Importations

```
import android.content.DialogInterface; /*for dialog interface */

import android.os.Bundle; /*this row help to get access to R. java is an
auto-generated file by AAPT (Android Asset Packaging Tool) that contains
resource IDs for all the resources of res/directory */

import androidx.appcompat.app.AlertDialog; /*for AlertDialog class which is
in app package*/
import androidx.appcompat.app.AppCompatActivity; /*for AppCompatActivity
class this is a Base class for activities that call when you wish to use
some of the newer platforms features on older Android devices which is in
appcompat.app package*/

import android.view.View; /*helps to get access on the for view interface
which is in view package*/

import cn.pedant.SweetAlert.SweetAlertDialog; /*helps to get access on the
SweetAlertDialog interface which is in cn.pendant.SweetAlert package*/

import android.content.ContentProvider; /*helps to get access on the
ContentProvider class which is in content package*/

import android.content.ContentUris; /*helps to get access on the
ContentUris class which is in content package*/

import android.content.ContentValues; /*helps to get access on the
ContentValues class which is in content package*/

import android.content.Context; /*helps to get access on the ContentContext
class which is in content package*/

import android.content.UriMatcher; /*helps to get access on the UriMatcher
class which is in content package*/

import android.database.Cursor; /*helps to get access on the Cursor class
which is in database package*/

import android.database.sqlite.SQLiteDatabase; /*helps to get access on the
SQLiteDatabase class which is in database.sqlite package*/

import android.database.sqlite.SQLiteException; /*helps to get access on
the SQLException class which is in database.sqlite package*/

import android.database.sqlite.SQLiteOpenHelper; /*helps to get access on
the SQLiteOpenHelper class which is in database.sqlite package*/

import android.database.sqlite.SQLiteQueryBuilder; /*helps to get access on
the SQLite class which is in database.sqlite package*/

import android.net.Uri; /*helps to get access on the ContentProvider class
which is in content package*/

import java.util.HashMap; /*helps to get access on the HashMap class. The
java.util.HashMap class is the Hash table based implementation of the Map
interface */

import android.support.v4.app.Fragment; /*helps to get access on the
Fragement class which is in support.v4.app.Fragement package */
```

```
mport android.app.NotificationManager; /*It is a class which helps to
manage notifications.
*/

imiport android.app.PendingIntent; /* access the PendingIntent Class which
is in app package. A PendingIntent itself is simply a reference to a token
maintained by the system describing the original data used to retrieve it.
*/
import android.content.Context; /* helps to access Context class under
content package*/

import android.content.Intent; /* helps to access Intent class under
content package*/

import android.support.v4.app.NotificationCompat; /* helps to access
NitificationCompat class under support.v4.app package*/

import android.widget.Button; /* helps to access Button class under widget
package*/


import android.support.v7.widget.Toolbar; /* helps to access Toolbar class
under support.v7.widget package*/

import android.view.Menu; /* helps to access Menu class under view
package*/

import android.view.MenuItem; /* helps to access MenuItem class under view
package*/

import android.widget.Toast; /* helps to access Toast class under widget
package*/

import android.view.LayoutInflater; /*helps to get access on the
LayoutInflater class which is in view */

import android.view.ViewGroup; /*helps to get access on the ViewGroup class
which is in the view package */

import android.appwidget.AppWidgetManager; /*helps to get access on the
ViewGroup class which is in the view package */

import android.appwidget.AppWidgetProvider; /*helps to get access on the
AppWidgetProvider class which is in the appwidget package */

import android.widget.RemoteViews;  /*helps to get access on the ViewGroup
class which is in the view package */

import android.os.Handler;  /*helps to get access on the Handler class
which is in the os package */

import android.widget.ProgressBar; /*helps to get access on the ProgressBar
class which is in the widget package */

import android.widget.TextView; /*helps to get access on the TextView class
which is in the widget package */

import android.widget.RatingBar; /*helps to get access on the RatingBar
class which is in the widget package */
```

```
import android.widget.AdapterView; /*helps to get access on the AdapterView
class which is in the widget package */
import android.widget.ArrayAdapter; /*helps to get access on the
ArrayAdapter class which is in the widget package */

import android.widget.Spinner; /*helps to get access on the Spinner class
which is in the widget package */

import android.widget.ListView; /*helps to get access on the ListView class
which is in the widget package */
```

