**Context Menu**

registerForContextMenu(lstContacts);

public class ContactActivity extends AppCompatActivity {  
 EditText txtName, txtPhoneNo;  
 Button btnAdd, btnUpdate;  
 DatabaseHelper db;  
 ListView lstContacts;  
 ArrayAdapter<String> adapter;  
 int contactID;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_contact*);  
 ControlInitialization();  
 ButtonClicks();  
 }  
  
 private void ControlInitialization()  
 {  
 txtName = findViewById(R.id.*txtName*);  
 txtPhoneNo = findViewById(R.id.*txtPhoneNo*);  
 btnAdd = findViewById(R.id.*btnAdd*);  
 btnUpdate = findViewById(R.id.*btnUpdate*);  
 lstContacts = findViewById(R.id.*lstContacts*);  
 db = new DatabaseHelper(this);  
 BindContacts();  
 }  
  
 private void ButtonClicks()  
 {  
 btnAdd.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String Name = txtName.getText().toString();  
 String PhoneNo = txtPhoneNo.getText().toString();  
  
 int ID = db.AddContacts(new Contact(Name, PhoneNo));  
 if(ID >= 1)  
 {  
 Toast.*makeText*(getApplicationContext(),"data inserted successfully",Toast.*LENGTH\_LONG*).show();  
 BindContacts();  
 }  
 else  
 {  
 Toast.*makeText*(getApplicationContext(),"Error in insertion",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 });  
  
 btnUpdate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String name = txtName.getText().toString();  
 String PhoneNo = txtPhoneNo.getText().toString();  
  
 int ID = db.UpdateContact(new Contact(contactID,name,PhoneNo));  
 Toast.*makeText*(getApplicationContext(),ID + " ", Toast.*LENGTH\_LONG*).show();  
 if(ID >= 1)  
 {  
  
 }  
 BindContacts();  
 }  
 });  
 }  
  
 private void BindContacts()  
 {  
 ArrayList<Contact> ContactList = db.GetAllContacts();  
 MyAdapter myAdapter = new MyAdapter(ContactActivity.this,ContactList);  
 *// adapter = new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,ContactList);* lstContacts.setAdapter(myAdapter);  
 registerForContextMenu(lstContacts);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 super.onCreateContextMenu(menu, v, menuInfo);  
 menu.setHeaderTitle("Select Option");  
 menu.add(0,v.getId(),0,"Update");  
 menu.add(0,v.getId(),0,"Delete");  
 }  
  
 @Override  
 public boolean onContextItemSelected(@NonNull MenuItem item) {  
 if(item.getTitle() == "Update")  
 {  
 int ID = GetID(item);  
  
 Contact c = db.GetContactDetails(ID);  
 txtName.setText(c.get\_Name());  
 txtPhoneNo.setText(c.get\_PhoneNo());  
 contactID = c.get\_ID();  
  
 }else if(item.getTitle() == "Delete")  
 {  
 int ID = GetID(item);  
 db.DeleteContact(new Contact(ID));  
 BindContacts();  
 }  
 return true;  
 }  
  
 private int GetID(MenuItem item)  
 {  
 AdapterView.AdapterContextMenuInfo menuInfo = (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();  
 int position = menuInfo.position;  
 String SelectedItem = adapter.getItem(position).toString();  
 String[] SelectedColumn = SelectedItem.split(",");  
 int ID = Integer.*parseInt*(SelectedColumn[0]);  
 return ID;  
 }  
}

**Content Provider**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.database.MatrixCursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.ContactsContract;  
import android.telephony.SmsManager;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;  
import android.widget.SimpleCursorAdapter;  
import android.widget.Toast;  
  
public class ContactProviderActivity extends AppCompatActivity {  
 ListView lstContacts;  
 SimpleCursorAdapter adapter;  
 MatrixCursor matrixCursor;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_contact\_provider*);  
 ControlInitialization();  
 }  
  
 private void ControlInitialization() {  
 lstContacts = findViewById(R.id.*lstContacts*);  
  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_CONTACTS*) +  
 ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*SEND\_SMS*)!=  
 PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*READ\_CONTACTS*,  
 Manifest.permission.*SEND\_SMS*}, 0);  
  
 } else {  
 matrixCursor = new MatrixCursor(new String[]{"\_Id", "Name", "PhoneNo", "Email"});  
  
 Uri contactUri = ContactsContract.Contacts.*CONTENT\_URI*;  
 Log.*i*("uri", contactUri.toString());  
  
  
 Cursor contactCursor = getContentResolver().query(contactUri, null,  
 null, null,  
 ContactsContract.Contacts.*DISPLAY\_NAME* + " DESC");  
 if (contactCursor.moveToFirst()) {  
 do {  
 long contactId = contactCursor.getLong(contactCursor.getColumnIndex("\_ID"));  
 String Name = contactCursor.getString(contactCursor.getColumnIndex(ContactsContract.  
 Contacts.*DISPLAY\_NAME*));  
  
