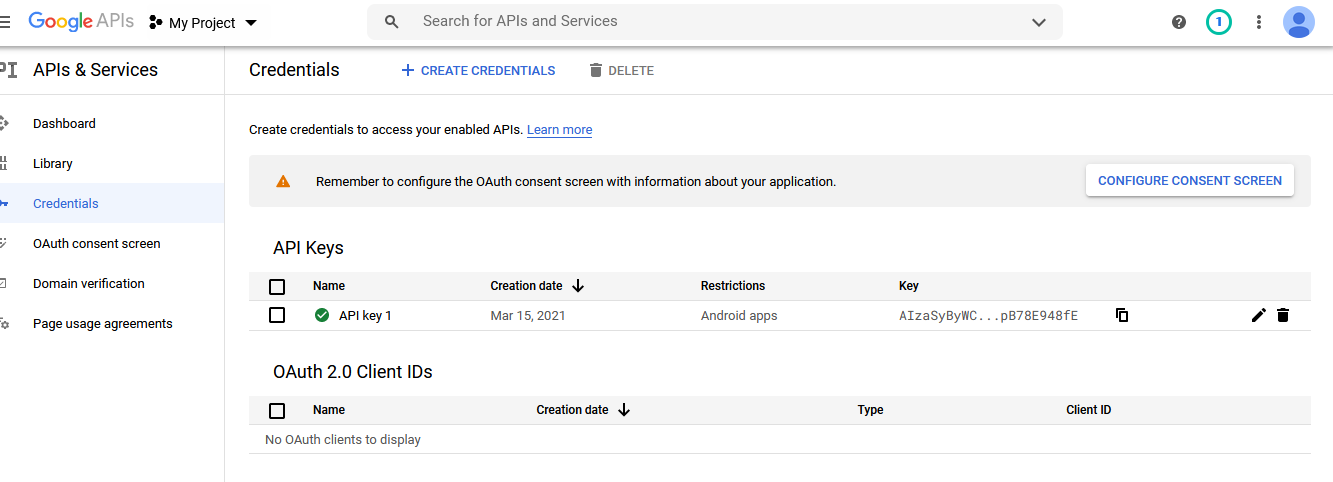
I went on the Tutorialpoints website and added the HTtpHandler and List\_item.xml files accordingly and then added in the code. I had to modify some parts.



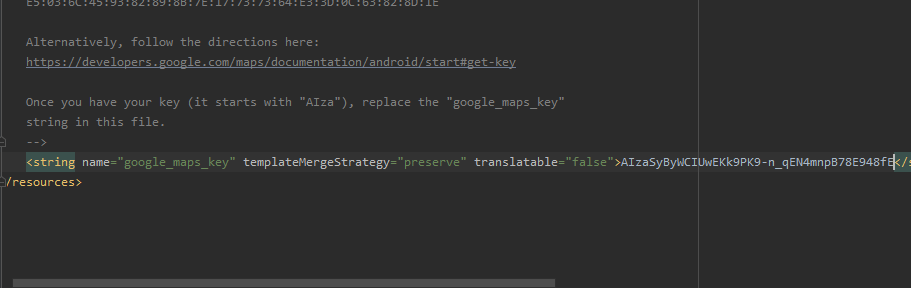
Initially I was getting the Error where the JSON call would fail. That’s when I looked in the Slack and Changed the Website to Https instead of HTTPS.

