

CMSC436: Programming Handheld Systems

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

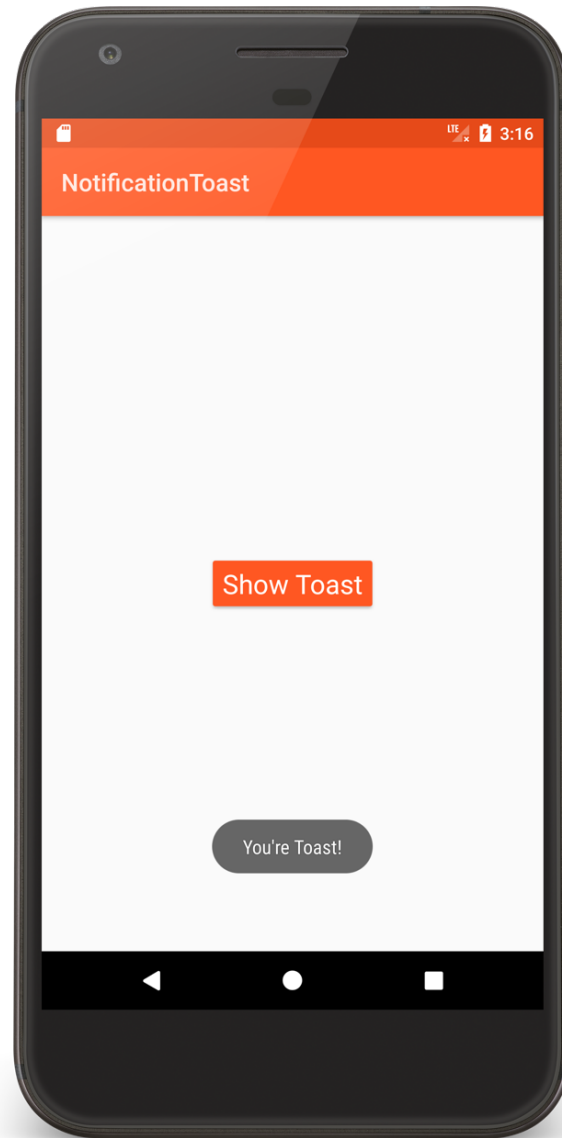
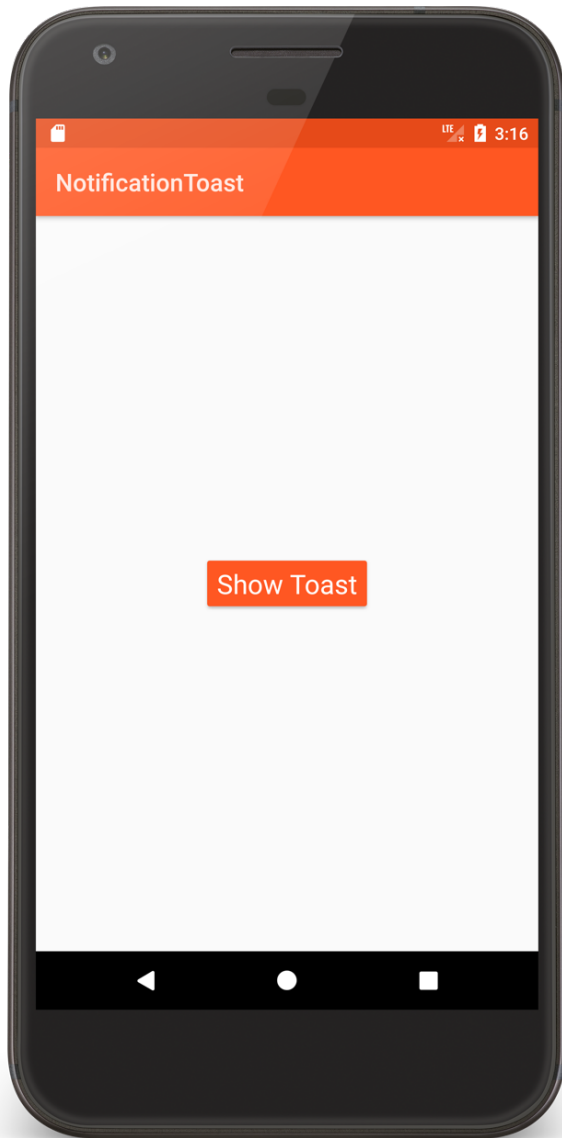
Instantiate a Toast object by calling