 Integer HasPhone = contactCursor.getInt(contactCursor.getColumnIndex(ContactsContract.Contacts.  
 *HAS\_PHONE\_NUMBER*));  
 String PhoneNo = "";  
 if (HasPhone > 0) {  
 Cursor cp = getContentResolver().query(ContactsContract.CommonDataKinds.Phone.*CONTENT\_URI*,  
 null,  
 ContactsContract.CommonDataKinds.Phone.*CONTACT\_ID* + "=?",  
 new String[]{String.*valueOf*(contactId)}, null);  
 if (cp != null && cp.moveToFirst()) {  
 PhoneNo = cp.getString(cp.getColumnIndex(ContactsContract.CommonDataKinds.Phone.*NUMBER*));  
 }  
 }  
  
 String Email = "";  
 Cursor ce = getContentResolver().query(ContactsContract.CommonDataKinds.Email.*CONTENT\_URI*, null,  
 ContactsContract.CommonDataKinds.Email.*CONTACT\_ID* + "=?",  
 new String[]{String.*valueOf*(contactId)}, null);  
 if (ce != null & ce.moveToFirst()) {  
 Email = ce.getString(ce.getColumnIndex(ContactsContract.CommonDataKinds.Email.*DATA*));  
 }  
  
 matrixCursor.addRow(new Object[]{Long.*toString*(contactId), Name, PhoneNo, Email});  
 } while (contactCursor.moveToNext());  
  
 adapter = new SimpleCursorAdapter(this, R.layout.*row\_item*, matrixCursor,  
 new String[]{"\_Id", "Name", "PhoneNo", "Email"}, new int[]{R.id.*lblID*, R.id.*lblName*,  
 R.id.*lblPhoneNo*,  
 R.id.*lblEmail* });  
 lstContacts.setAdapter(adapter);  
  
 lstContacts.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 String MobileNo = matrixCursor.getString(2);  
  
 */\*SmsManager manager = SmsManager.getDefault();  
 manager.sendTextMessage(MobileNo,null,"Hello...Good Morning",null,null);  
 Toast.makeText(getApplicationContext(),"SMS Sent successfully",Toast.LENGTH\_LONG).show();\*/* Intent objIntent= new Intent(Intent.*ACTION\_SENDTO*);  
 objIntent.putExtra("sms\_body","Good Morning");  
 objIntent.putExtra("address",MobileNo);  
 objIntent.setData(Uri.*parse*("smsto:"));  
 startActivity(objIntent);  
 }  
 });  
 }  
 }  
 }  
  
  
}

**File Operation**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.os.Environment;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
  
public class FileActivity extends AppCompatActivity {  
 EditText txtContent;  
 Button btnRead, btnWrite, btnWriteExternal, btnReadExternal;  
 private static String *FileName* = "MyFile.txt";  
 private static String *FilePath* = "MyStorage";  
 File MyExternalFile;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_file*);  
 ControlInitialization();  
 Buttonclicks();  
  
  
 }  
  
 private void ControlInitialization(){  
 txtContent = findViewById(R.id.*txtContent*);  
 btnRead = findViewById(R.id.*btnRead*);  
 btnWrite = findViewById(R.id.*btnWrite*);  
 btnReadExternal = findViewById(R.id.*btnReadExternal*);  
 btnWriteExternal = findViewById(R.id.*btnWriteExternal*);  
 IsExternalStorageAvailable();  
 IsExternalStorageReadOnly();  
  
 if(IsExternalStorageAvailable() && !IsExternalStorageReadOnly())  
 {  
 MyExternalFile = new File(getExternalFilesDir(*FilePath*),*FileName*);  
 }  
  
 }  
  
 private void Buttonclicks()  
 {  
 btnWrite.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try {  
 FileOutputStream fos = openFileOutput(*FileName*,*MODE\_APPEND*);  
 OutputStreamWriter writer = new OutputStreamWriter(fos);  
  
 writer.write(txtContent.getText().toString());  
 writer.flush();  
 writer.close();  
  
  
 txtContent.setText("");  
 Toast.*makeText*(getApplicationContext(), *FileName* +  
 " created successfully",Toast.*LENGTH\_LONG*).show();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 btnRead.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String line,FileContent = "";  
 try {  
 FileInputStream fis = openFileInput(*FileName*);  
  