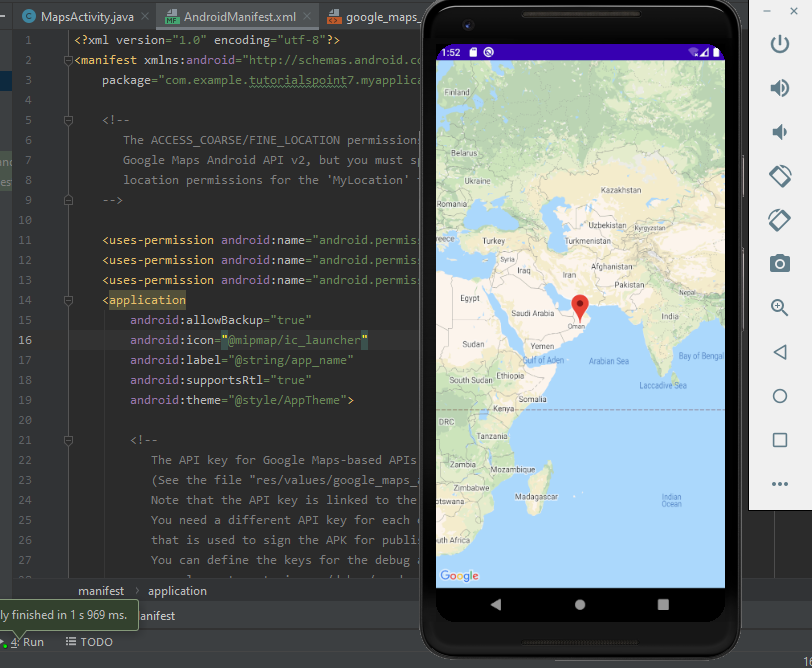


And I got JSON call to work and display all the Names and Contact info.

