

# CMSC436: Programming Handheld Systems

Fall 2017

# User Notification

# Today's Topics

Toast

Notification Area Notifications

# User Notifications

Messages provided to the user outside of the normal UI

# User Notifications

These include messages aimed at  
User feedback

- Toasts

- Dialogs

Event notification

- Notification Area notifications

# Toast

Transitory messages that pop up on the current window

e.g., to inform user that an operation has completed successfully

Automatically fade into & out of view

No user interaction or response

# Creating Toast Notifications

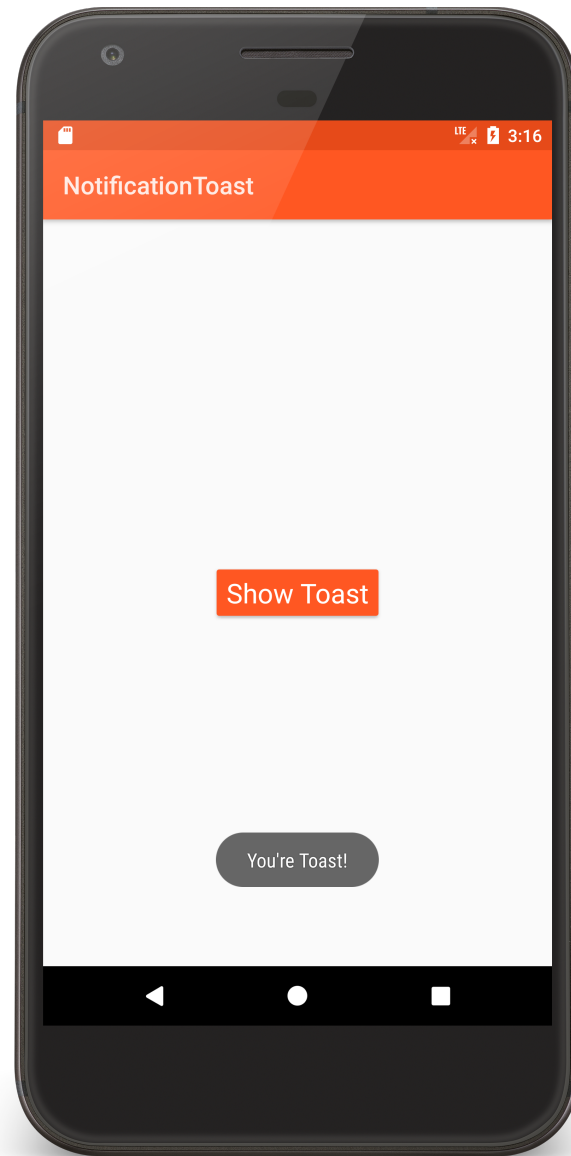
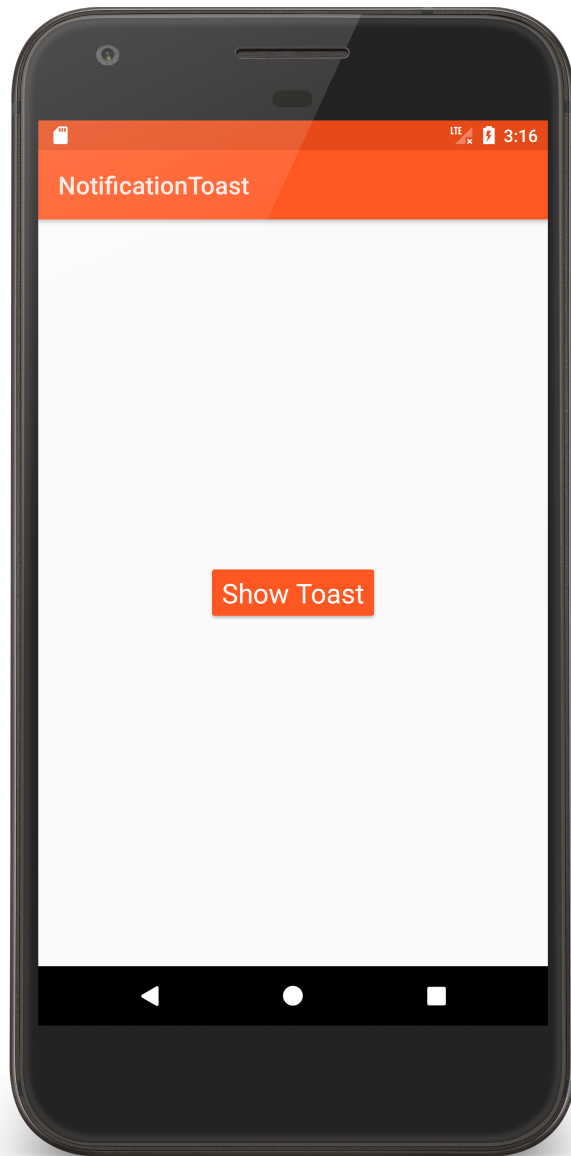
Instantiate a Toast object by calling

```
Toast.makeText(context, text, duration)
```

