

Android Notification

Android Notification provides short, timely information about the action happened in the application, even it is not running. The notification displays the icon, title and some amount of the content text.

Set Android Notification Properties

The properties of Android notification are set using **NotificationCompat.Builder** object. Some of the notification properties are mention below:

- **setSmallIcon()**: It sets the icon of notification.
- **setContentTitle()**: It is used to set the title of notification.
- **setContentText()**: It is used to set the text message.
- **setAutoCancel()**: It sets the cancelable property of notification.
- **setPriority()**: It sets the priority of notification.

Android Notification Example

In this example, we will create a notification message which will launch another activity after clicking on it.

activity_main.xml

Add the following code in an activity_main.xml file.

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"`
3. `xmlns:app="http://schemas.android.com/apk/res-auto"`
4. `xmlns:tools="http://schemas.android.com/tools"`
5. `android:layout_width="match_parent"`
6. `android:layout_height="match_parent"`
7. `tools:context="example.javatpoint.com.androidnotification.MainActivity">`
- 8.
9. `<TextView`
10. `android:layout_width="wrap_content"`
11. `android:layout_height="wrap_content"`
12. `android:text="ANDROID NOTIFICATION"`
13. `app:layout_constraintBottom_toBottomOf="parent"`
14. `app:layout_constraintLeft_toLeftOf="parent"`

```

15.     app:layout_constraintRight_toRightOf="parent"
16.     app:layout_constraintTop_toTopOf="parent"
17.     app:layout_constraintVertical_bias="0.091"
18.     android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"/>
19.
20. <Button
21.     android:layout_width="wrap_content"
22.     android:layout_height="wrap_content"
23.     android:id="@+id/button"
24.     android:layout_marginBottom="112dp"
25.     android:layout_marginEnd="8dp"
26.     android:layout_marginStart="8dp"
27.     android:text="Notify"
28.     app:layout_constraintBottom_toBottomOf="parent"
29.     app:layout_constraintEnd_toEndOf="parent"
30.     app:layout_constraintStart_toStartOf="parent" />
31.
32. </android.support.constraint.ConstraintLayout>

```

Create an activity named as **activity_notification_view.xml** and add the following code. This activity will be launched on clicking the notification. TextView is used to display the notification message.

activity_notification_view.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com
   /apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent"
7.     tools:context="example.javatpoint.com.androidnotification.NotificationView">
8.
9.     <TextView
10.         android:id="@+id/textView2"
11.         android:layout_width="fill_parent"
12.         android:layout_height="wrap_content"
13.         android:gravity="center"
14.         android:text="your detail of notification..."

```

```

15.     android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />
16.
17. <TextView
18.     android:id="@+id/textView"
19.     android:layout_width="wrap_content"
20.     android:layout_height="wrap_content"
21.     android:layout_marginBottom="8dp"
22.     android:layout_marginEnd="8dp"
23.     android:layout_marginStart="8dp"
24.     android:layout_marginTop="8dp"
25.     app:layout_constraintBottom_toBottomOf="parent"
26.     app:layout_constraintEnd_toEndOf="parent"
27.     app:layout_constraintHorizontal_bias="0.096"
28.     app:layout_constraintStart_toStartOf="parent"
29.     app:layout_constraintTop_toBottomOf="@+id/textView2"
30.     app:layout_constraintVertical_bias="0.206"
31.     android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"/>
32.
33. </android.support.constraint.ConstraintLayout>

```

MainActivity.java

In the **MainActivity.java** class adds the following code. In this class, clicking the button calls the `addNotification()` method where we implement the `NotificationCompat.Builder` object to set the notification properties. The `NotificationManager.notify()` method is used to display the notification. The `Intent` class is used to call another activity (`NotificationView.java`) on taping the notification.

```

1. import android.app.NotificationManager;
2. import android.app.PendingIntent;
3. import android.content.Context;
4. import android.content.Intent;
5. import android.support.v4.app.NotificationCompat;
6. import android.support.v7.app.AppCompatActivity;
7. import android.os.Bundle;
8. import android.view.View;
9. import android.widget.Button;
10.
11. public class MainActivity extends AppCompatActivity {
12.     Button button;

```

```

13.  @Override
14.  protected void onCreate(Bundle savedInstanceState) {
15.      super.onCreate(savedInstanceState);
16.      setContentView(R.layout.activity_main);
17.      button = findViewById(R.id.button);
18.      button.setOnClickListener(new View.OnClickListener() {
19.          @Override
20.          public void onClick(View v) {
21.              addNotification();
22.          }
23.      });
24.  }
25.
26.  private void addNotification() {
27.      NotificationCompat.Builder builder =
28.          new NotificationCompat.Builder(this)
29.              .setSmallIcon(R.drawable.messageicon) //set icon for notification
30.              .setContentTitle("Notifications Example") //set title of notification
31.              .setContentText("This is a notification message")//this is notification message
32.              .setAutoCancel(true) // makes auto cancel of notification
33.              .setPriority(NotificationCompat.PRIORITY_DEFAULT); //set priority of notificatio
n
34.
35.
36.      Intent notificationIntent = new Intent(this, NotificationView.class);
37.      notificationIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
38.      //notification message will get at NotificationView
39.      notificationIntent.putExtra("message", "This is a notification message");
40.
41.      PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, notificationIntent,
42.          PendingIntent.FLAG_UPDATE_CURRENT);
43.      builder.setContentIntent(pendingIntent);
44.
45.      // Add as notification
46.      NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFI
CATION_SERVICE);
47.      manager.notify(0, builder.build());
48.  }

```

49.}

NotificationView.java

The **NotificationView.java** class receives the notification message and is displayed in TextView. This class is invoked while taping the notification.

```
1. import android.support.v7.app.AppCompatActivity;
2. import android.os.Bundle;
3. import android.widget.TextView;
4. import android.widget.Toast;
5.
6. public class NotificationView extends AppCompatActivity {
7.     TextView textView;
8.     @Override
9.     protected void onCreate(Bundle savedInstanceState) {
10.         super.onCreate(savedInstanceState);
11.         setContentView(R.layout.activity_notification_view);
12.         textView = findViewById(R.id.text_view);
13.         //getting the notification message
14.         String message=getIntent().getStringExtra("message");
15.         textView.setText(message);
16.     }
17. }
```

strings.xml

```
1. <resources>
2.     <string name="app_name">AndroidNotification</string>
3.     <string name="notification_activity">NotificationView</string>
4. </resources>
```

AndroidManifest.xml

Add the following code in AndroidManifest.xml file.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.     package="example.javatpoint.com.androidnotification">
4.
```

```
5. <application
6.     android:allowBackup="true"
7.     android:icon="@mipmap/ic_launcher"
8.     android:label="@string/app_name"
9.     android:roundIcon="@mipmap/ic_launcher_round"
10.    android:supportRtl="true"
11.    android:theme="@style/AppTheme">
12.    <activity android:name=".MainActivity">
13.        <intent-filter>
14.            <action android:name="android.intent.action.MAIN" />
15.
16.            <category android:name="android.intent.category.LAUNCHER" />
17.        </intent-filter>
18.    </activity>
19.    <activity android:name=".NotificationView"
20.        android:label="@string/notification_activity"
21.        android:parentActivityName=".MainActivity">
22.        <meta-data
23.            android:name="android.support.PARENT_ACTIVITY"
24.            android:value=".MainActivity"/>
25.    </activity>
26. </application>
27.
28. </manifest>
```

Output:

