1. **Android Notifications**

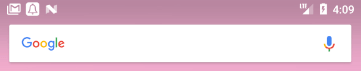
(BigTextStyle, BigPictureStyle, InboxStyle)

In android, **Notification** is a message which is used to alert the users about some events that happening in our app.

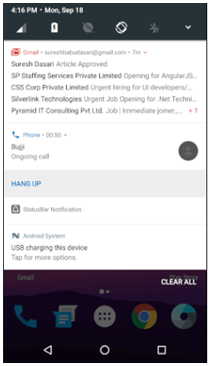
Generally, the android Notifications will be displayed outside of our app’s normal UI and alert the users without interrupting their current activities.

In android, we can alert the users about our app notifications in different forms like flash the LED or make a sounds or display an icon in status bar, etc.

When we tell the system to issue a notification, first it will display an icon in notification bar like as shown below.



To see the details of our android app notification, we need to open the notification drawer like as shown below.



Now we will see how to create and issue a notifications in android applications with examples.

Here we are going to use **NotificationCompat** class to implement a notification in our android application. The **NotificationCompat** class supports a different type of notification views, such as normal view, big view and it provides a best support for a wide range of platforms.

## **Create a Notification in Android**

To create a notification, we need to specify the UI content and required actions with a **NotificationCompat.Builder**object. To display an icon, title and detailed text of notification we need to set following properties in Builder object.

* **setSmallIcon()** - It is used to set the small icon for our notification.
* **setContentTitle()** - It is used to set the title of our notification.
* **setContentText()** - It is used to set the detailed text to display in notification.

The above mentioned properties are necessary to display a notification and we can set a different type of properties to our notification like setStyle, setSound, setLights, setLargeIcon, etc. based on our requirements using Builder object.

Following is the example of creating a notification using **NotificationCompat.Build** object and setting the notification properties.

NotificationCompat.Builder nBuilder =  new NotificationCompat.Builder(this)  
                .setSmallIcon(R.drawable.notification\_icon)  
                .setContentTitle("Sample notification")  
                .setContentText("Hi, Welcome to Tutlane.com");

## **Define the Android Notification Actions**

If we assign an action to the notification, it will allow users to go directly from the notification to an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) of our app. We can also add buttons to the notification to perform an additional actions such as hang up the call or responding immediately to a text message; this feature is available as of Android 4.1.

In android, we can define an action inside of notification by using **PendingIntent** object which contains an [Intent](https://www.tutlane.com/tutorial/android/android-intents-implicit-explicit) that starts an [Activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) of our app.

Following is the example of defining an action inside of notification using **PendingIntent** object.

NotificationCompat.Builder nBuilder =  new NotificationCompat.Builder(this)  
…...  
Intent resultIntent = new Intent(this, MainActivity.class);  
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, resultIntent, 0);  
nBuilder.setContentIntent(pendingIntent);

## **Issue the Android Notification**

Once we are done with creation of notification, we need to pass a notification to the system by using **NotificationManager.notify()** method and we need to specify a **ID** in the notification to use this **ID** to update a notification later if required.

Following is the example of sending a notification to the system using **Notificationmanager.notify** method.

NotificationCompat.Builder nBuilder =  new NotificationCompat.Builder(this);  
....  
int mNotificationId = 999;  
NotificationManager mNotifyMgr = (NotificationManager)getSystemService(NOTIFICATION\_SERVICE);  
// Builds the notification and issues it.  
mNotifyMgr.notify(mNotificationId, nBuilder.build());

Now we will see how to create and show the notification in android application with example.

## **Android Notification Example**

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

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

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 Notification"  
        android:layout\_marginTop="200dp"

android:layout\_marginLeft="100dp"/>  
</LinearLayout>

If you 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 notification in notification bar when we click on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples).

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.notificationexample** path and write the code like as shown below.