Show toast by calling

```
Toast.show()
```

# Notification Toast





```
public class NotificationToastActivity extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

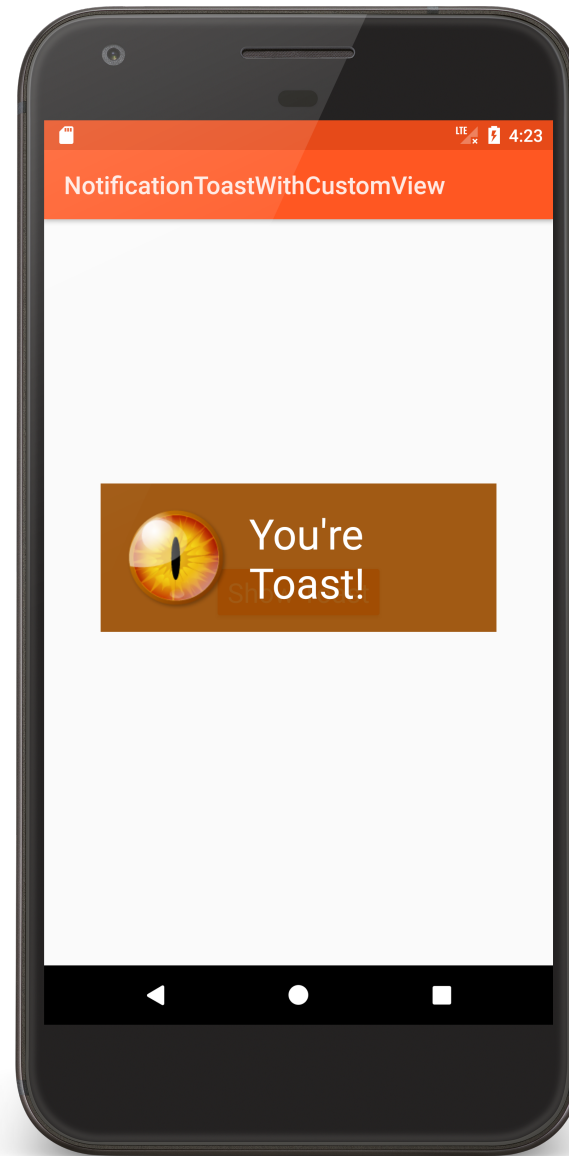
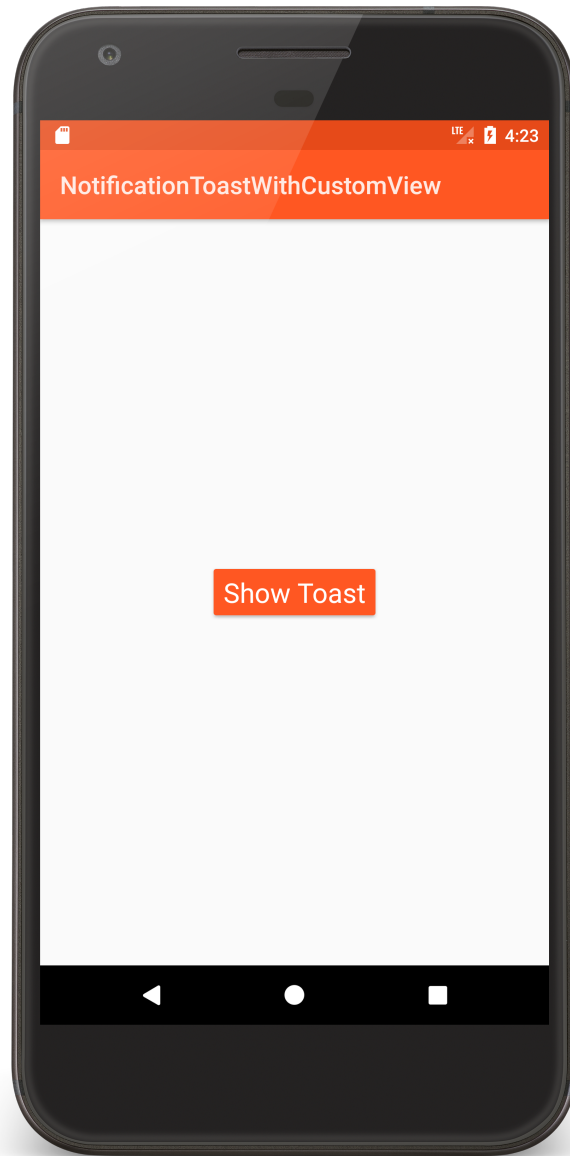
```
    public void onClick(View v) {  
        Toast.makeText(getApplicationContext(),  
            getString(R.string.youre_toast_string),  
            Toast.LENGTH_LONG).show();  
    }  
}
```

# Toast with Custom Views

Can create custom layouts in XML

Set them using `Toast.setView()`

# NotificationToast WithCustomView