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

Show toast by calling

```
Toast.show()
```

Notification Toast



NotificationToastActivity.kt

```
class NotificationToastActivity : Activity() {
    public override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.main)
    }

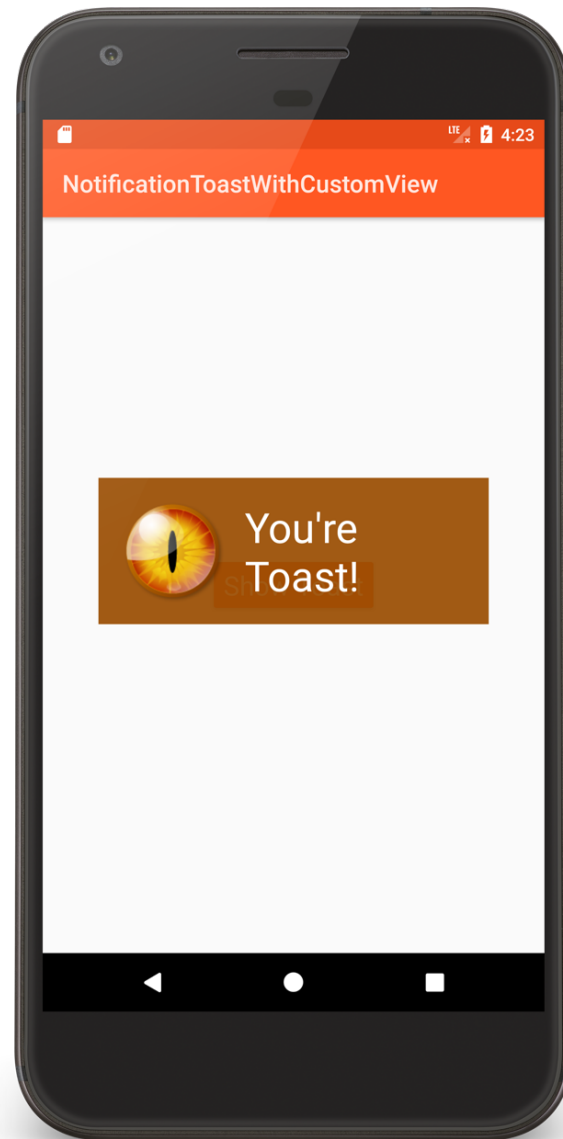
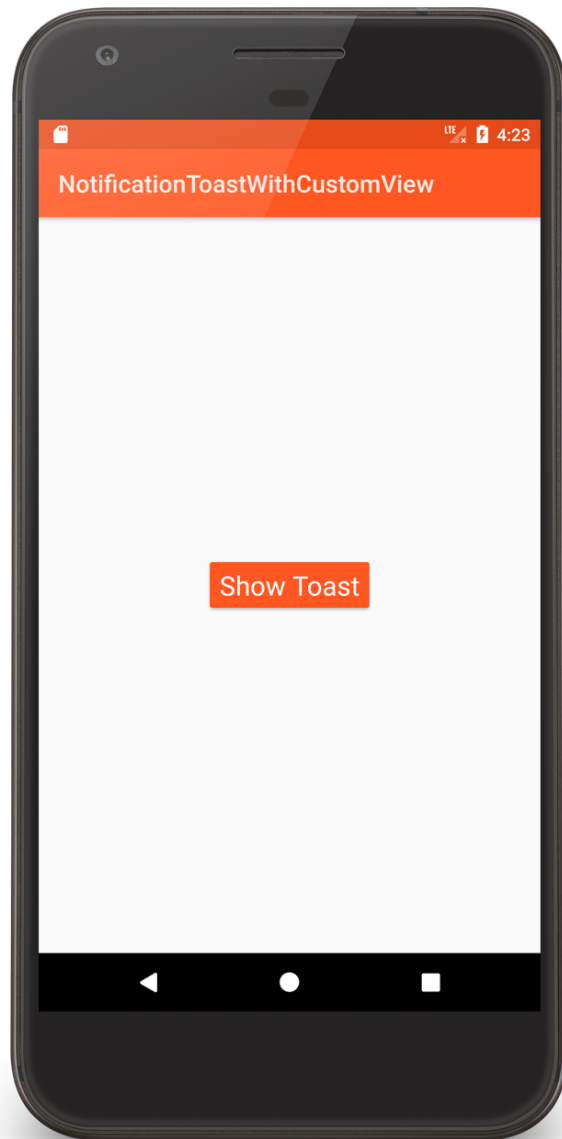
    fun onClick(v: View) {
        Toast.makeText(applicationContext,
            getString(R.string.youre_toast_string), Toast.LENGTH_LONG).show()
    }
}
```