## **MainActivity.java**

package com.tutlane.notificationexample;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.support.v4.app.NotificationCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btnNotify = (Button)findViewById(R.id.btnShow);  
        btnNotify.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                NotificationCompat.Builder mBuilder =  new NotificationCompat.Builder(MainActivity.this)  
                                .setSmallIcon(R.drawable.ic\_notification)  
                                .setContentTitle("Tutlane Send New Message")  
                                .setContentText("Hi, Welcome to tutlane tutorial site");  
                // Set the intent to fire when the user taps on notification.  
                Intent resultIntent = new Intent(MainActivity.this, MainActivity.class);  
                PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, resultIntent, 0);  
                mBuilder.setContentIntent(pendingIntent);  
                // Sets an ID for the notification  
                int mNotificationId = 001;  
                NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);  
                // It will display the notification in notification bar  
                notificationManager.notify(mNotificationId, mBuilder.build());  
            }  
        });  
    }  
}

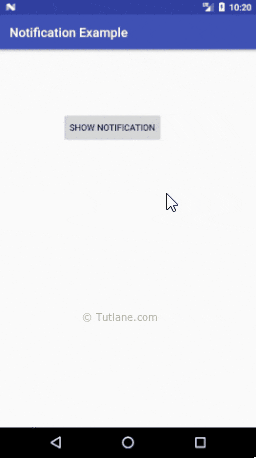
If we observe above code we are creating a notification, adding action inside of notification using [intent](https://www.tutlane.com/tutorial/android/android-intents-implicit-explicit) object to open the [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) in our app and showing notification on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click using **NotificationManager**.

Here we added an **ic\_notification** image in **drawable** folder to show it as a notification icon so please add a required image in your **drawable** folder and use it in your application.

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 Notification 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 notification and shown it on Button click using NotificationCompat.Builder based on our requirements.

## **Android Big Text Style Notification Example**

We learned how to show the android notification in normal view, in case if we want to show the large icon and large text in notification, then that can be achieved using **NotificationCompat.BigTextStyle**.

We need to modify our main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** like as shown below.

## **MainActivity.java**

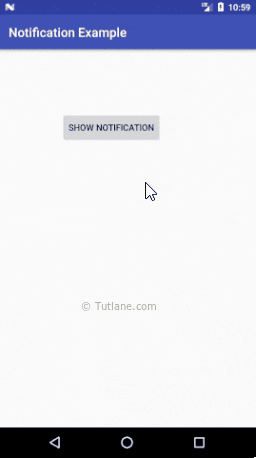
package com.tutlane.notificationexample;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.support.v4.app.NotificationCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btnNotify = (Button)findViewById(R.id.btnShow);  
        btnNotify.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                //To set large icon in notification  
                Bitmap licon = BitmapFactory.decodeResource(getResources(), R.drawable.notification\_icon);  
                //Assign BigText style notification  
                NotificationCompat.BigTextStyle bigText = new NotificationCompat.BigTextStyle();  
                bigText.bigText("Welcome to tutlane, it provides a tutorials related to all technologies in software industry. Here we covered complete tutorials from basic to adavanced topics from all technologies");  
                bigText.setSummaryText("By: Tutlane");  
  
                NotificationCompat.Builder mBuilder =  new NotificationCompat.Builder(MainActivity.this)  
                                .setSmallIcon(R.drawable.ic\_notification)  
                                .setContentTitle("Big Text Notification Example")  
                                .setLargeIcon(licon)  
                                .setStyle(bigText);  
                // Set the intent to fire when the user taps on notification.  
                Intent resultIntent = new Intent(MainActivity.this, MainActivity.class);  
                PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, resultIntent, 0);  
                mBuilder.setContentIntent(pendingIntent);  
                // Sets an ID for the notification  
                int mNotificationId = 001;  
                NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);  
                // It will display the notification in notification bar  
                notificationManager.notify(mNotificationId, mBuilder.build());  
            }  
        });  
    }  
}

If we observe above code we are creating a big text style notification using **NotificationCompat.BigTextStyle** and appended to notification using **setStyle()** property.

Here we added a **ic\_notification**, **notification\_icon** (**bmp**) images in **drawable** folder to show it in notification icon so please add a required images in your **drawable** folder and use it in your application.

## **Output of Android Big Text Style Notification 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 big text style notification with large image and shown it on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click based on our requirements.

## **Android Inbox Style Notification Example**

By using **NotificationCompat.InboxStyle** object we can implement inbox style notification.

We need to modify our main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** like as shown below.