```
public class NotificationToastActivity extends Activity {
```

```
...
```

```
public void onClick(View v) {
```

```
    Toast toast = new Toast(getApplicationContext());
```

```
    toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
```

```
    toast.setDuration	Toast.LENGTH_LONG);
```

```
    toast.setView(getLayoutInflater().inflate(R.layout.custom_toast, null));
```

```
    toast.show();
```

```
}
```

```
}
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/toast_layout_root"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="@dimen/activity_margin"
    android:background="@color/accent">
<ImageView android:id="@+id/image"
    android:layout_width="88dp"
    android:layout_height="88dp"
    android:layout_marginRight="@dimen/activity_margin"
    android:src="@drawable/fire_eye_alien"
    android:layout_centerVertical="true"
    android:contentDescription="@string/eyeball_string"/>
```

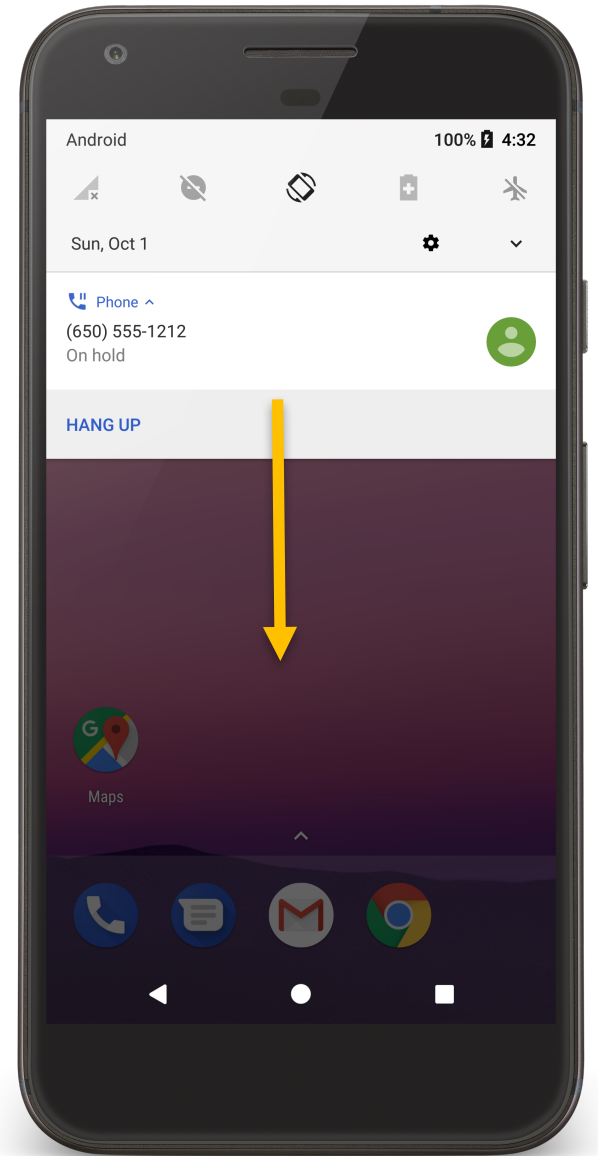
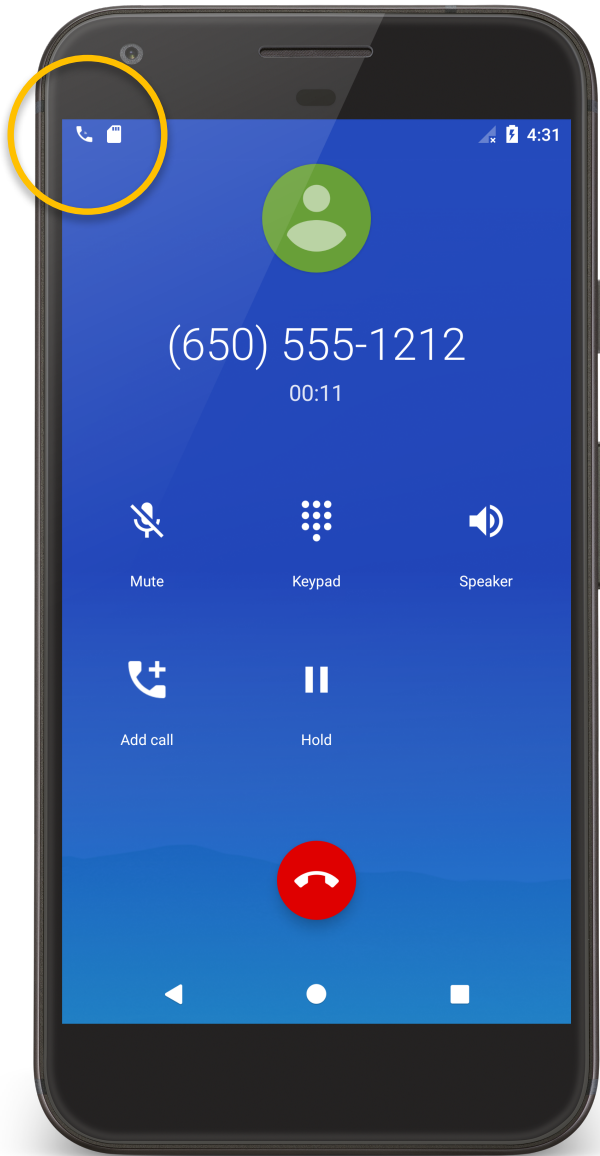
...

