* MainActivity.java :-

package com.example.practical8weathernews;  
  
public class MainActivity extends WearableActivity {  
 final String API\_KEY= "b57868d6e05293718850f411a9b05bf3";  
 final String Weather\_URL="https://api.openweathermap.org/data/2.5/weather";  
   
 final long MIN\_TIME=5000;  
 final float MIN\_DISTANCE=1000;  
 final int REQUEST\_CODE=101;  
  
 String Location\_Provider= LocationManager.GPS\_PROVIDER;  
  
 TextView NameofCity, weatherState, Temperature;  
  
 ImageView mweatherIcon;  
  
 LocationManager mLocationManager;  
 LocationListener mLocationListner;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 weatherState = findViewById(R.id.weatherCondition);  
 Temperature = findViewById(R.id.temperature);  
 mweatherIcon = findViewById(R.id.weatherIcon);  
 NameofCity = findViewById(R.id.cityName);  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
  
 Intent mIntent=getIntent();  
 String city= mIntent.getStringExtra("City");  
 getWeatherForCurrentLocation();  
 }  
  
 private void getWeatherForNewCity(String city)  
 {  
 RequestParams params=new RequestParams();  
 params.put("q",city);  
 params.put("appid",API\_KEY);  
 letsdoSomeNetworking(params);  
  
 }

private void getWeatherForCurrentLocation() {  
  
 mLocationManager = (LocationManager)

getSystemService(Context.LOCATION\_SERVICE);  
 mLocationListner = new LocationListener() {  
 @Override  
 public void onLocationChanged(Location location) {  
 String Latitude = String.valueOf(location.getLatitude());  
 String Longitude = String.valueOf(location.getLongitude());  
  
 RequestParams params =new RequestParams();  
 params.put("lat",Latitude);  
 params.put("lon",Longitude);  
 params.put("appid",API\_KEY);  
 letsdoSomeNetworking(params);  
 }  
 };  
  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 *// TODO: Consider calling  
 // ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* ActivityCompat.*requestPermissions*(this,new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*,Manifest.permission.*ACCESS\_COARSE\_LOCATION*},REQUEST\_CODE);  
 return;  
 }  
 mLocationManager.requestLocationUpdates(Location\_Provider, MIN\_TIME, MIN\_DISTANCE, mLocationListner);  
  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 if(requestCode==REQUEST\_CODE)  
 {  
 if(grantResults.length>0 && grantResults[0]==PackageManager.PERMISSION\_GRANTED)  
 {  
 Toast.makeText(MainActivity.this,"Locationget Succesffully",Toast.LENGTH\_SHORT).show();  
 getWeatherForCurrentLocation();  
 }  
 else  
 {  
 *//user denied the permission* }  
 }  
 }

private void letsdoSomeNetworking(RequestParams params)  
 {  
 AsyncHttpClient client = new AsyncHttpClient();  
 client.get(Weather\_URL,params,new JsonHttpResponseHandler()  
 {  
 @Override  
 public void onSuccess(int statusCode, Header[] headers, JSONObject response) {  
 super.onSuccess(statusCode, headers, response);  
 Toast.makeText(MainActivity.this,"Data Get Success",Toast.LENGTH\_SHORT).show();  
  
 weatherData weatherD=weatherData.fromJson(response);  
 updateUI(weatherD);  
 }  
  
 @Override  
 public void onFailure(int statusCode, Header[] headers, Throwable throwable, JSONObject errorResponse) {  
 super.onFailure(statusCode, headers, throwable, errorResponse);  
 }  
 });  
 }  
  
 private void updateUI(weatherData weather) {  
  
  
 Temperature.setText(weather.getmTemperature());  
 NameofCity.setText(weather.getMcity());  
 weatherState.setText(weather.getmWeatherType());  
 int resourceID = getResources().getIdentifier(weather.getMicon(), "drawable", getPackageName());  
 mweatherIcon.setImageResource(resourceID);  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if(mLocationManager!=null)  
 {  
 mLocationManager.removeUpdates(mLocationListner);  
 }  
 }  
}

* weatherData.java :-

package com.example.practical8weathernews;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
  
public class weatherData {  
 private String mTemperature,micon,mcity,mWeatherType;  
 private int mCondition;  
  
 public static weatherData fromJson(JSONObject jsonObject)  
 {  
  
 try  
 {  
 weatherData weatherD=new weatherData();  
 weatherD.mcity=jsonObject.getString("name");  
 weatherD.mCondition=jsonObject.getJSONArray("weather").getJSONObject(0).getInt("id");  
 weatherD.mWeatherType=jsonObject.getJSONArray("weather").getJSONObject(0).getString("main");  
 weatherD.micon=*updateWeatherIcon*(weatherD.mCondition);  
 double tempResult=jsonObject.getJSONObject("main").getDouble("temp")-273.15;  
 int roundedValue=(int)Math.*rint*(tempResult);  
 weatherD.mTemperature=Integer.*toString*(roundedValue);  
 return weatherD;  
 }  
  
  
 catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
  
  
 }  
  
  
 private static String updateWeatherIcon(int condition)  
 {  
 if(condition>=0 && condition<=300)  
 {  
 return "thunderstrom1";  
 }  
 else if(condition>=300 && condition<=500)  
 {  
 return "lightrain";  
 }  
 else if(condition>=500 && condition<=600)  
 {  
 return "shower";  
 }  
 else if(condition>=600 && condition<=700)  
 {  
 return "snow2";  
 }  
 else if(condition>=701 && condition<=771)  
 {  
 return "fog";  
 }  
  
 else if(condition>=772 && condition<=800)  
 {  
 return "overcast1";  
 }  
 else if(condition==800)  
 {  
 return "sunny";  
 }  
 else if(condition>=801 && condition<=804)  
 {  
 return "cloudy";  
 }  
 else if(condition>=900 && condition<=902)  
 {  
 return "thunderstrom1";  
 }  
 if(condition==903)  
 {  
 return "snow1";  
 }  
 if(condition==904)  
 {  
 return "sunny";  
 }  
 if(condition>=905 && condition<=1000)  
 {  
 return "thunderstrom2";  
 }  
  
 return "dunno";  
  
  
 }  
  
 public String getmTemperature() {  
 return mTemperature+"°C";  
 }  
  
 public String getMicon() {  
 return micon;  
 }  
  
 public String getMcity() {  
 return mcity;  
 }  
  
 public String getmWeatherType() {  
 return mWeatherType;  
 }  
}