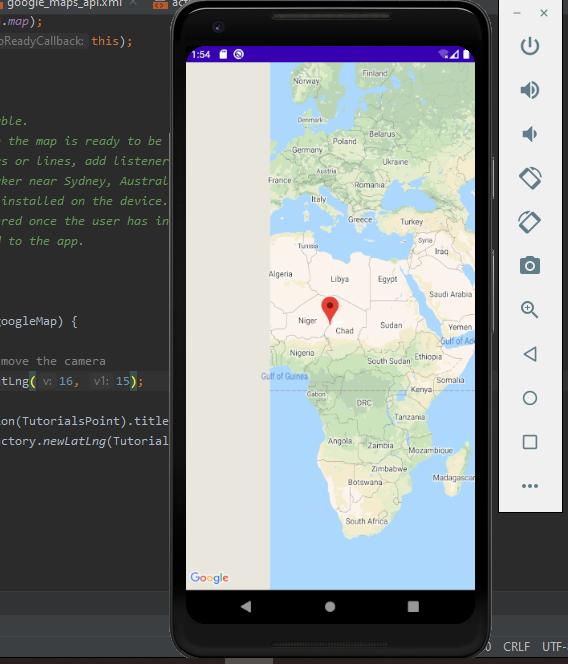


I followed the Link and created the API

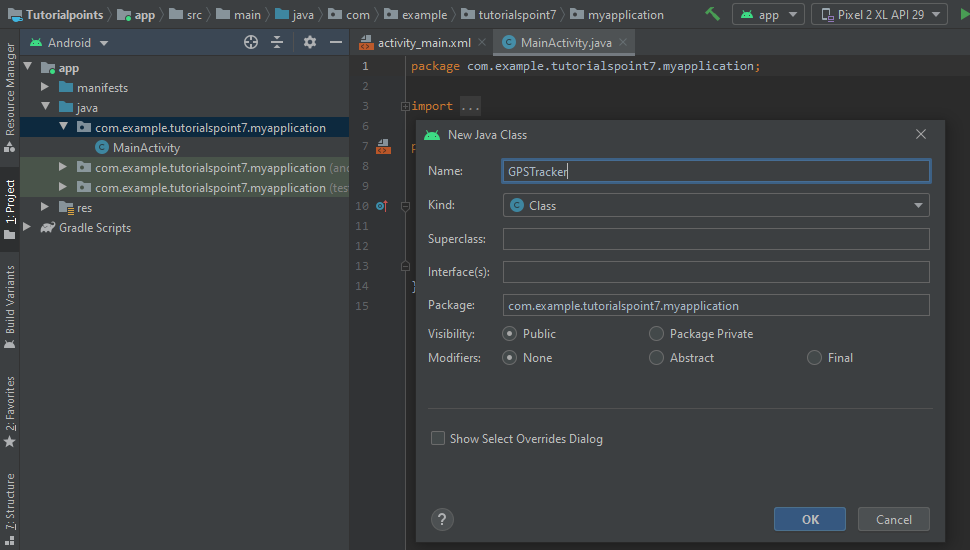




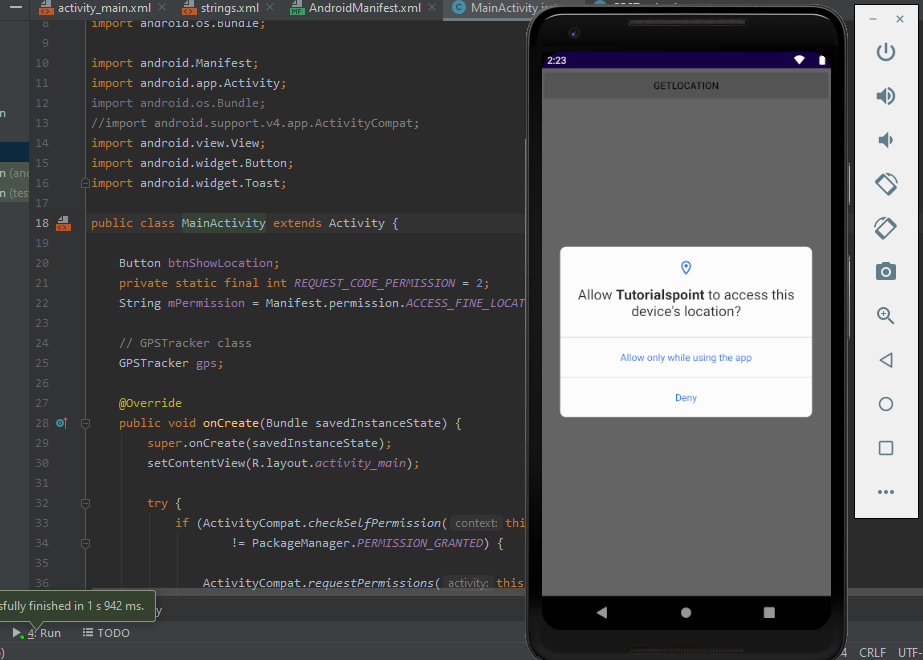
And Voila. I All I really needed to do was add the API key and change the MainActivity.Java file. A lot of the Files were left as default from what Android Studio Created and it came preloaded with Coordinates.

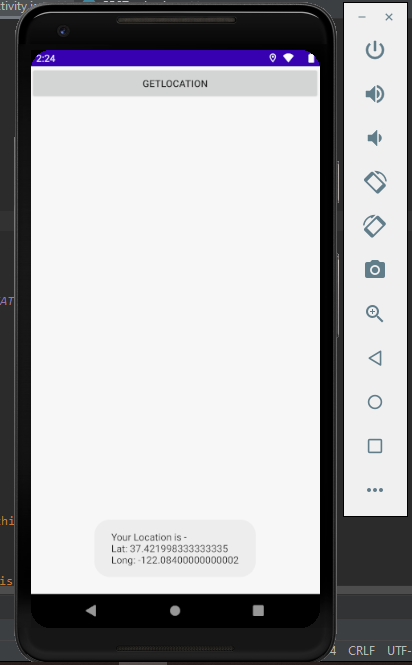


I slightly Readjusted the Coordinates in the Main.Activity folder.



I created a new file called Tutorialpoints and are creating all the necessary classes.