```
...  
<TextView android:id="@+id/text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textColor="#FFF"  
    android:text="@string/toast_string"  
    android:layout_centerVertical="true"  
    android:layout_toRightOf="@id/image"  
    android:textAppearance="@android:style/TextAppearance.Material.Display1"  
/>  
</RelativeLayout>
```

# Notification Area Notifications

Android uses the notification area to alert users of important events

Also provides a notification drawer that users can pull down to see more detailed information about notifications





# Notification Manager

System Service that manages Notifications

Can send & cancel notifications

# Notification Architecture

## Notification

Title, detail, small icon

## Notification Area

Ticker text, small icon

## Notification Drawer

View

Action

## Required Notification Contents

A small icon, set by `setSmallIcon()`.

A title, set by `setContentTitle()`.

Detail text, set by `setContentText()`.

A notification channel ID (On API level 26 and higher)

# Notification Channels

Settings associated with each notification channel

Channel characteristics include:

- Importance

- Sound

- Lights

- Vibration

- Show on lockscreen

- Override do not disturb

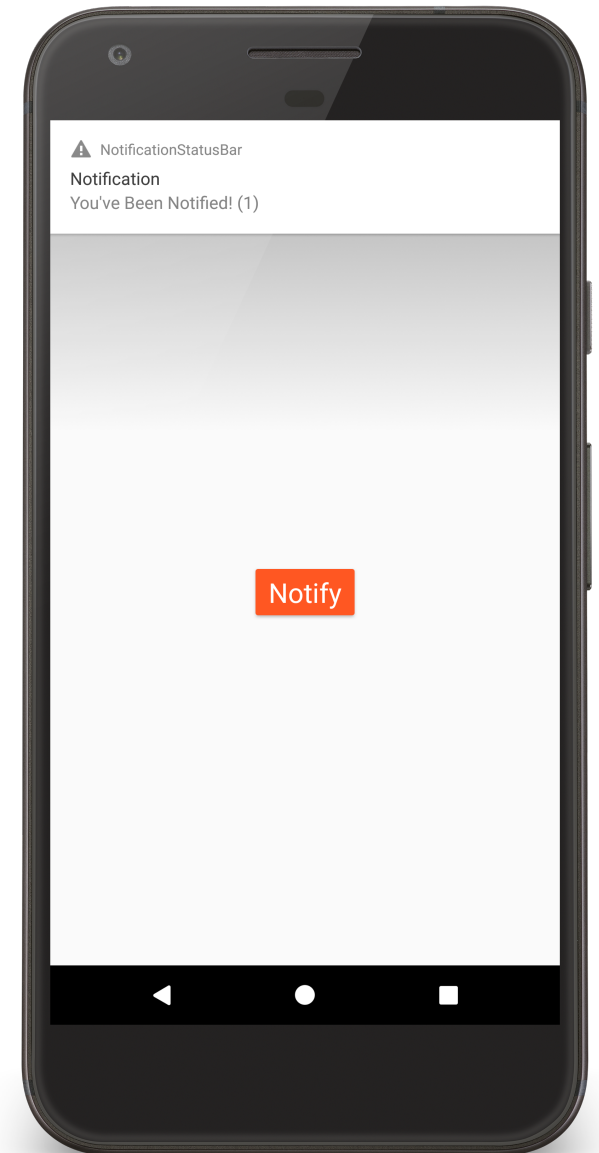
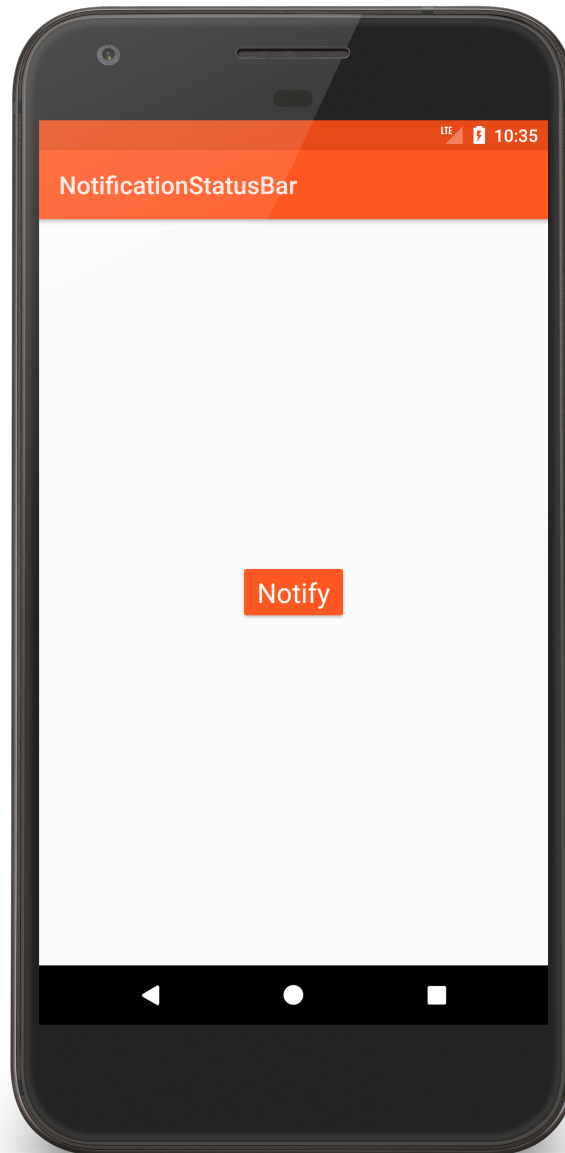
# Creating a Notification Channel

Construct a notification channel object with a package-unique ID

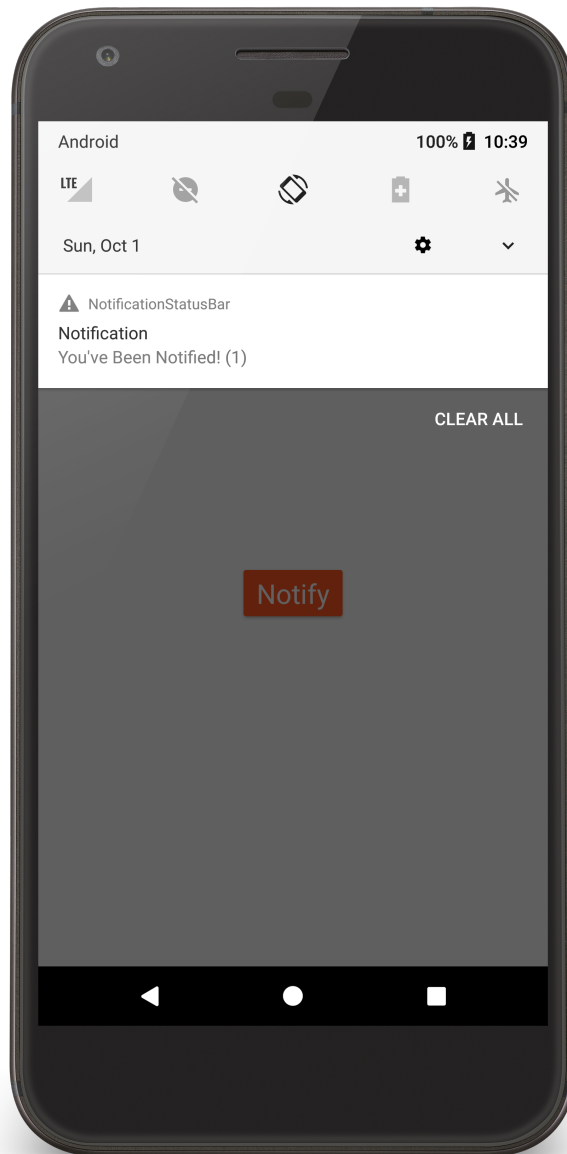
Configure the notification channel object

Submit the notification channel object to the notification manager

# Notification StatusBar



# Notification StatusBar