 BufferedReader reader = new BufferedReader(new InputStreamReader(fis));  
 while((line = reader.readLine())!= null)  
 {  
 FileContent += line;  
 }  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 txtContent.setText(FileContent);  
 Toast.*makeText*(getApplicationContext(), *FileName* +  
 "successfully retrieved",Toast.*LENGTH\_LONG*).show();  
 }  
 });  
  
 btnWriteExternal.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try {  
 FileOutputStream fos =new FileOutputStream(MyExternalFile,true);  
 fos.write(txtContent.getText().toString().getBytes());  
 fos.flush();  
 fos.close();  
 txtContent.setText("");  
  
 */\* OutputStreamWriter writer = new OutputStreamWriter(fos);  
 writer.write(txtContent.getText().toString());\*/* Toast.*makeText*(getApplicationContext(), *FileName* + " created successfully",Toast.*LENGTH\_LONG*).show();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 btnReadExternal.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String line, FileContent ="";  
 try {  
 FileInputStream fis = new FileInputStream(MyExternalFile);  
 BufferedReader br = new BufferedReader(new InputStreamReader(fis));  
 while((line = br.readLine()) != null)  
 {  
 FileContent += line;  
 }  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 txtContent.setText(FileContent);  
 }  
 });  
 }  
  
 private boolean IsExternalStorageAvailable()  
 {  
 String ExtState = Environment.*getExternalStorageState*();  
 Toast.*makeText*(getApplicationContext(), ExtState, Toast.*LENGTH\_LONG*).show();  
 if(Environment.*MEDIA\_MOUNTED*.equals(ExtState))  
 {  
 return true;  
 }  
 return false;  
 }  
  
 private boolean IsExternalStorageReadOnly()  
 {  
 String ExtState = Environment.*getExternalStorageState*();  
 if(Environment.*MEDIA\_MOUNTED\_READ\_ONLY*.equals(ExtState))  
 {  
 return true;  
 }  
 return false;  
 }  
}

**Fragment**

package com.example.activitylifecycledemo;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
public class Fragment1 extends Fragment {  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 return inflater.inflate(R.layout.*fragment1*,container,false);  
 }  
}

*<?*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"  
 tools:context=".FragmentActivity"  
 android:orientation="vertical"  
 android:weightSum="2">  
 <fragment  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:name="com.example.activitylifecycledemo.Fragment1"  
 android:id="@+id/fragment1"  
 android:layout\_weight="1"/>  
 <fragment  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:name="com.example.activitylifecycledemo.Fragment2"  
 android:id="@+id/fragment2"  
 android:layout\_weight="2"/>  
</LinearLayout>

**GridView**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.GridView;  
  
public class GridActivity extends AppCompatActivity {  
 GridView gvFlowers;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_grid*);  
 ControlInitialization();  
 }  
  
 private void ControlInitialization()  
 {  
 gvFlowers = findViewById(R.id.*gvFlowers*);  
 ImageAdapter1 adapter = new ImageAdapter1(this);  
 gvFlowers.setAdapter(adapter);  
  
 gvFlowers.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent objIntent = new Intent(getApplicationContext(),SelectedImageActivity.class);  
 objIntent.putExtra("position",i);  
 startActivity(objIntent);  
 }  
 });  
 }  
}

**ImageAdapter - Class**

package com.example.activitylifecycledemo;  
  
import android.content.Context;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.GridLayout;  
import android.widget.GridView;  
import android.widget.ImageView;  
  
public class ImageAdapter1 extends BaseAdapter {  
 Integer[] FlowerImages = {R.drawable.*img1*,  
 R.drawable.*img2*, R.drawable.*img3*, R.drawable.*img4*,  
 R.drawable.*img5*, R.drawable.*img6*, R.drawable.*img7*};  
 Context c;  
 public ImageAdapter1(Context c)  
 {  
 this.c = c;  
 }  
 @Override  
 public int getCount() {  
 return FlowerImages.length;  
 }  
  
 @Override  
 public Object getItem(int i) {  
 return FlowerImages[i];  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return FlowerImages[i];  
 }  
  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
 ImageView img;  
 if(view == null)  
 {  
 img = new ImageView(c);  
 img.setLayoutParams(new GridView.LayoutParams(400,400));  
 img.setScaleType(ImageView.ScaleType.*CENTER\_CROP*);  
 img.setPadding(10,10,10,10);  
 }  
 else  
 {  
 img = (ImageView)view;  
 }  
 img.setImageResource(FlowerImages[i]);  
 return img;  
 }  
}