Description and Examples

content Provider

In android, Content Provider will act as a central repository to store the data of the application in one place and make that data available for different applications to access whenever it's required.

```
public class UsersProvider extends ContentProvider {
      static final String PROVIDER_NAME =
      "com.tutlane.contentprovider.UserProvider";
      static final String URL = "content://" + PROVIDER_NAME + "/users";
      static final Uri CONTENT_URI = Uri.parse(URL);
      static final String id = "id";
      static final String name = "name";
      static final int uriCode = 1;
      static final UriMatcher uriMatcher;
      private static HashMap<String, String> values;
      static {
            uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
            uriMatcher.addURI(PROVIDER_NAME, "users", uriCode);
            uriMatcher.addURI(PROVIDER_NAME, "users/*", uriCode);
      }
      @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 boolean onCreate() {
      Context context = getContext();
      DatabaseHelper dbHelper = new DatabaseHelper(context);
      db = dbHelper.getWritableDatabase();
      if (db != null) { return true; }
      return false;
   }
}
```

alert Dialog with SweetAlert interface

It is a pop-up box that appears in response to any action of the user. **AlertBox** is very useful when it comes to validation, it can be used to display confirmation messages.

```
public void showDialog(View view) {
      switch (view.getId()) {
            case R.id.basic_dialog:
            new SweetAlertDialog(this)
```

```
            .setTitleText("Here's a message!")
            .setContentText("This is Basic Dialog").show();
            break;
            case R.id.error_dialog:
            new SweetAlertDialog(this, SweetAlertDialog.ERROR_TYPE)
            .setTitleText("Oops...")
            .setContentText("Something went wrong!").show();
            break;
            case R.id.warning_dialog:
            new SweetAlertDialog(this, SweetAlertDialog.WARNING_TYPE)
            .setTitleText("Are you sure?")
            .setContentText("Won't be able to recover this file !")
            .setConfirmText("Yes, delete it!").show();
            break;
            case R.id.success_dialog:
            new SweetAlertDialog(this, SweetAlertDialog.SUCCESS_TYPE)
            .setTitleText("Great!")
            .setContentText("You completed this task.").show();
            break;
            case R.id.custom_dialog:
            new SweetAlertDialog(this, SweetAlertDialog.CUSTOM_IMAGE_TYPE)
            .setTitleText("Android")
            .setContentText("Thi is custom dialog")
            .setCustomImage(R.drawable.ic_android_black).show();
            break;
      }
}
```

alert Dialog

Alert Dialog shows the Alert message and gives the answer in the form of yes or no. Alert Dialog displays the message to warn you and then according to your response, the next step is processed.

```
@Override
public void onBackPressed() {
      // Create the object of AlertDialog Builder class
      AlertDialog.Builder builder = new
      AlertDialog.Builder(MainActivity.this);
      // Set the message show for the Alert time
      builder.setMessage("Do you want to exit ?");
      // Set Alert Title
      builder.setTitle("Alert !");
      // Set Cancelable false for when the user clicks on the outside the
      Dialog Box then it will remain show
      builder.setCancelable(false);
      // Set the positive button with yes name Lambda OnClickListener
      method is use of DialogInterface interface.
      builder.setPositiveButton("Yes", (DialogInterface.OnClickListener)
      (dialog, which) -> {
      // When the user click yes button then app will close
      finish();
});
 .
```

Fragment

Android **Fragment** is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity. Fragments represent multiple screen inside one activity.

```
/* this snippet code is putted in onCreate function of the activity*/

      @Override
       public View onCreateView(LayoutInflater inflater, ViewGroup
      container,
       Bundle savedInstanceState) {
       return inflater.inflate(R.layout.fragment_fragment2, container,
      false);
       }
```

Android notifications

Android **Notification** provides short, timely information about the action happened in the application, even it is not running. The notification displays the icon, title and some amount of the content text. Notification Manager. Android allows to put notification into the titlebar of your application.

```
private void addNotification() {
 NotificationCompat.Builder builder =
 new NotificationCompat.Builder(this)
 .setSmallIcon(R.drawable.messageicon) //set icon for notification
.setContentTitle("Notifications Example") //set title of notification
      setContentText("This is a notification message")//this is
      notification message
      setAutoCancel(true) // makes auto cancel of notification
      setPriority(NotificationCompat.PRIORITY_DEFAULT); //set priority of
      notification   Intent notificationIntent = new Intent(this,
      NotificationView.class);
      notificationIntent.putExtra("message", "This is a notification
      message");
      PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
      notificationIntent,
      PendingIntent.FLAG_UPDATE_CURRENT);
 builder.setContentIntent(pendingIntent); NotificationManager manager =
(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
 manager.notify(0, builder.build());
}
```

Android option menu

Android Option Menus are the primary menus of android. They can be used for settings, search, delete item etc.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
      getMenuInflater().inflate(R.menu.menu_main, menu);
      return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
      int id = item.getItemId();
           switch (id)
          {
            case R.id.item1:
            Toast.makeText(getApplicationContext(),"Item 1
            Selected",Toast.LENGTH_LONG).show();
            return true;
            case R.id.item2:
```

```
            Toast.makeText(getApplicationContext(),"Item 2
            Selected",Toast.LENGTH_LONG).show();
            return true;
            case R.id.item3:
            Toast.makeText(getApplicationContext(),"Item 3
            Selected",Toast.LENGTH_LONG).show();
             return true;
             default:
             return super.onOptionsItemSelected(item);
          }
 }
```

Widget components

Widgets are the UI components that enable users to interact with the app while saving their time.These are actually one of the best features of the android app as they display the information to the users on their screen.