```
public class NotificationStatusBarActivity extends Activity {  
    // Notification ID to allow for future updates  
    private static final int MY_NOTIFICATION_ID = 1;  
    // Notification Count  
    private int mNotificationCount;  
    // Notification Text Elements  
    private final CharSequence tickerText = "This is ... Long Notification Message!";  
    private final CharSequence contentTitle = "Notification";  
    private final CharSequence contentText = "You've Been Notified!";  
    private final long[] mVibratePattern = {100, 200, 300, 400, 500, 400, 300, 200, 400};  
    private String mChannelID;  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        createNotificationChannel();  
    }  
}
```



```
private void createNotificationChannel() {  
    NotificationManager mNotificationManager =  
        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);  
    mChannelID = getPackageName() + ".channel_01";  
    CharSequence name = getString(R.string.channel_name);  
    ...  
    mNotificationManager.createNotificationChannel(mChannel);  
}
```

```
public void onClick(View v) {  
    // Define action Intent  
    Intent mNotificationIntent = new Intent(getApplicationContext(),  
        NotificationSubActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    PendingIntent mContentIntent = PendingIntent.getActivity(getApplicationContext(), 0,  
        mNotificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
```

*// Define the Notification's expanded message and Intent:*

Notification.Builder notificationBuilder =

