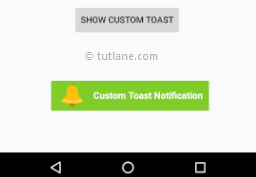
**3. Android Custom Toast with Examples**

In android, [**Toast**](https://www.tutlane.com/tutorial/android/android-toast-with-examples) is a small popup notification which is used to display an information about the operation which we performed in our app. The Toast will show the message for a small period of time and it will disappear automatically after a timeout.

Generally, the size of Toast will be adjusted based on the space required for the message and it will be displayed on the top of main content of an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) for a short period of time.

Generally, the Toast notification in android will be displayed with simple text like as shown in above image. In android, we can customize the layout of our toast notification to change the appearance of based on requirements like include images in toast notification or change the background colour of toast notification, etc.

Following is the pictorial representation of using **Custom Toast** notification in android applications.



To customize the appearance of Toast notification, we need to create a custom layout in our XML or application code and pass the root View object to the **setView(View)** method.

**Create a Custom Toast in Android**

To create a custom Toast notification in android, we need to define a custom [View](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) layout in XML, for that create a custom XML file (**custom\_toast.xml**) in layout (**/layout**) folder and write the code like as shown below.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/custom\_toast\_container"  
    android:orientation="horizontal"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:paddingLeft="10dp"  
    android:paddingRight="10dp"  
    android:background="#80CC28">  
    <ImageView android:src="@drawable/ic\_notification"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginRight="10dp" />  
    <TextView android:id="@+id/txtvw"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="13dp"  
        android:textColor="#FFF"  
        android:textStyle="bold"  
        android:textSize="15dp" />  
</LinearLayout>

To use this custom layout as [Toast](https://www.tutlane.com/tutorial/android/android-toast-with-examples) notification in our android application, we need to inflate the layout using following code.

LayoutInflater inflater = getLayoutInflater();  
View layout = inflater.inflate(R.layout.custom\_toast, (ViewGroup) findViewById(R.id.custom\_toast\_layout));  
TextView tv = (TextView) layout.findViewById(R.id.txtvw);  
tv.setText("Custom Toast Notification");  
Toast toast = new Toast(getApplicationContext());  
toast.setDuration(Toast.LENGTH\_LONG);  
toast.setView(layout);  
toast.show();

If we observe above code, we created an instance of **LayoutInflater** with **getLayoutInflater()**, and then inflate our XML layout using **inflate(int, ViewGroup)**. Here the first parameter is the **layout resource ID** and the second is the **root View**and this inflated layout will help us to find the [View](https://www.tutlane.com/tutorial/android/android-view-and-viewgroup-with-examples) objects in the layout. After that we created a new [Toast](https://www.tutlane.com/tutorial/android/android-toast-with-examples) with **Toast(Context)** and set required properties of the [toast](https://www.tutlane.com/tutorial/android/android-toast-with-examples), then we call **setView(View)** and pass it to the inflated layout.

Once we are done with required configurations, then we can show the custom toast notification by calling **show()**method.

Now we will see how to implement a custom Toast  [notification](https://www.tutlane.com/tutorial/android/android-notifications-bigtextstyle-bigpicturestyle-inboxstyle" \o "Android Notifications with Examples" \t "_blank) in android applications with examples.

**Android Custom Toast Example**

Create a new android application using android studio and give names as **ToastExample**. 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 Custom Toast"  
        android:layout\_marginTop="150dp" android:layout\_marginLeft="110dp"/>  
</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 custom toast [notification](https://www.tutlane.com/tutorial/android/android-notifications-bigtextstyle-bigpicturestyle-inboxstyle) when we click on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples).

Now we need to create a custom layout for our toast notification, for that create a new xml file (**custom\_toast.xml**) in **/layout** folder and write the code like as shown below.

**Custom\_toast.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/custom\_toast\_layout"  
    android:orientation="horizontal"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:paddingLeft="10dp"  
    android:paddingRight="10dp"  
    android:background="#80CC28">  
    <ImageView android:src="@drawable/ic\_notification"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginRight="10dp" />  
    <TextView android:id="@+id/txtvw"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="13dp"  
        android:textColor="#FFF"  
        android:textStyle="bold"  
        android:textSize="15dp" />  
</LinearLayout>

If we observe above code we are loading image (**ic\_notification**) from **drawable** folder so we need to add your icon in **drawable** folder to show it in notification.

Once we are done with creation of layout with required controls, 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 file **MainActivity.java** from **\java\com.tutlane.toastexample** path and write the code like as shown below.

**MainActivity.java**

package com.tutlane.customtoastexample;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
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);  
        btn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                LayoutInflater inflater = getLayoutInflater();  
                View layout = inflater.inflate(R.layout.custom\_toast, (ViewGroup) findViewById(R.id.custom\_toast\_layout));  
                TextView tv = (TextView) layout.findViewById(R.id.txtvw);  
                tv.setText("Custom Toast Notification");  
                Toast toast = new Toast(getApplicationContext());  
                toast.setGravity(Gravity.CENTER\_VERTICAL, 0, 100);  
                toast.setDuration(Toast.LENGTH\_LONG);  
                toast.setView(layout);  
                toast.show();  
            }  
        });  
    }  
}

If we observe above code we are calling our custom toast notification by using **LayoutInflater** object and showing a toast notification on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click.

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 Custom Toast Example**

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



If we observe above result we created a custom toast notification and showing it on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click based on our requirements.

This is how we can create custom toast notifications in android applications based on our requirements.