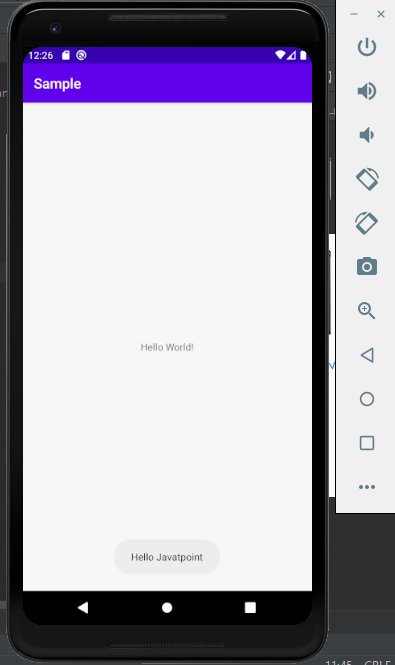




**Creating Toast**

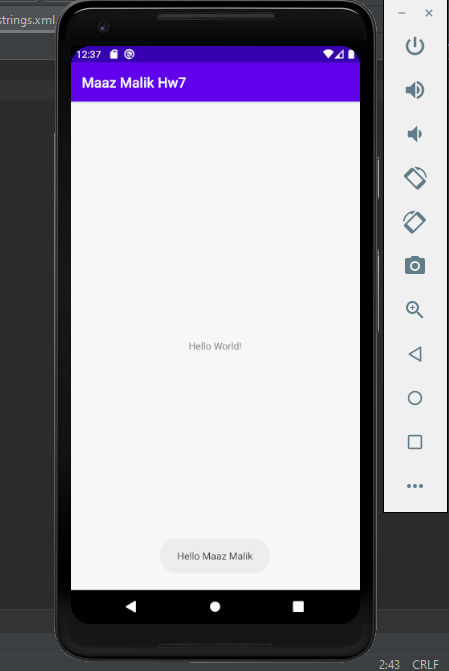
So for this part of the Assignment I actually ran into an issue that took me 2 hours to fix. The reason is because the Android app successfully ran but it would immediately Close out on the Emulator.

In all the previous exercises I had done for this assignment, We had already used Toast to make a pop up like for the Json and the map so I knew it worked. I eventually figured that the Activity\_main.xml was the issue since I changed completely to whatever was on the tutorial. It Must’ve been outdated or something but after that the Toast Call worked!



I found an online Guide from a website called Javapoint to help me. That’s why it says Hello Javapoint.

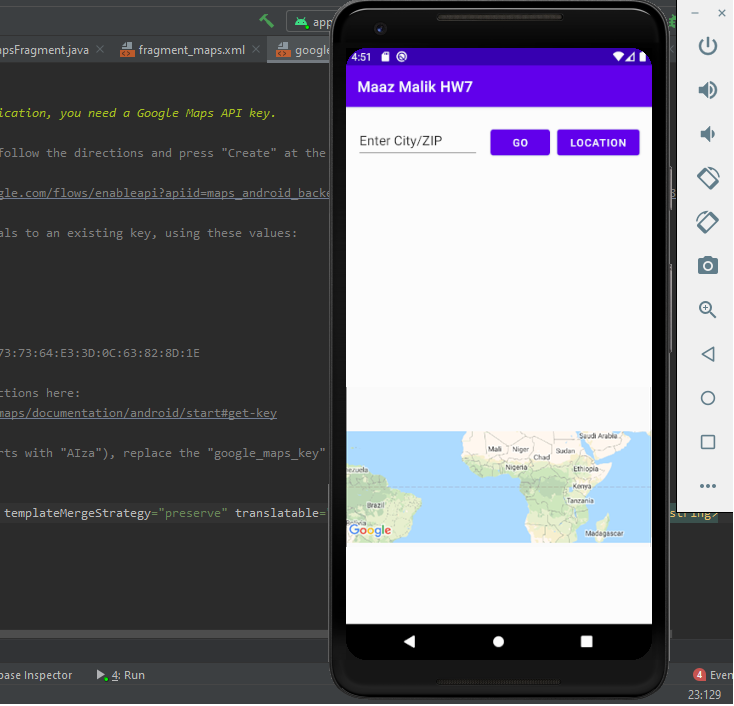
Here’s the Link: <https://www.javatpoint.com/android-toast-example>



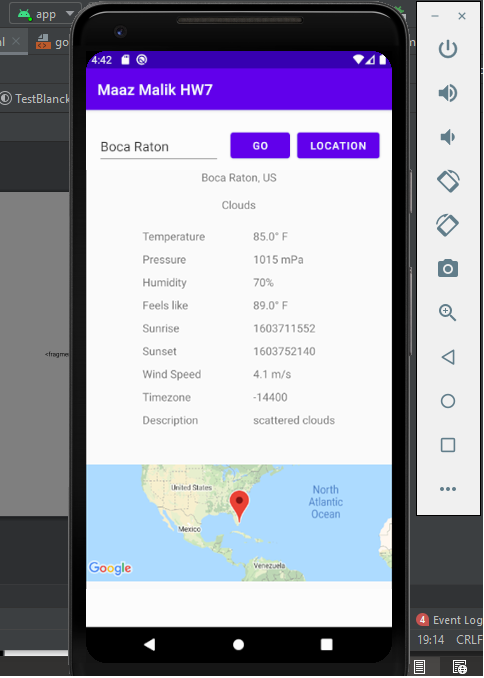
Here it is readjusted.

Part 2

Starting off with this part I started off with my previous HW4 file and combined the application sets together including the HTTPSHandler and GPSTracker. Implementing onPost, background and preexecute methods allowed me to parse the openweatherAPI.

a

I designed the App and Used the Map Fragment implementation tool that the TA showed in that video to implement this. Volley was already added and updated to the Gradle Scripts.