**Web API JSON**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
  
  
public class JsonActivity extends AppCompatActivity {  
 ListView lstPlayer;  
 String url = "https://demonuts.com/Demonuts/JsonTest/Tennis/json\_parsing.php";  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_json*);  
 ControlInitialization();  
 }  
  
 private void ControlInitialization()  
 {  
 lstPlayer = findViewById(R.id.*lstPlayers*);  
 JsonRequest();  
 }  
  
 private void JsonRequest()  
 {  
 StringRequest request = new StringRequest(Request.Method.*GET*, url,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 try {  
 JSONObject object = new JSONObject(response);  
 ArrayList<PlayerModel> PlayerModelList = new ArrayList<PlayerModel>();  
 if(object.optString("status").equals("true"))  
 {  
 JSONArray dataArray = object.getJSONArray("data");  
 for(int i=0;i<dataArray.length();i++)  
 {  
 JSONObject dataObject = dataArray.getJSONObject(i);  
 */\* PlayerModelList.add(dataObject.getString("id") + "," +  
 dataObject.getString("name") + "," +  
 dataObject.getString("country") + "," +  
 dataObject.getString("city") + "," +  
 dataObject.getString("imgURL"));\*/* PlayerModelList.add(new PlayerModel(dataObject.getString("id"),dataObject.getString("name"),  
 dataObject.getString("country"),dataObject.getString("city"),dataObject.getString("imgURL")));  
 }  
  
 */\* ArrayAdapter<String> adapter = new ArrayAdapter<String>(getApplicationContext(),  
 android.R.layout.simple\_list\_item\_1,PlayerModelList);  
 lstPlayer.setAdapter(adapter);\*/* JsonCustomAdapter adapter = new JsonCustomAdapter(getApplicationContext(),PlayerModelList);  
 lstPlayer.setAdapter(adapter);  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 error.printStackTrace();  
 }  
 });  
  
 RequestQueue queue = Volley.*newRequestQueue*(this);  
 queue.add(request);  
 }  
  
}

**JSON Custom Adapter**

package com.example.activitylifecycledemo;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
  
public class JsonCustomAdapter extends BaseAdapter {  
 ArrayList<PlayerModel> playerModels;  
 Context context;  
 TextView txtId, txtName, txtCountry, txtCity;  
 ImageView img;  
  
 public JsonCustomAdapter(Context c,ArrayList<PlayerModel> playerModel)  
 {  
 this.context = c;  
 this.playerModels = playerModel;  
 }  
 @Override  
 public int getCount() {  
 return playerModels.size();  
 }  
  
 @Override  
 public Object getItem(int i) {  
 return i;  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return i;  
 }  
  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
 view = LayoutInflater.*from*(context).inflate(R.layout.*json\_list\_item*,viewGroup,false);  
  
 txtId = view.findViewById(R.id.*txtId*);  
 txtName = view.findViewById(R.id.*txtName*);  
 txtCountry = view.findViewById(R.id.*txtCountry*);  
 txtCity = view.findViewById(R.id.*txtCity*);  
 img = view.findViewById(R.id.*img*);  
  
 txtId.setText(playerModels.get(i).getId());  
 txtName.setText(playerModels.get(i).getName());  
 txtCity.setText(playerModels.get(i).getCity());  
 txtCountry.setText(playerModels.get(i).getCountry());  
  
 Picasso.*with*(context).load(playerModels.get(i).getImageURL()).into(img);  
 return view;  
 }  
}

**Notification**

NotificationCompat.Builder buider = new NotificationCompat.Builder(context)  
 .setSmallIcon(R.drawable.*img1*)  
 .setContentTitle("New App installed")  
 .setContentText("My Receiver called")  
 .setAutoCancel(true)  
 .setPriority(NotificationCompat.*PRIORITY\_DEFAULT*);  
  
NotificationManager nm = (NotificationManager) context.getSystemService(Context.*NOTIFICATION\_SERVICE*);  
nm.notify(0, buider.build());

**Download**

DownloadManager manager = (DownloadManager)getBaseContext().getSystemService  
 (*DOWNLOAD\_SERVICE*);  
Uri Download\_URI = Uri.*parse*(url.toString());  
  
DownloadManager.Request request = new DownloadManager.Request(Download\_URI);  
request.setAllowedNetworkTypes(DownloadManager.Request.*NETWORK\_WIFI* |  
 DownloadManager.Request.*NETWORK\_MOBILE*);  
request.setAllowedOverRoaming(false);  
request.setNotificationVisibility(DownloadManager.  
 Request.*VISIBILITY\_VISIBLE\_NOTIFY\_COMPLETED*);  
request.allowScanningByMediaScanner();  
request.setDestinationInExternalPublicDir(Environment.*DIRECTORY\_DOWNLOADS*,  
 "MyDownloads/" + "Sample");  
  
manager.enqueue(request);