Toast with Custom Views

Can create custom layouts in XML

Set them using `Toast.setView()`

NotificationToast WithCustomView



NotificationToastActivity.kt

```
fun onClick(v: View) {  
  
    val toast = Toast(applicationContext)  
    toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0)  
    toast.duration = Toast.LENGTH_LONG  
    toast.view = inflater.inflate(  
        R.layout.custom_toast,  
        findViewById(R.id.toast_layout_root)  
    )  
    toast.show()  
}
```

custom_toast.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    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_marginEnd="@dimen/activity_margin"
        android:src="@drawable/fire_eye_alien"
        android:layout_centerVertical="true"
        android:contentDescription="@string/eyeball_string"/>
```

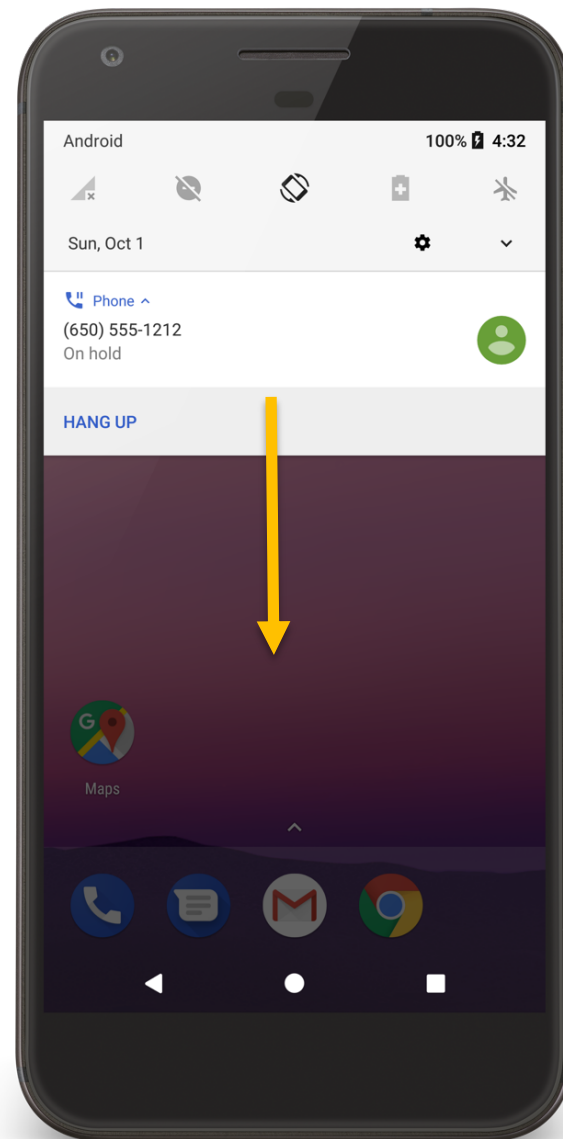
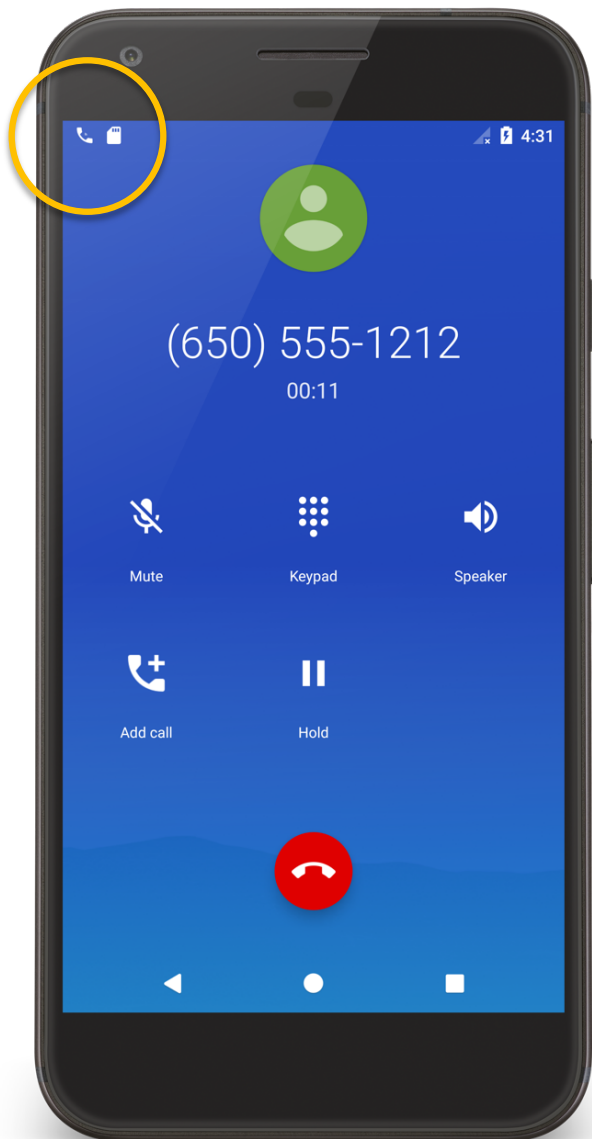
custom_toast.xml