I used the Search function and got the Results for Boca Raton as well as the Google Maps Location. Unfortunately whenever I press the Location button nothing happens and I am trying to figure out how to fix that. I also implemented everything into one Activity\_main.xml page. It was unfortunate but due to time constraints I could not figure it out and I am very close to deadline and rushing to submit part 2. I also have retried this part with multiple different attempts including the TA’s version, the starter code the Professor Posted, and looking at the previous homework’s.

package com.example.testblanck;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import android.Manifest;  
import android.content.pm.PackageManager;  
  
  
import android.os.AsyncTask;  
import android.util.Log;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import org.json.JSONException;  
import org.json.JSONObject;  
import org.json.JSONArray;  
  
public class MainActivity extends AppCompatActivity {  
  
 public static final String *EXTRA\_MESSAGE* = "com.example.myfirstapp.MESSAGE";  
 private static final int *REQUEST\_CODE\_PERMISSION* = 2;  
 String mPermission = Manifest.permission.*ACCESS\_FINE\_LOCATION*;  
 GPSTracker gps;  
  
 private static final String *TAG* = HttpHandler.class.getSimpleName();  
  
 private String search;  
  
 private String key = "61f65b37afd4806851872c9fe3f4b7e2";  
 private String city;  
 private String country;  
 private String temperature;  
 private String feels\_like;  
 private String pressure;  
 private String humidity;  
 private String wind\_speed;  
 private String timezone;  
 private String sunrise;  
 private String sunset;  
 private String clouds;  
 private String description;  
 private String lon;  
 private String lat;  
 private GoogleMap mMap;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 try {  
 if (ActivityCompat.*checkSelfPermission*(this, mPermission)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
  
 ActivityCompat.*requestPermissions*(this, new String[]{mPermission},*REQUEST\_CODE\_PERMISSION*);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void locationOnClick(View view)  
 {  
 EditText edittext = (EditText) findViewById(R.id.search);  
 search = edittext.getText().toString();  
  
 if(*isNumeric*(search)){  
 search = "zip=" + search;  
 }  
 else{  
 search = "q=" + search;  
 }  
 new GetJsonData().execute();  
 }  
  
 public void gpsOnClick(View view)  
 {  
 gps = new GPSTracker(MainActivity.this);  
 if(gps.canGetLocation()){  
 double latitude = gps.getLatitude();  
 double longitude = gps.getLongitude();  
  
 search = "lat=" + latitude + "&lon=" + longitude;  
  
 new GetJsonData().execute();  
 }else{  
  
 gps.showSettingsAlert();  
 }  
 }  
  
  
 public static boolean isNumeric(String str) {  
 try {  
 Double.*parseDouble*(str);  
 return true;  
 } catch(NumberFormatException e){  
 return false;  
 }  
 }  
  
 public String toF(String temp)  
 {  
 double num = Double.*parseDouble*(temp);  
 num = ((num - 273.15)\*1.8)+32;  
 num = Math.*round*(num);  
  
 return String.*valueOf*(num) + "\u00B0 F";  
 }  
  
  
 private class GetJsonData extends AsyncTask<Void, Void, Void>  
 {  
  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
  
 }  
  
 @Override  
 protected Void doInBackground(Void... voids) {  
  
 String url = "https://api.openweathermap.org/data/2.5/weather?" + search + "&appid=" + key;  
  
 HttpHandler sh = new HttpHandler();  
 String jsonStr = sh.makeServiceCall(url);  
  
 Log.*e*(*TAG*, "Response from url: " + jsonStr);  
 if(jsonStr != null){  
 try{  
  
 JSONObject json = new JSONObject(jsonStr);  
 city = json.getString("name");  
 timezone = json.getString("timezone");  
  
 JSONObject sys = json.getJSONObject("sys");  
 country = sys.getString("country");  
 sunrise = sys.getString("sunrise");  
 sunset = sys.getString("sunset");  
  
 JSONObject main = json.getJSONObject("main");  
 temperature = toF(main.getString("temp"));  
 feels\_like = toF(main.getString("feels\_like"));  
 pressure = main.getString("pressure") + " mPa";  
 humidity = main.getString("humidity") + "%";  
  