**Telephony API**

package com.example.activitylifecycledemo;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import android.Manifest;  
import android.content.Context;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.telephony.PhoneStateListener;  
import android.telephony.TelephonyManager;  
import android.widget.TextView;  
  
public class TelephonyActivity extends AppCompatActivity {  
 TextView txtPhoneDetails;  
 TelephonyManager tm;  
 PhoneStateListener listenr;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_telephony*);  
 ControlInitialization();  
 }  
  
 private void ControlInitialization() {  
 txtPhoneDetails = findViewById(R.id.*txtPhoneDetails*);  
 tm = (TelephonyManager) getSystemService(Context.*TELEPHONY\_SERVICE*);  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_PHONE\_STATE*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*READ\_PHONE\_STATE*},  
 1);  
 }  
 else  
 {  
 PhoneDetails();  
 }  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 if (requestCode == 1) {  
 PhoneDetails();  
 }  
 }  
  
 private void PhoneDetails() {  
 String IMEINO = tm.getDeviceId();  
 String SimSerialNo = tm.getSimSerialNumber();  
 String NetworkCountryISO = tm.getNetworkCountryIso();  
 String SimCountryISO = tm.getSimCountryIso();  
  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_PHONE\_STATE*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
  
 return;  
 }  
 String SoftwareVersion = tm.getDeviceSoftwareVersion();  
 String VoiceMail = tm.getVoiceMailNumber();  
 boolean IsROming = tm.isNetworkRoaming();  
  
 String strPhoneType = "";  
 int PhoneType = tm.getPhoneType();  
 switch (PhoneType)  
 {  
 case TelephonyManager.*PHONE\_TYPE\_NONE*:  
 strPhoneType = "None";  
 break;  
 case TelephonyManager.*PHONE\_TYPE\_CDMA*:  
 strPhoneType = "CDMA";  
 break;  
 case TelephonyManager.*PHONE\_TYPE\_GSM*:  
 strPhoneType = "GSM";  
 break;  
 }  
 String PhoneDetails = " Phone Details : \n" +  
 "IMEI No : " + IMEINO + "\n" +  
 "Sim Serial No : " + SimSerialNo + "\n" +  
 "Network Country ISO : " + NetworkCountryISO + "\n" +  
 "Sim Country ISO : " + SimCountryISO + "\n" +  
 "Software Version : " + SoftwareVersion + "\n" +  
 "Voice Mail : " + VoiceMail + "\n" +  
 "Is Roming ? " + IsROming + "\n" +  
 "Phone Type : " + strPhoneType + "\n";  
  
 listenr = new PhoneStateListener()  
 {  
 @Override  
 public void onCallStateChanged(int state, String phoneNumber) {  
 String State = "";  
 switch (state)  
 {  
 case TelephonyManager.*CALL\_STATE\_IDLE*:  
 State = "Idle";  
 break;  
 case TelephonyManager.*CALL\_STATE\_RINGING*:  
 State = "Ringing";  
 break;  
 case TelephonyManager.*CALL\_STATE\_OFFHOOK*:  
 State = "Received";  
 break;  
 }  
 txtPhoneDetails.append(String.*format*("\n on call state changed : %s",State));  
 }  
 };  
 tm.listen(listenr,PhoneStateListener.*LISTEN\_CALL\_STATE*);  
 txtPhoneDetails.setText(PhoneDetails);  
 }  
}

**Web View**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.webkit.WebResourceRequest;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;  
import android.widget.Button;  
  
public class webviewDemo extends AppCompatActivity {  
 WebView wv;  
 Button btnOpen;  
 String URL;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_webview\_demo*);  
 ControlInitialization();  
 ButtonClicks();  
 }  
  
 *//Initialization of controls* private void ControlInitialization()  
 {  
 wv = findViewById(R.id.*wv*);  
 wv.setWebViewClient(new MyBrowser());  
 btnOpen = findViewById(R.id.*btnOpen*);  
 }  
  
 *//Button Clicks* private void ButtonClicks()  
 {  
 btnOpen.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 URL = "https://www.google.com";  
 wv.getSettings().setJavaScriptEnabled(true);  
 wv.getSettings().setLoadsImagesAutomatically(true);  
 wv.loadUrl(URL);  
 }  
 });  
 }  
  
 private class MyBrowser extends WebViewClient  
 {  
 @Override  
 public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {  
 view.loadUrl(URL);  
 return true;  
 }  
 }  
  
  
}

**Bounded Service**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Service;  
import android.content.ComponentName;  
import android.content.Context;  
import android.content.Intent;  
import android.content.ServiceConnection;  
import android.os.Bundle;  
import android.os.IBinder;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import org.w3c.dom.Text;  
  
public class BoundedServiceActivity extends AppCompatActivity {  
 TextView txtRandom;  
 Button btnRandom;  
 MyBoundedService boundedService;  
 boolean IsBounded = false;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_bounded\_service*);  
  