## **MainActivity.java**

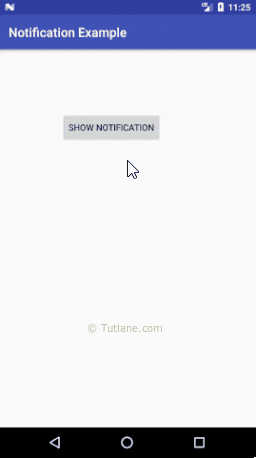
package com.tutlane.notificationexample;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.support.v4.app.NotificationCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btnNotify = (Button)findViewById(R.id.btnShow);  
        btnNotify.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                //Implement inbox style notification  
                NotificationCompat.InboxStyle iStyle =  new NotificationCompat.InboxStyle();  
                iStyle.addLine("Message 1.");  
                iStyle.addLine("Message 2.");  
                iStyle.addLine("Message 3.");  
                iStyle.addLine("Message 4.");  
                iStyle.addLine("Message 5.");  
                iStyle.setSummaryText("+2 more");  
  
                NotificationCompat.Builder mBuilder =  new NotificationCompat.Builder(MainActivity.this)  
                                .setSmallIcon(R.drawable.ic\_notification)  
                                .setContentTitle("Inbox Style Notification Example")  
                                .setStyle(iStyle);  
                // Set the intent to fire when the user taps on notification.  
                Intent resultIntent = new Intent(MainActivity.this, MainActivity.class);  
                PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, resultIntent, 0);  
                mBuilder.setContentIntent(pendingIntent);  
                // Sets an ID for the notification  
                int mNotificationId = 001;  
                NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);  
                // It will display the notification in notification bar  
                notificationManager.notify(mNotificationId, mBuilder.build());  
            }  
        });  
    }  
}

If you observe above code we are creating a inbox style notification using **NotificationCompat.InboxStyle** and appended to notification using **setStyle()** property.

Here we added a **ic\_notification** in **drawable** folder to show it in notification icon so please add a required images in your **drawable** folder and use it in your application.

## **Output of Android Inbox Style Notification 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 an inbox style notification and shown it on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click based on our requirements.

## **Android Big Picture Notification Example**

By using **NotificationCompat.BigPictureStyle** object we can implement inbox style notification.

We need to modify our main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** like as shown below.

## **MainActivity.java**

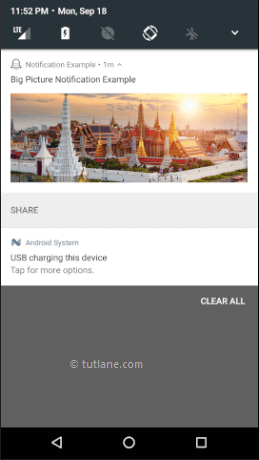
package com.tutlane.notificationexample;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.graphics.BitmapFactory;  
import android.net.Uri;  
import android.support.v4.app.NotificationCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btnNotify = (Button)findViewById(R.id.btnShow);  
        btnNotify.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                // Assign big picture notification  
                NotificationCompat.BigPictureStyle bpStyle = new NotificationCompat.BigPictureStyle();  
                bpStyle.bigPicture(BitmapFactory.decodeResource(getResources(), R.drawable.big\_img)).build();  
                // Set the intent to fire when the user taps on notification.  
                Intent rIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse("http://tutlane.com/"));  
                PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, rIntent, 0);  
                NotificationCompat.Builder mBuilder =  new NotificationCompat.Builder(MainActivity.this)  
                                .setSmallIcon(R.drawable.ic\_notification)  
                                .setContentTitle("Big Picture Notification Example")  
                                .addAction(R.drawable.ic\_share, "Share", pendingIntent)  
                                .setStyle(bpStyle);  
  
                mBuilder.setContentIntent(pendingIntent);  
                // Sets an ID for the notification  
                int mNotificationId = 001;  
                NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);  
                // It will display the notification in notification bar  
                notificationManager.notify(mNotificationId, mBuilder.build());  
            }  
        });  
    }  
}

If we observe above code we are creating a inbox style notification using **NotificationCompat.BigPicture** and appended to notification using **setStyle()** property.

Here we added a **ic\_share, big\_img** images in **drawable** folder to show it in notification so please add a required images in your **drawable** folder and use it in your application.

## **Output of Android Big Picture Style Notification 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 big picture style notification and shown it on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click based on our requirements.

This is how we can create and show the notifications in android applications based on our requirements.