 JSONObject wind = json.getJSONObject("wind");  
 wind\_speed = wind.getString("speed") + " m/s";  
  
 JSONArray weather = json.getJSONArray("weather");  
 JSONObject w = weather.getJSONObject(0);  
 clouds = w.getString("main");  
 description = w.getString("description");  
  
 JSONObject coord = json.getJSONObject("coord");  
 lon = coord.getString("lon");  
 lat = coord.getString("lat");  
  
 }  
 catch(final JSONException e){  
  
 }  
 }  
  
  
 return null;  
 }  
  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 String temp;  
 TextView textView = findViewById(R.id.row1);  
 temp = "Temperature ";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row2);  
 temp = "Pressure";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row3);  
 temp = "Humidity";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row4);  
 temp = "Feels like";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row5);  
 temp = "Sunrise";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row6);  
 temp = "Sunset";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row7);  
 temp = "Wind Speed";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row8);  
 temp = "Timezone";  
 textView.setText(temp);  
  
 textView = findViewById(R.id.row9);  
 temp = "Description";  
 textView.setText(temp);  
  
  
  
 textView = findViewById(R.id.place);  
 temp = city + ", " + country;  
 textView.setText(temp);  
  
 textView = findViewById(R.id.display1);  
 textView.setText(clouds);  
  
 textView = findViewById(R.id.temperature);  
 textView.setText(temperature);  
  
 textView = findViewById(R.id.pressure);  
 textView.setText(pressure);  
  
 textView = findViewById(R.id.humidity);  
 textView.setText(humidity);  
  
 textView = findViewById(R.id.feelslike);  
 textView.setText(feels\_like);  
  
 textView = findViewById(R.id.sunrise);  
 textView.setText(sunrise);  
  
 textView = findViewById(R.id.sunset);  
 textView.setText(sunset);  
  
 textView = findViewById(R.id.windspeed);  
 textView.setText(wind\_speed);  
  
 textView = findViewById(R.id.timezone);  
 textView.setText(timezone);  
  
 textView = findViewById(R.id.description);  
 textView.setText(description);  
  
 SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  
 .findFragmentById(R.id.*map*);  
 mapFragment.getMapAsync(new OnMapReadyCallback() {  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 mMap = googleMap;  
  
 double lati = Double.*parseDouble*(lat);  
 double longi = Double.*parseDouble*(lon);  
  
 LatLng TutorialsPoint = new LatLng(lati, longi);  
 mMap.addMarker(new MarkerOptions().position(TutorialsPoint).title("Your Location"));  
 mMap.moveCamera(CameraUpdateFactory.*newLatLng*(TutorialsPoint));  
  
 }  
 });  
  
 }  
  
  
 }  
}

Activity

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 android:gravity="center\_vertical"  
 android:orientation="vertical"  
 tools:context=".MainActivity"> <TextView  
 android:id="@+id/text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 android:text="My Weather"  
 android:textSize="30sp" /> <TextView  
 android:id="@+id/tempView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 android:gravity="center\_vertical"  
 android:text="Temperature" /> <TextView  
 android:id="@+id/latlonView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 android:text="lat/lon" /> <TextView  
 android:id="@+id/countryView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal|center\_vertical"  
 android:text="country" />  
  
 <fragment  
 android:id="@+id/map"  
 android:name="com.google.android.gms.maps.SupportMapFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="225dp"  
 tools:context="com.example.tutorialspoint7.myapplication.MapsActivity" />  
  
 <TextView  
 android:id="@+id/email"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingBottom="2dip"  
 android:textColor="@color/colorAccent" />  
  
 <TextView  
 android:id="@+id/mobile"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="#5d5d5d"  
 android:textStyle="bold" />  
   
   
  
</LinearLayout>

Android Manifest

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.hw7">  
  
 <!--  
 The ACCESS\_COARSE/FINE\_LOCATION permissions are not required to use  
 Google Maps Android API v2, but you must specify either coarse or fine  
 location permissions for the "MyLocation" functionality.  
 -->  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.WAKE\_LOCK" />  
  
 <uses-feature android:name="android.hardware.type.watch" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
  