 ControlInitialization();  
 }  
  
 private void ControlInitialization()  
 {  
 txtRandom = findViewById(R.id.*txtRandom*);  
 btnRandom = findViewById(R.id.*btnRandom*);  
 Intent i = new Intent(this,MyBoundedService.class);  
 bindService(i,serviceConnection,Context.*BIND\_AUTO\_CREATE*);  
  
 btnRandom.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 txtRandom.setText(String.*valueOf*(boundedService.getRandom()));  
 }  
 });  
 }  
  
 private ServiceConnection serviceConnection = new ServiceConnection() {  
 @Override  
 public void onServiceConnected(ComponentName componentName, IBinder iBinder) {  
 MyBoundedService.LocalBinder binder = (MyBoundedService.LocalBinder)iBinder;  
 boundedService = binder.getService();  
 IsBounded = true;  
 }  
  
 @Override  
 public void onServiceDisconnected(ComponentName componentName) {  
 IsBounded = false;  
 }  
 };  
  
}

**MyBoundedService**

package com.example.activitylifecycledemo;  
  
import android.app.Service;  
import android.content.Intent;  
import android.os.Binder;  
import android.os.IBinder;  
  
import androidx.annotation.Nullable;  
  
import java.util.Random;  
  
public class MyBoundedService extends Service {  
 private IBinder iBinder = new LocalBinder();  
 private Random mGenerator = new Random();  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return iBinder;  
 }  
  
 public class LocalBinder extends Binder  
 {  
 MyBoundedService getService()  
 {  
 return MyBoundedService.this;  
 }  
 }  
  
 public int getRandom()  
 {  
 return mGenerator.nextInt(500);  
 }  
}

**IntentService**

package com.example.activitylifecycledemo;  
  
import android.annotation.SuppressLint;  
import android.app.DownloadManager;  
import android.app.IntentService;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Environment;  
import android.os.SystemClock;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
import androidx.localbroadcastmanager.content.LocalBroadcastManager;  
  
  
import java.net.MalformedURLException;  
import java.net.URL;  
  
public class MyIntentService extends IntentService {  
 public MyIntentService() {  
 super("My Intent Service");  
 }  
  
 @SuppressLint("WrongThread")  
 @Override  
 protected void onHandleIntent(@Nullable Intent intent) {  
 String message = intent.getStringExtra("message");  
 Log.*d*("message",message);  
 SystemClock.*sleep*(3000);  
 intent.setAction(ServiceActivity.*FILTER\_KEY*);  
 String broadcastMessage = "Intent service is called after the " +  
 "pause of 3 seconds " + message;  
 LocalBroadcastManager.*getInstance*(getApplicationContext()).sendBroadcast(  
 intent.putExtra("broadcastMessage",broadcastMessage));  
 */\* DownloadManager manager = (DownloadManager)getBaseContext().getSystemService(DOWNLOAD\_SERVICE);  
  
 Uri download = Uri.parse("http://www.gadgetsaint.com/wp-content/uploads/2016/11/cropped-web\_hi\_res\_512.png");  
  
 DownloadManager.Request request = new DownloadManager.Request(download);  
 request.setAllowedNetworkTypes(DownloadManager.Request.NETWORK\_WIFI |  
 DownloadManager.Request.NETWORK\_MOBILE);  
 request.setAllowedOverRoaming(false);  
 request.allowScanningByMediaScanner();  
 request.setNotificationVisibility(DownloadManager.Request.VISIBILITY\_VISIBLE\_NOTIFY\_ONLY\_COMPLETION);  
 request.setDestinationInExternalPublicDir(Environment.DIRECTORY\_DOWNLOADS,"/MyIntentDownloads/" + "Sample");  
  
 manager.enqueue(request);\*/* }  
  
  
}

**Service**

package com.example.activitylifecycledemo;  
  
import android.app.DownloadManager;  
import android.app.NotificationManager;  
import android.app.Service;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.net.Uri;  
import android.os.AsyncTask;  
import android.os.Environment;  
import android.os.IBinder;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
  
import java.net.MalformedURLException;  
import java.net.URI;  
import java.net.URL;  
import java.util.Timer;  
import java.util.TimerTask;  
  
public class MyService extends Service {  
 Timer timer = new Timer();  
 int count = 0;  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 *//Toast.makeText(getApplicationContext(),  
 // "Service is created",Toast.LENGTH\_LONG).show();* }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 *// PrintMessageRepetedly();* try {  
 new BackgroundTasks().execute(new URL("http://www.gadgetsaint.com/wp-content/uploads/2016/11/cropped-web\_hi\_res\_512.png"));  
 new BackgroundTasks().execute(new URL("http://utu.ac.in/bmiit/download/2021-22/Sem7/AAD\_AP.pdf"));  
 } catch (MalformedURLException e) {  
 e.printStackTrace();  
 }  
 return *START\_STICKY*;  
 }  
  
 private class BackgroundTasks extends AsyncTask<URL,Integer,Integer>  
 {  
 @Override  
 protected Integer doInBackground(URL... urls) {  
 int count = urls.length;  
 int progress = 0;  
 for(int i=0; i<count;i++)  
 {  
 FileDownload(urls[i]);  
 progress += (((i+1)/count) \* 100);  
 publishProgress(progress);  
 }  
 return progress;  
 }  
  
