Contents

[MainActivity.Java……….. 1](#_Toc113703578)

[Activity\_mail.xml 3](#_Toc113703579)

[MyContentProvider.Java 6](#_Toc113703580)

[DatabaseHelper.Java 10](#_Toc113703581)

Application Name: CustomContentProvider

……………………………………………………….

# MainActivity.Java………..

package com.example.customcontentprovider;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.content.ContentResolver;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.inputmethod.InputMethodManager;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.net.URI;  
  
public class MainActivity extends AppCompatActivity {  
 EditText textname;  
 TextView resultView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 textname=findViewById(R.id.*textName*);  
 *// inserting complete table details in this text field* resultView= findViewById(R.id.*res*);  
 }  
  
  
 public void onClickAddDetails(View view) {  
 *// class to add values in the database* ContentValues values = new ContentValues();  
 String data=textname.getText().toString();  
 *// fetching text from user  
 //c.put("id", ssss)* values.put(MyContentProvider.*name*, data);  
  
 ContentResolver resolver= getContentResolver();  
 *// inserting into database through content URI* Uri uri=resolver.insert(MyContentProvider.*CONTENT\_URI*, values);  
 data= uri.toString();  
  
 Toast.*makeText*(getBaseContext(), data, Toast.*LENGTH\_LONG*).show();  
 *// displaying a toast message* Toast.*makeText*(getBaseContext(), "New Record Inserted", Toast.*LENGTH\_LONG*).show();  
 }  
  
 @SuppressLint("Range")  
 public void onClickShowDetails(View view) {  
 *// creating a cursor object of the content URI* ContentResolver resolver = getContentResolver();  
  
 Cursor cursor = resolver.query(Uri.*parse*("content://com.sonal.demo.provider/users"),  
 null, null, null, null);  
  
 *// iteration of the cursor to print whole table* if(cursor.moveToFirst()) {  
 StringBuilder strBuild=new StringBuilder();  
 while (!cursor.isAfterLast()) {  
 strBuild.append("\n"+cursor.getString(cursor.getColumnIndex("id"))+ "-"+ cursor.getString(cursor.getColumnIndex("name")));  
 cursor.moveToNext();  
 }  
 resultView.setText(strBuild);  
 }  
 else {  
 resultView.setText("No Records Found");  
 }  
 }  
  
 public void onClickDeleteDetails(View view) {  
 String name1=textname.getText().toString();  
 ContentResolver resolver= getContentResolver();  
 int count=resolver.delete(MyContentProvider.*CONTENT\_URI*,"name=?", new String[] {name1});  
 String total=Integer.*toString*(count);  
 Toast.*makeText*(this,total,Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void onClickupdateDetails(View view) {  
   
  
 }  
}

/////////////////////////////////////////////////////

# Activity\_mail.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout  
 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:background="#168BC34A"  
 tools:context=".MainActivity">  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:orientation="vertical"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.13"  
 tools:ignore="MissingConstraints">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="70dp"  
 android:text="@string/heading"  
 android:textAlignment="center"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textColor="@android:color/holo\_green\_dark"  
 android:textSize="36sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/textName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:hint="@string/hintText" />  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
 <Button  
 android:id="@+id/insertButton"  
 android:layout\_width="95dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickAddDetails"  
 android:text="@string/insertButtontext"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
  
 <Button  
 android:id="@+id/loadButton"  
 android:layout\_width="95dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickShowDetails"  
 android:text="@string/loadButtonText"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
 <Button  
 android:id="@+id/updateButton"  
 android:layout\_width="95dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickDeleteDetails"  
 android:text="@string/updateButtonText"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
  
 <Button  
 android:id="@+id/deleteButton"  
 android:layout\_width="90dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickDeleteDetails"  
 android:text="@string/deleteButtontext"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
  
</LinearLayout>  
 <TextView  
 android:id="@+id/res"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:clickable="false"  
 android:ems="10"  
 android:textColor="@android:color/holo\_green\_dark"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 </LinearLayout>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

////////////////////////////////////////////////////////////////////////////////////////////////////

# MyContentProvider.Java

package com.example.customcontentprovider;  
  
import android.content.ContentProvider;  
import android.content.ContentUris;  
import android.content.ContentValues;  
import android.content.Context;  
import android.content.UriMatcher;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteException;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.database.sqlite.SQLiteQueryBuilder;  
import android.net.Uri;  
import android.widget.Toast;  
  
import java.util.HashMap;  
  
public class MyContentProvider extends ContentProvider {  
 public MyContentProvider() {  
 }  
 *//Variables related to Database....................* private SQLiteDatabase db;  
 static final String *TABLE\_NAME* = "Users";  
 static final String *id* = "id";  
 static final String *name* = "name";  
 *//static final String marks="marks";  
  
