## **Lab 14: Android Clipboard**

# **Introduction**

For copying and pasting various kinds of data, including text strings, images, binary stream data, and other complex data types, Android's Clipboard framework is helpful. The simple text data is typically stored immediately in the clipboard by the android clipboard framework, while the complex data is typically stored as a reference that the pasting application resolves with a content provider. The clipboard copying and pasting functionality in Android is available both within and between apps that use the framework.

**Let’s get Started:**

In this exercise, we will learn how to use clipboards in our Android application using Kotlin as our programming language.

**Step 1:** open **activity\_main.xml** file from **\res\layout** folder path and write the code like as shown below.

**activity\_main.xml**

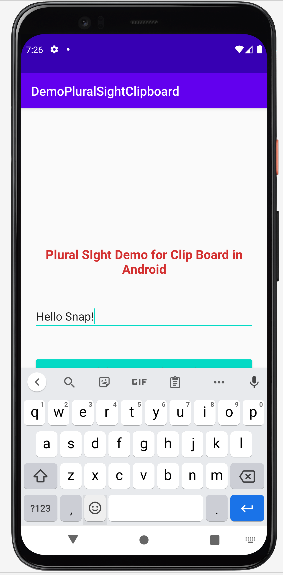
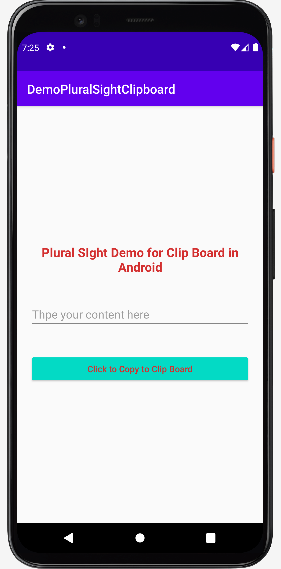
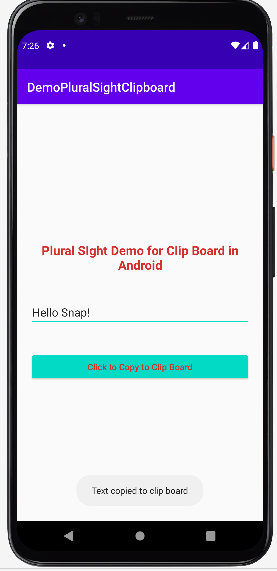
*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 *<!-- creating text view for displaying heading-->* <TextView  
 android:id="@+id/idTVHeading"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_above="@id/idEdtMsg"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:gravity="center"  
 android:text="Plural Sight Demo for Clip Board in Android"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textColor="#D32F2F"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
<EditText  
 android:id="@+id/idEdtMsg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:layout\_margin="20dp"  
 android:hint="Thpe your content here" />  
<Button  
 android:id="@+id/idBtnCopyClipBoard"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/idEdtMsg"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Click to Copy to Clip Board"  
 android:textAllCaps="false"  
 android:backgroundTint="@color/teal\_200"  
 android:textColor="#D32F2F" />  
</RelativeLayout>

**Step 2:** Open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.kt** and write the code like as shown below.

**MainActivity.kt**

package com.example.demopluralsightclipboard  
  
import android.content.ClipData  
import android.content.ClipboardManager  
import android.os.Bundle  
import android.support.v7.app.AppCompatActivity  
import android.widget.\*  
class MainActivity : AppCompatActivity() {  
 lateinit var copyClipBoardBtn: Button  
 lateinit var msgEdt: EditText  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 copyClipBoardBtn = findViewById(R.id.*idBtnCopyClipBoard*)  
 msgEdt = findViewById(R.id.*idEdtMsg*)  
 copyClipBoardBtn.setOnClickListener {  
 val clipboardManager = getSystemService(*CLIPBOARD\_SERVICE*) as ClipboardManager  
 val message = msgEdt.*text*.toString()  
 val clipData = ClipData.newPlainText("text", message)  
 clipboardManager.setPrimaryClip(clipData)  
 Toast.makeText(*applicationContext*, "Text copied to clip board", Toast.*LENGTH\_SHORT*)  
 .show()  
 }  
 }  
}

**Step 3: Check Output on Android Emulator and it should look like as given below.**

** **

**Voila!!** We have successfully completed this lab.