**Learn About Android Start New Activity Example**

**Here we will learn the simplest way to start a new activity in android. We can create and start a new android activity from any other activities. Also we can start a new android activity from any other place (like from Android Services, Button click etc). To make the example very clear, here we will create a new activity and start the activity from another activity’s button click event. We can also find the complete code description here.**

**Android Code Snippet to Start a new Activity**

Intent intent=new Intent(getApplicationContext(),TestActivity.class);

startActivity(intent);

**That’s it. In the above code snippet Android Activity Manager will start your requested Activity. If your Activity code if ready, then you can put the above code snippet in your code from where you want to start a Activity. Please make sure to replace the Activity name (TestActivity) with your Activity name from the above code snippet. Also make sure to add your Activity’s name in your app’s Manifest file. If you are still facing any issue then don’t worry below you can download complete code with description.**

**Android Start New Activity Example Details**

**In the Application we will create 2 Android Activities. One will have a simple button from where we will start the new Activity. For an Android Activity, we need two parts (One is the xml layout file and other is the Activity code logic in the java file). So let’s start.**

**1. Layout xml File (activity\_main.xml) for MainActivity**

**This xml layout file will be used for showing GUI for our MainActivity, So we have added below GUI options in this xml layout file.**

**1. One a Android TextView (for showing simple information). The text view is optional.**

**2. One Android Button (we will start the new activity from this button click option)**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true"

android:text="@string/hello\_world" />

<Button

android:id="@+id/buttonStartActivity"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/textView"

android:text="Start a New Activity"

android:onClick="newActivityOnClik"/>

</RelativeLayout>

**2. Layout xml File (activity\_test.xml) for New Activity**

**This simple xml layout file will be used for showing GUI for our new Activity, which will be start from the main Activity. Here we have added one simple TextView to show some information on the new Activity.**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true"

android:text="Hurry.This is your New Activity" />

</RelativeLayout>

**3. MainActivity.Java File**

**This activity is the main activity from where we will start the new activity.**

package com.mm214.activityexample;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.Menu;

import android.view.View;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

//this function will be called when you click on the button

//This OnClick option has been set in the activity\_main.xlm file

public void newActivityOnClik(View V){

// Create an Intent, which will be use for TestActivity and Start the Activity

//Below //TestActivity.Class is the activity Class, which we will start from here.

Intent intent=new Intent(getApplicationContext(),TestActivity.class);

startActivity(intent);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

}

**4. TestActivity.Java**

**This simple java file contains the code for your new Activity, which we are going to launch from the MainActivity.**

package com.techblogon.activityexample;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

public class TestActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_test);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

}

**5. Manifest File**

We need to add each android Activity name in this file, which we are using in our application. So Below is the Manifest file in which we have added (both MainActivity and TestActivity ) names.

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.mm214.activityexample"

android:versionCode="1"

android:versionName="1.0" >

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="17" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name">

<activity

android:name="com.mm214.activityexample.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name="com.mm214.activityexample.TestActivity"

android:label="@string/app\_name" >

</activity>

</application>

</manifest>

**Note: If you will forget to add your Activity name in the Manifest file, then Android will not find it in the package and will show a Force Close Error, when you will run the application.**