 //Variables related to URI....................  
 // defining authority so that other application can access it* static final String *PROVIDER\_NAME* = "com.sonal.demo.provider";  
 *// defining content URI* static final String *URL* = "content://" + *PROVIDER\_NAME* + "/users";  
 *// parsing the content URI* static final Uri *CONTENT\_URI* = Uri.*parse*(*URL*);  
 static final int *uriCode* = 1;  
 static final int *uriCode2* = 2;  
 static final int *uriCode3* = 3;  
 static final UriMatcher *uriMatcher*;  
  
 private static HashMap<String, String> *values*;  
  
 static {  
 *// to match the content URI every time user access table under content provider  
 uriMatcher* = new UriMatcher(UriMatcher.*NO\_MATCH*);  
 *// to access whole table  
 uriMatcher*.addURI(*PROVIDER\_NAME*, "users", *uriCode*);  
 *// to access a particular row of the table  
 uriMatcher*.addURI(*PROVIDER\_NAME*, "users/\*", *uriCode2*);  
 *uriMatcher*.addURI(*PROVIDER\_NAME*, "users/#", *uriCode3*);  
 *uriMatcher*.addURI(*PROVIDER\_NAME*, "users/#/\*/#", *uriCode3*);  
 }  
  
 @Override  
 public boolean onCreate() {  
 *// TODO: Implement this to initialize your content provider on startup.  
 //Context context = getContext();* DatabaseHelper dbHelper = new DatabaseHelper(getContext());  
 db = dbHelper.getWritableDatabase();  
 if (db != null) {  
 return true;  
 }  
 return false;  
 }  
  
 @Override  
 public Uri insert(Uri uri, ContentValues values) {  
 *// TODO: Implement this to handle requests to insert a new row.* long rowID = db.insert(*TABLE\_NAME*, "", values);  
 if (rowID > 0) {  
 Uri \_uri = ContentUris.*withAppendedId*(*CONTENT\_URI*, rowID);  
 *//getContext().getContentResolver().notifyChange(\_uri, null);* return \_uri;  
 }  
 throw new SQLiteException("Failed to add a record into " + uri);  
 }  
 @Override  
 public Cursor query(Uri uri, String[] projection, String selection,  
 String[] selectionArgs, String sortOrder) {  
 *// TODO: Implement this to handle query requests from clients.* SQLiteQueryBuilder qb = new SQLiteQueryBuilder();  
 qb.setTables(*TABLE\_NAME*);  
 switch (*uriMatcher*.match(uri)) {  
 case *uriCode*:  
 qb.setProjectionMap(*values*);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI " + uri);  
 }  
 if (sortOrder == null || sortOrder == "") {  
 sortOrder = *id*;  
 }  
 Cursor c = qb.query(db, projection, selection, selectionArgs, null,  
 null, sortOrder);  
 c.setNotificationUri(getContext().getContentResolver(), uri);  
 return c;  
  
 }  
 @Override  
 public int delete(Uri uri, String selection, String[] selectionArgs) {  
 *// Implement this to handle requests to delete one or more rows.* int count = 0;  
 switch (*uriMatcher*.match(uri)) {  
 case *uriCode*:  
 count = db.delete(*TABLE\_NAME*, selection, selectionArgs);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI " + uri);  
 }  
 getContext().getContentResolver().notifyChange(uri, null);  
 return count;  
 }  
  
 @Override  
 public String getType(Uri uri) {  
 switch (*uriMatcher*.match(uri)) {  
 case *uriCode*:  
 return "vnd.android.cursor.dir/users";  
 default:  
 throw new IllegalArgumentException("Unsupported URI: " + uri);  
 }  
 }  
  
  
 @Override  
 public int update(Uri uri, ContentValues values, String selection,  
 String[] selectionArgs) {  
 int count = 0;  
  
 switch (*uriMatcher*.match(uri)) {  
 case *uriCode*:  
 count = db.update(*TABLE\_NAME*, values, selection, selectionArgs);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI " + uri);  
 }  
 getContext().getContentResolver().notifyChange(uri, null);  
 return count;  
 }  
  
  
}  
  
/////////////////////////

# DatabaseHelper.Java

package com.example.customcontentprovider;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
  
public class DatabaseHelper extends SQLiteOpenHelper {  
  
 *// creating object of database to perform query  
  
 // declaring name of the database* static final String *DATABASE\_NAME* = "UserDB";  
 *// declaring table name of the database* static final String *TABLE\_NAME* = "Users";  
 *// declaring version of the database* static final int *DATABASE\_VERSION* = 1;  
 *// sql query to create the table* static final String *CREATE\_DB\_TABLE* = " CREATE TABLE " + *TABLE\_NAME* + " (id INTEGER PRIMARY KEY AUTOINCREMENT, "  
 + " name TEXT NOT NULL);";  
 *// creating a database  
 // defining a constructor* DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 *// creating a table in the database* @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(*CREATE\_DB\_TABLE*);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 *// sql query to drop a table having similar name* db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
 onCreate(db);  
 }  
}