 @Override  
 protected void onPostExecute(Integer integer) {  
 Toast.*makeText*(getApplicationContext(),"Download completed",  
 Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onProgressUpdate(Integer... values) {  
 Toast.*makeText*(getApplicationContext(), " Downloaded " + values[0] + "%",  
 Toast.*LENGTH\_LONG*).show();  
  
 }  
 }  
 private void FileDownload(URL url)  
 {  
 DownloadManager manager = (DownloadManager)getBaseContext().getSystemService  
 (*DOWNLOAD\_SERVICE*);  
 Uri Download\_URI = Uri.*parse*(url.toString());  
  
 DownloadManager.Request request = new DownloadManager.Request(Download\_URI);  
 request.setAllowedNetworkTypes(DownloadManager.Request.*NETWORK\_WIFI* |  
 DownloadManager.Request.*NETWORK\_MOBILE*);  
 request.setAllowedOverRoaming(false);  
 request.setNotificationVisibility(DownloadManager.  
 Request.*VISIBILITY\_VISIBLE\_NOTIFY\_COMPLETED*);  
 request.allowScanningByMediaScanner();  
 request.setDestinationInExternalPublicDir(Environment.*DIRECTORY\_DOWNLOADS*,  
 "MyDownloads/" + "Sample");  
  
 manager.enqueue(request);  
 }  
 private void PrintMessageRepetedly()  
 {  
 timer.scheduleAtFixedRate(new TimerTask() {  
 @Override  
 public void run() {  
 Log.*d*("Repeated Service",String.*valueOf*(++count));  
 }  
 },0,1000);  
 }  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 Toast.*makeText*(getApplicationContext(),  
 "Service is stopped",Toast.*LENGTH\_LONG*).show();  
 if(timer != null)  
 {  
 timer.cancel();  
 }  
 }  
  
  
}

**Service LocalBroadcast**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.localbroadcastmanager.content.LocalBroadcastManager;  
  
  
import android.Manifest;  
import android.app.ProgressDialog;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.content.pm.PackageManager;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.URL;  
  
public class ServiceActivity extends AppCompatActivity {  
 Button btnStartService, btnStopService;  
 EditText txtMessage;  
 TextView lblDetails;  
 MyReceiver receiver;  
 public static String *FILTER\_KEY* = "broadcastkey";  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_service*);  
 ControlInitialization();  
  
 }  
 private void ControlInitialization()  
 {  
 btnStartService = findViewById(R.id.*btnStartService*);  
 btnStopService = findViewById(R.id.*btnStopService*);  
 lblDetails = findViewById(R.id.*lblDetails*);  
 txtMessage = findViewById(R.id.*txtMessage*);  
  
  
 btnStartService.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 *//String Message = txtMessage.getText().toString();* Intent objIntent = new Intent(getApplicationContext(),  
 MyService.class);  
 *//objIntent.putExtra("message",Message);* startService(objIntent);  
  
 }  
 });  
  
 btnStopService.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 stopService(new Intent(ServiceActivity.this,  
 MyService.class));  
 }  
 });  
  
  
 }  
  
 @Override  
 protected void onStart() {  
 setReceiver();  
 super.onStart();  
 }  
  
 @Override  
 protected void onDestroy() {  
 unregisterReceiver(receiver);  
 super.onDestroy();  
 }  
  
 private void setReceiver()  
 {  
 receiver = new MyReceiver();  
 IntentFilter filter = new IntentFilter();  
 filter.addAction(*FILTER\_KEY*);  
 LocalBroadcastManager.*getInstance*(getApplicationContext()).  
 registerReceiver(receiver,filter);  
 }  
 private class MyReceiver extends BroadcastReceiver  
 {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String Message = intent.getStringExtra("broadcastMessage");  
 lblDetails.setText(lblDetails.getText() + "\n" + Message);  
 }  
 }  
}

**JSON CRUD**

package com.example.activitylifecycledemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.Toast;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Map;  
  
public class StudActivity extends AppCompatActivity {  
 String url="http://192.168.56.1/StudentAPI.php";  
 String Addurl = "http://192.168.56.1/AddStudent.php";  
 String deleteurl = "http://192.168.56.1/DeleteStudent.php";  
 String updateurl = "http://192.168.56.1/UpdateStudent.php";  
 ListView lstStud;  
 EditText txtStudentName, txtCity, txtContactNo;  
 Button btnAdd;  
 int StudentID;  
 ArrayAdapter<String> adapter;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_stud*);  
 ControlInitialization();  
  