```
    new Notification.Builder(getApplicationContext(), mChannelID)
        .setTicker(tickerText)
        .setSmallIcon(android.R.drawable.stat_sys_warning)
        .setAutoCancel(true)
        .setContentTitle(contentTitle)
        .setContentText(contentText + " (" + ++mNotificationCount + ")")
        .setContentIntent(mContentIntent);
```

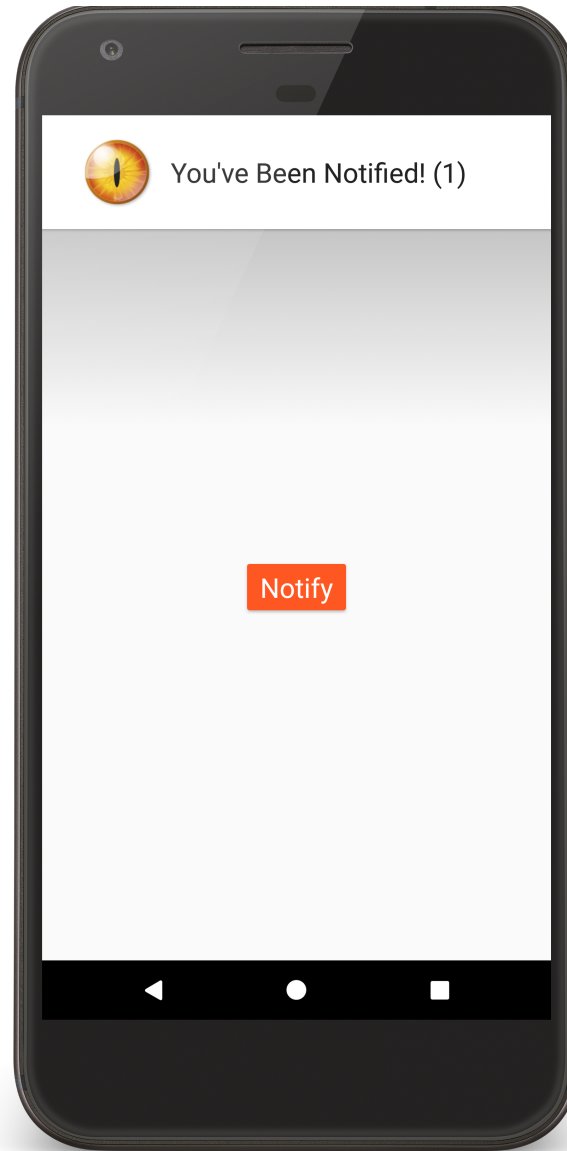
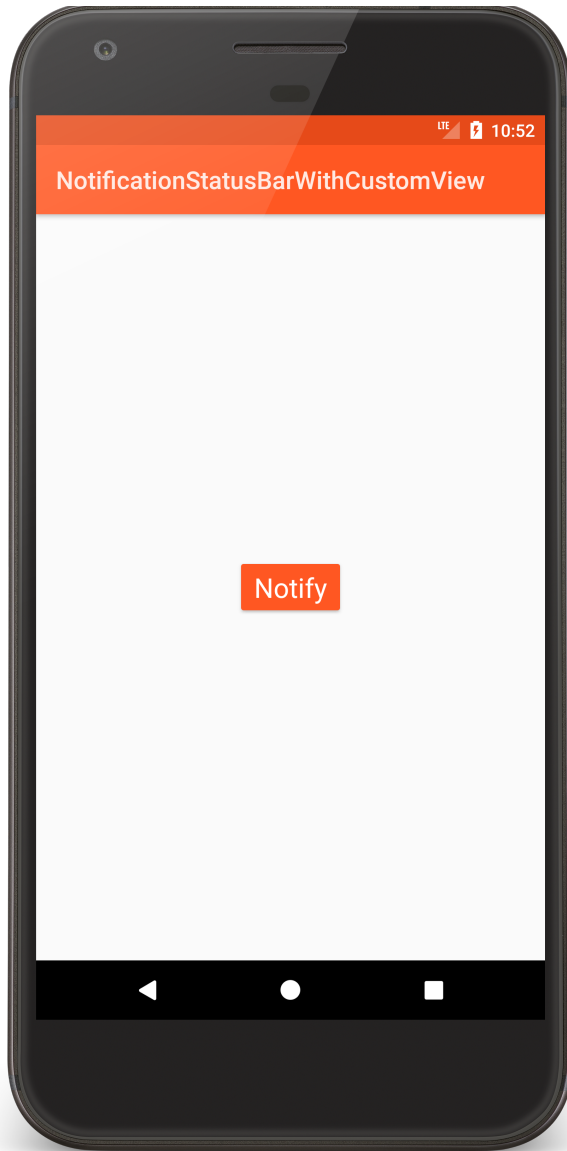
*// Pass the Notification to the NotificationManager:*

NotificationManager mNotificationManager =

```
    (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    mNotificationManager.notify(MY_NOTIFICATION_ID, notificationBuilder.build());
```

...

Notification  
StatusBar  
WithCustomView



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/toast_layout_root"
    ... >
    <ImageView
        android:id="@+id/notification_image"
        android:src="@drawable/fire_eye_alien"
        ... />
    <TextView
        android:id="@+id/notification_text"
        ... />

</RelativeLayout>
```

```
public void onClick(View v) {  
    ...  
    RemoteViews contentView = new RemoteViews(  
        getPackageName(),  
        R.layout.custom_notification);  
    contentView.setTextViewText(R.id.notification_text,  
                                mContentText + " (" + ++mNotificationCount + ")");  
    Notification.Builder notificationBuilder = new Notification.Builder(  
        getApplicationContext(), mChannelID)  
        .setTicker(mTickerText)  
        .setSmallIcon(android.R.drawable.stat_sys_warning)  
        .setAutoCancel(true)  
        .setContentIntent(contentIntent)  
        .setCustomContentView(contentView);  
    mNotificationManager.notify(MY_NOTIFICATION_ID,  
        notificationBuilder.build());  
}
```

Next Time

Threads