```
public class my_widget_activity extends AppWidgetProvider

      public void onUpdate(Context context, AppWidgetManager
      appWidgetManager, int[] widgetIds) {
            //In this loop, Pending Intent is created which will take us to
            the URL that we've mentioned and a toast would be generated
            informing that the widget has been added.
            for (int i = 0; i < widgetIds.length; i++) {
            int my_widgetId = widgetIds[i];
            String url = "https://data-flair.training/";
            Intent intent_new = new Intent(Intent.ACTION_VIEW);
            intent_new.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent_new.setData(Uri.parse(url));
            PendingIntent pending_intent =
            PendingIntent.getActivity(context, 0, intent_new, 0);
            RemoteViews r_views = new RemoteViews(context.getPackageName(),
            R.layout.activity_main);
            r_views.setOnClickPendingIntent(R.id.btn, pending_intent);
            appWidgetManager.updateAppWidget(my_widgetId, r_views);
            Toast.makeText(context, "widget has been added",
            Toast.LENGTH_SHORT).show();
            }
      }
}
```

Android progressBar

ProgressBar is a user interface control that is used to indicate the progress of an operation. For example, downloading a file, uploading a file.

```
<ProgressBar android:id="@+id/pBar"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:max="100" android:progress="50" />

public class MainActivity extends AppCompatActivity {
      private ProgressBar pgsBar;
      private int i = 0;
      private TextView txtView;
      private Handler hdlr = new Handler();
      @Override protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
```

```
            pgsBar = (ProgressBar) findViewById(R.id.pBar);
            txtView = (TextView) findViewById(R.id.tView);
            Button btn = (Button)findViewById(R.id.btnShow);
            btn.setOnClickListener(new View.OnClickListener() {
            @Override public void onClick(View v) {
                  i = pgsBar.getProgress();
                        new Thread(new Runnable() {
                              public void run() {
                              while (i < 100) {
                                    i += 1;
                                    view hdlr.post(new Runnable() {
                                    public void run() {
                                    pgsBar.setProgress(i);
                                    txtView.setText(i+"/"+pgsBar.getMax());
                                    }
                                    });
                                          try {
                                            Thread.sleep(100);
                                          } catch (InterruptedException e)
                                          {
                                                e.printStackTrace();
                                          }
                                    }
                              }
                        }).start();
                  }
            });
      }
}
```

Android RatingBar

RatingBar is a UI control that is used to get the rating from the user. The RatingBar is an extension of SeekBar and ProgressBar that shows a rating in stars and it allows users to set the rating value by touch or click on the stars.

```
public class MainActivity extends AppCompatActivity {
      private RatingBar rBar;
      private TextView tView;
      private Button btn;
      @Override protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            rBar = (RatingBar) findViewById(R.id.ratingBar1);
            tView = (TextView) findViewById(R.id.textview1);
            btn = (Button)findViewById(R.id.btnGet);
            btn.setOnClickListener(new View.OnClickListener() {
            @Override public void onClick(View v) {
                  int noofstars = rBar.getNumStars();
                  float getrating = rBar.getRating();
                  tView.setText("Rating: "+getrating+"/"+noofstars);
                  }
            });
      }
}
```

Android Spinner

Spinner is a view that allows a user to select one value from the list of values. The spinner in
android will behave same as a dropdown list in other programming languages.

```
<Spinner android:id="@+id/spinner1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>

String[] users = { "Test 1", "Test 2", "Test 3" };
Spinner spin = (Spinner) findViewById(R.id.spinner1);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item,
users);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_it
em);
spin.setAdapter(adapter);
```

Android ListView

Android **ListView** is a view which contains the group of items and displays in a scrollable list.
ListView is implemented by importing android.widget.ListView class.
ListView is a default scrollable which does not use other scroll view.

```
listView=(ListView)findViewById(R.id.listView);
textView=(TextView)findViewById(R.id.textView);
listItem = getResources().getStringArray(R.array.array_technology);
final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, android.R.id.text1, listItem);
listView.setAdapter(adapter);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
      public void onItemClick(AdapterView<?> adapterView, View view, int
      position, long l) {
            String value=adapter.getItem(position);
            Toast.makeText(getApplicationContext(),value,Toast.LENGTH_SHORT
            ).show();
      }
});
```

Android Intent

An **intent** is an abstract description of an operation to be performed. It can be used with startActivity to launch an Activity, broadcastIntent to send it to any interested BroadcastReceiver components, and Context.startService(Intent) or Context.bindService(Intent, ServiceConnection, int) to communicate with a background Service.

Explicit intent specifies the component to be invoked from activity. In other words, we can call another activity in android by explicit intent.

```
 Intent i = new Intent(getApplicationContext(), SecondActivity.class);
 i.putExtra("Value1", "Android1");
 i.putExtra("Value2", "Simple Test");
 startActivity(i);
```

Implicit Intent doesn't specifiy the component. In such case, intent provides information of available components provided by the system that is to be invoked.

```
Intent intent=new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("https://www.google.com"));
startActivity(intent);
```