 <!--  
 The API key for Google Maps-based APIs is defined as a string resource.  
 (See the file "res/values/google\_maps\_api.xml").  
 Note that the API key is linked to the encryption key used to sign the APK.  
 You need a different API key for each encryption key, including the release key that is used to  
 sign the APK for publishing.  
 You can define the keys for the debug and release targets in src/debug/ and src/release/.  
 -->  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="@string/google\_maps\_key" />  
  
 <activity android:name=".MapActivity" />  
 <activity android:name=".WeatherDisplay" />  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

Maps

package com.example.hw7;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
  
public class MapsFragment extends Fragment {  
  
 private OnMapReadyCallback callback = new OnMapReadyCallback() {  
  
 */\*\*  
 \* Manipulates the map once available.  
 \* This callback is triggered when the map is ready to be used.  
 \* This is where we can add markers or lines, add listeners or move the camera.  
 \* In this case, we just add a marker near Sydney, Australia.  
 \* If Google Play services is not installed on the device, the user will be prompted to  
 \* install it inside the SupportMapFragment. This method will only be triggered once the  
 \* user has installed Google Play services and returned to the app.  
 \*/* @Override  
 public void onMapReady(GoogleMap googleMap) {  
 LatLng sydney = new LatLng(-34, 151);  
 googleMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));  
 googleMap.moveCamera(CameraUpdateFactory.*newLatLng*(sydney));  
 }  
 };  
  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater,  
 @Nullable ViewGroup container,  
 @Nullable Bundle savedInstanceState) {  
 return inflater.inflate(R.layout.*fragment\_maps*, container, false);  
 }  
  
 @Override  
 public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {  
 super.onViewCreated(view, savedInstanceState);  
 SupportMapFragment mapFragment =  
 (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.*map*);  
 if (mapFragment != null) {  
 mapFragment.getMapAsync(callback);  
 }  
 }  
}

https handler

package com.example.hw7;  
  
import android.util.Log;  
  
import java.io.BufferedInputStream;  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.ProtocolException;  
import java.net.URL;  
  
public class HttpHandler {  
  
 private static final String *TAG* = HttpHandler.class.getSimpleName();  
  
 public HttpHandler() {  
 }  
  
 public String makeServiceCall(String reqUrl) {  
 String response = null;  
 try {  
 URL url = new URL(reqUrl);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.setRequestMethod("GET");  
 // read the response  
 InputStream in = new BufferedInputStream(conn.getInputStream());  
 response = convertStreamToString(in);  
 } catch (MalformedURLException e) {  
 Log.*e*(*TAG*, "MalformedURLException: " + e.getMessage());  
 } catch (ProtocolException e) {  
 Log.*e*(*TAG*, "ProtocolException: " + e.getMessage());  
 } catch (IOException e) {  
 Log.*e*(*TAG*, "IOException: " + e.getMessage());  
 } catch (Exception e) {  
 Log.*e*(*TAG*, "Exception: " + e.getMessage());  
 }  
 return response;  
 }  
  
 private String convertStreamToString(InputStream is) {  
 BufferedReader reader = new BufferedReader(new InputStreamReader(is));  
 StringBuilder sb = new StringBuilder();  
  
 String line;  
 try {  
 while ((line = reader.readLine()) != null) {  
 sb.append(line).append('\n');  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 try {  
 is.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 return sb.toString();  
 }  
}

google PAI

<resources>  
 <!--  
 *TODO: Before you run your application, you need a Google Maps API key.*  
  
 To get one, follow this link, follow the directions and press "Create" at the end:  
  
 https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&keyType=CLIENT\_SIDE\_ANDROID&r=E5:03:6C:45:93:82:89:8B:7E:17:73:73:64:E3:3D:0C:63:82:8D:1E%3Bcom.example.hw7  
  
 You can also add your credentials to an existing key, using these values:  
  
 Package name:  
 com.example.hw7  
  
 SHA-1 certificate fingerprint:  
 E5:03:6C:45:93:82:89:8B:7E:17:73:73:64:E3:3D:0C:63:82:8D:1E  
  
 Alternatively, follow the directions here:  
 https://developers.google.com/maps/documentation/android/start#get-key  
  
 Once you have your key (it starts with "AIza"), replace the "google\_maps\_key"  
 string in this file.  
 -->  
 <string name="google\_maps\_key" templateMergeStrategy="preserve" translatable="false">AIzaSyByWCIUwEKk9PK9-n\_qEN4mnpB78E948fE</string>  
</resources>