**3. Android Context Menu with Examples**

In android, **Context Menu** is like a floating menu and that appears when the user performs a long press or click on an element and it is useful to implement an actions that effect the selected content or context frame.

In android, the Context Menu offers an actions that effect a specific item or context frame in the UI and we can provide a context menu for any [view](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples). The context menu won’t support any item shortcuts and item icons.

**Create Android Context Menu in Activity**

The [views](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) which we used to show the context menu on long press, we need to register that [views](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) using **registerForContextMenu(View)** in our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) and we need to override **onCreateContextMenu()** in our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) or [fragment](https://www.tutlane.com/tutorial/android/android-fragments-with-examples).

When the registered [view](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) receives a long click event, the system calls our **onCreateContextMenu()** method. By using **onCreateContextMenu()** method, we can create our menu items like as shown below.

|  |
| --- |
| @Override protected void onCreate(Bundle savedInstanceState) {     super.onCreate(savedInstanceState);     setContentView(R.layout.activity\_main);     Button btn = (Button) findViewById(R.id.btnShow);     registerForContextMenu(btn); } @Override public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {     super.onCreateContextMenu(menu, v, menuInfo);     menu.setHeaderTitle("Context Menu");     menu.add(0, v.getId(), 0, "Upload");     menu.add(0, v.getId(), 0, "Search");    } |

If we observe above code, we registered our [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) control using **registerForContextMenu()** to show the context menu on button long-click and binding the Context Menu items using **onCreateContextMenu()** method.

**Handle Android Context Menu Click Events**

In android, we can handle a context menu item click events using **onContextItemSelected()** method.

Following is the example of handling a context menu item click event using **onContextItemSelected()** method.

|  |
| --- |
| @Override public boolean onContextItemSelected(MenuItem item) {     if (item.getTitle() == "Save") {         // do your coding     }     else {         return  false;     }     return true; } |

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

**Android Context Menu Example**

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

Create a new android application using android studio and give names as **ContextMenuExample**.

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="Long press me"         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 context menu when we do long press on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples).

Once we are done with creation of layout with required control, we need to load the XML layout resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) **onCreate()** callback method, for that open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** from **\java\com.sarker.contextmenuexample** path and write the code like as shown below.

**MainActivity.java**

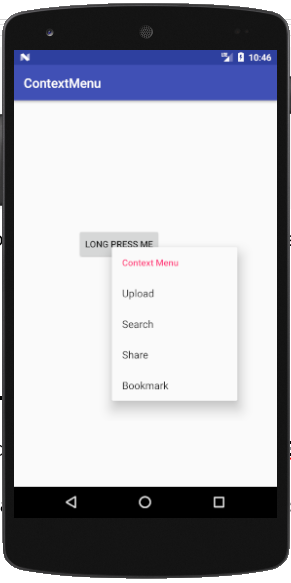
|  |
| --- |
| package com.sarker.contextmenuexample; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.ContextMenu; import android.view.MenuItem; import android.view.View; import android.widget.Button; import android.widget.Toast;  public class MainActivity extends AppCompatActivity {     @Override     protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         setContentView(R.layout.activity\_main);         Button btn = (Button) findViewById(R.id.btnShow);         registerForContextMenu(btn);     }     @Override     public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {         super.onCreateContextMenu(menu, v, menuInfo);         menu.setHeaderTitle("Context Menu");         menu.add(0, v.getId(), 0, "Upload");         menu.add(0, v.getId(), 0, "Search");         menu.add(0, v.getId(), 0, "Share");         menu.add(0, v.getId(), 0, "Bookmark");     }     @Override     public boolean onContextItemSelected(MenuItem item) {         Toast.makeText(this, "Selected Item: " +item.getTitle(), Toast.LENGTH\_SHORT).show();         return true;     } } |

If we observe above code we are overriding **onCreateContextMenu()** method in activity to create context menu and registered view for context menu using **registerForContextMenu()**.

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 Context 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 **Context Menu** in android applications to show the menu list when the user long press on defined element in our application.

**Questions**

1. How to use **registerForContextMenu()** method for registering view for context menu?
2. Which method is used to handle item click event for context menu (**onContextItemSelected())?**.
3. [Difference between context menu and option menu in android](https://stackoverflow.com/questions/1620191/difference-between-context-menu-and-option-menu-in-android).

Options Menu - the menu you see when pressing the 'Menu' button  
Context Menu - the menu shown when you press and hold an item.