 }  
  
 private void ControlInitialization() {  
 lstStud = findViewById(R.id.*lstStud*);  
 txtStudentName = findViewById(R.id.*txtName*);  
 txtCity = findViewById(R.id.*txtCity*);  
 txtContactNo = findViewById(R.id.*txtContactNo*);  
 btnAdd = findViewById(R.id.*btnAdd*);  
 btnAdd.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 *//AddRequest();  
  
 //RequestJSON();* UpdateRequest();  
 }  
 });  
  
 lstStud.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 String selectedItem = adapter.getItem(i);  
 String[] SelectedColumn = selectedItem.split(",");  
 StudentID = Integer.*parseInt*(SelectedColumn[0]);  
 txtStudentName.setText(SelectedColumn[1]);  
 txtContactNo.setText((SelectedColumn[2]));  
 txtCity.setText(SelectedColumn[3]);  
 *// DeleteRequest(StudentID);* }  
 });  
 RequestJSON();  
  
 }  
 private void UpdateRequest()  
 {  
 StringRequest updateRequest = new StringRequest(Request.Method.*POST*, updateurl, new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 Toast.*makeText*(getApplicationContext(),response,Toast.*LENGTH\_LONG*).show();  
 RequestJSON();  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String, String> params = new HashMap<String,String>();  
 params.put("StudentName",txtStudentName.getText().toString());  
 params.put("City",txtCity.getText().toString());  
 params.put("ContactNo",txtContactNo.getText().toString());  
 params.put("StudentID",String.*valueOf*(StudentID));  
 return params;  
 }  
 };  
 RequestQueue updateQueue = Volley.*newRequestQueue*(getApplicationContext());  
 updateQueue.add(updateRequest);  
 }  
 private void DeleteRequest(final int studID)  
 {  
 StringRequest deleteRequest = new StringRequest(Request.Method.*POST*, deleteurl, new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 Toast.*makeText*(getApplicationContext(),response,Toast.*LENGTH\_LONG*).show();  
 RequestJSON();  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 error.printStackTrace();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String,String> param = new HashMap<String,String>();  
 param.put("StudentID",String.*valueOf*(studID));  
 return param;  
 }  
 };  
  
 RequestQueue deleteQueue = Volley.*newRequestQueue*(getApplicationContext());  
 deleteQueue.add(deleteRequest);  
  
 }  
 private void AddRequest()  
 {  
 StringRequest addRequest = new StringRequest(Request.Method.*POST*, Addurl,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 Toast.*makeText*(getApplicationContext(),response,Toast.*LENGTH\_LONG*).show();  
 txtStudentName.setText("");  
 txtCity.setText("");  
 txtContactNo.setText("");  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 error.printStackTrace();  
 }  
 })  
 {  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String,String> params = new HashMap<String,String>();  
 params.put("StudentName",txtStudentName.getText().toString());  
 params.put("City",txtCity.getText().toString());  
 params.put("ContactNo",txtContactNo.getText().toString());  
 return params;  
 }  
 };  
  
 RequestQueue addqueue = Volley.*newRequestQueue*(getApplicationContext());  
 addqueue.add(addRequest);  
 }  
  
   
 private void RequestJSON()  
 {  
 StringRequest request = new StringRequest(Request.Method.*GET*, url, new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 try {  
 Log.*d*("Stringobject",response);  
 JSONObject object = new JSONObject(response);  
 JSONArray dataArray = object.getJSONArray("stud");  
 ArrayList<String> StudentArrayList = new ArrayList<String>();  
 for(int i=0;i<dataArray.length();i++)  
 {  
 JSONObject dataObject = dataArray.getJSONObject(i);  
 StudentArrayList.add(dataObject.getString("StudentID") + "," +  
 dataObject.getString("StudentName") + "," +  
 dataObject.getString("City") + "," +  
 dataObject.getString("ContactNo"));  
 }  
  
 adapter = new ArrayAdapter<String>(getApplicationContext(),  
 android.R.layout.*simple\_list\_item\_1*,StudentArrayList);  
 lstStud.setAdapter(adapter);  
  
  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 error.printStackTrace();  
 }  
 }){  
 @Override  
 public Priority getPriority() {  
 return Priority.*LOW*;  
 }  
 };  
  
 JsonObjectRequest jsonRequest = new JsonObjectRequest(url, null, new Response.Listener<JSONObject>() {  
 @Override  
 public void onResponse(JSONObject response) {  
 Log.*d*("Json response","JSOn REsponse");  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
  
 }  
 }){  
 @Override  
 public Priority getPriority() {  
 return Priority.*IMMEDIATE*;  
 }  
 };  
  
 RequestQueue queue = Volley.*newRequestQueue*(this);  
 queue.add(request);  
 queue.add(jsonRequest);  
 }  
}