**4. Android Popup Menu with Examples**

In android, **Popup Menu** displays a list of items in a modal popup window that is anchored to the [view](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples). The popup menu will appear below the [view](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) if there is a room or above the view in case if there is no space and it will be closed automatically when we touch outside of the popup.

The android Popup Menu provides an overflow style menu for actions that are related to specific content.

In android, the Popup Menu provides an overflow of actions that are related to specific content and the actions in popup menu won’t affect the corresponding content.  The popup menu won’t support any item shortcuts and item icons.

In android, Popup menu is available with API level 11 (Android 3.0) and higher versions. If you are using Android 3.0 +, the Popup Menu won’t support any item shortcuts and item icons in the menu.

**Create Android Popup Menu in XML File**

In android, to define **popup menu**, we need to create a new folder **menu** inside of our project resource directory (**res/menu/**) and add a new XML (menu\_example) file to build the menu.

Following is the example of defining a menu in XML file (**menu\_example.xml**).

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <menu xmlns:android="http://schemas.android.com/apk/res/android">     <item android:id="@+id/mail"         android:icon="@drawable/ic\_mail"         android:title="@string/mail" />     <item android:id="@+id/upload"         android:icon="@drawable/ic\_upload"         android:title="@string/upload"         android:showAsAction="ifRoom" />     <item android:id="@+id/share"         android:icon="@drawable/ic\_share"         android:title="@string/share" /> </menu> |

Once we are done with the creation of menu, we need to create a [view](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) element which anchored the menu.

|  |
| --- |
| <Button     android:id="@+id/btnShow"     android:layout\_width="wrap\_content"     android:layout\_height="wrap\_content"     android:text="Show Popup Menu"     android:onClick="showPopup" /> |

 Now in our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) we need to implement showPopup method to show the popup menu.

**Load Android Popup Menu from an Activity**

To show the popup menu for the view, we need to instantiate **Popup** constructor and use **MenuInflater** to load the defined menu resource using **MenuInflater.inflate()** like as shown below.

|  |
| --- |
| public void showPopup(View v) {     PopupMenu popup = new PopupMenu(this, v);     MenuInflater inflater = popup.getMenuInflater();     inflater.inflate(R.menu.menu\_example, popup.getMenu());     popup.show(); } |

**Handle Android Popup Menu Click Events**

To perform an action when the user selects a menu item, we need to implement the **PopupMenu.OnMenuItemClickListener** interface and register it with our **PopupMenu** by calling **setOnMenuItemclickListener()**. When the user selects an item, the system calls the **onMenuItemClick()** callback in your interface.

Following is the example of handling a popup menu item click event using **onMenuItemClick()**.

|  |
| --- |
| public void showMenu(View v) {     PopupMenu popup = new PopupMenu(this, v);     popup.setOnMenuItemClickListener(this);     popup.inflate(R.menu.actions);     popup.show(); }  @Override public boolean onMenuItemClick(MenuItem item) {     switch (item.getItemId()) {         case R.id.archive:             archive(item);             return true;         case R.id.delete:             delete(item);             return true;         default:             return false;     } } |

**Note**: If we are using Android 3.0 +, the Popup Menu won’t support any item shortcuts and item icons in the menu.

**Android Popup Menu Example**

Following is the example of implementing a **Popup Menu** in android application.

Create a new android application using android studio and give names as **PopupMenuExample**. In case if you are not aware of creating an app in android studio check this article [Android Hello World App](https://www.tutlane.com/tutorial/android/android-hello-world-app-example).

Now open an **activity\_main.xml** file from **\res\layout** path and write the code like as shown below

**activity\_main.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"     android:layout\_width="match\_parent"     android:layout\_height="match\_parent"     android:orientation="vertical" >      <Button         android:id="@+id/btnShow"         android:layout\_width="wrap\_content"         android:layout\_height="wrap\_content"         android:text="Show Popup Menu"         android:layout\_marginTop="200dp"  android:layout\_marginLeft="100dp"/> </LinearLayout> |

If we observe above code we created a one [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) control in XML Layout file to show the popup menu when we click on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples).

In android, to define **popup menu**, we need to create a new folder **menu** inside of our project resource directory (**res/menu/**) and add a new XML (**popup\_menu.xml**) file to build the menu.

Now open newly created xml (**popup\_menu.xml**) file and write the code like as shown below.

**popup\_menu.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <menu xmlns:android="http://schemas.android.com/apk/res/android" >     <item android:id="@+id/search\_item"         android:title="Search" />     <item android:id="@+id/upload\_item"         android:title="Upload" />     <item android:id="@+id/copy\_item"         android:title="Copy" />     <item android:id="@+id/print\_item"         android:title="Print" />     <item android:id="@+id/share\_item"         android:title="Share" />     <item android:id="@+id/bookmark\_item"         android:title="BookMark" /> </menu> |

Once we are done with creation of **menu**, we need to load this menu XML resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) by instantiating a **Popup** constructor, for that open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** from **\java\com.sarker.popupmenuexample** path and write the code like as shown below.

**MainActivity.java**

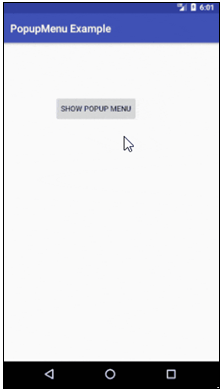
|  |
| --- |
| package com.sarker.popupmenuexample; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.MenuItem; import android.view.View; import android.widget.Button; import android.widget.PopupMenu; import android.widget.Toast;  public class MainActivity extends AppCompatActivity implements PopupMenu.OnMenuItemClickListener {     @Override     protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         setContentView(R.layout.activity\_main);         Button btn = (Button) findViewById(R.id.btnShow);         btn.setOnClickListener(new View.OnClickListener() {             @Override             public void onClick(View v) {                 PopupMenu popup = new PopupMenu(MainActivity.this, v);                 popup.setOnMenuItemClickListener(MainActivity.this);                 popup.inflate(R.menu.popup\_menu);                 popup.show();             }         });     }     @Override     public boolean onMenuItemClick(MenuItem item) {         Toast.makeText(this, "Selected Item: " +item.getTitle(), Toast.LENGTH\_SHORT).show();         switch (item.getItemId()) {             case R.id.search\_item:                 // do your code                 return true;             case R.id.upload\_item:                 // do your code                 return true;             case R.id.copy\_item:                 // do your code                 return true;             case R.id.print\_item:                 // do your code                 return true;             case R.id.share\_item:                 // do your code                 return true;             case R.id.bookmark\_item:                 // do your code                 return true;             default:                 return false;         }     } } |

If we observe above code we are trying to show popup menu on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click, loaded defined menu resource using **Popup.inflate()** and implement popup menu items click event.

Generally, during the launch of our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle), **onCreate()** callback method will be called by android framework to get the required layout for an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle).

**Output of Android Popup Menu Example**

When we run above example using android virtual device (AVD) we will get a result like as shown below.



This is how we can create **Popup Menu** in android applications to handle global actions.

**Questions**

1. Write the **showPopup()** method code.
2. Which method is used to handle item click event for popup menu (**onMenuItemClick())?**