```
<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_toEndOf="@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

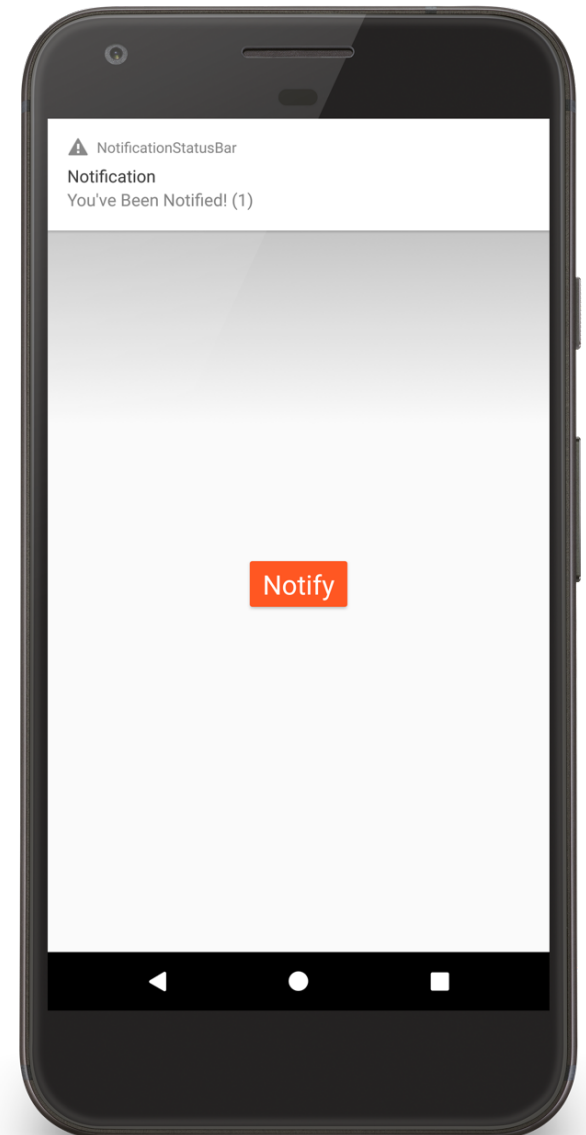
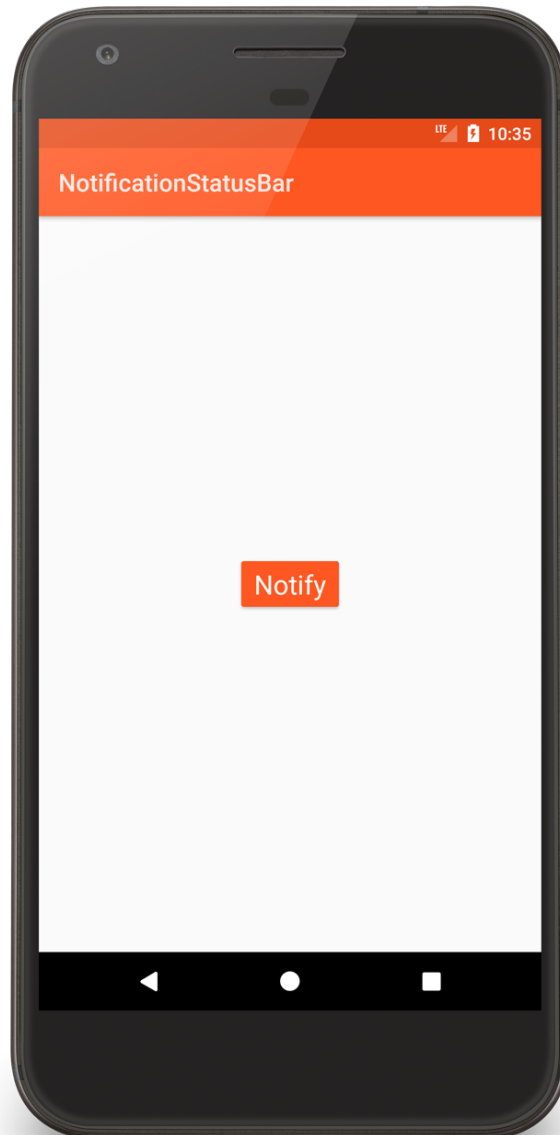
Creating a Notification Channel

Construct a notification channel object with a package-unique ID

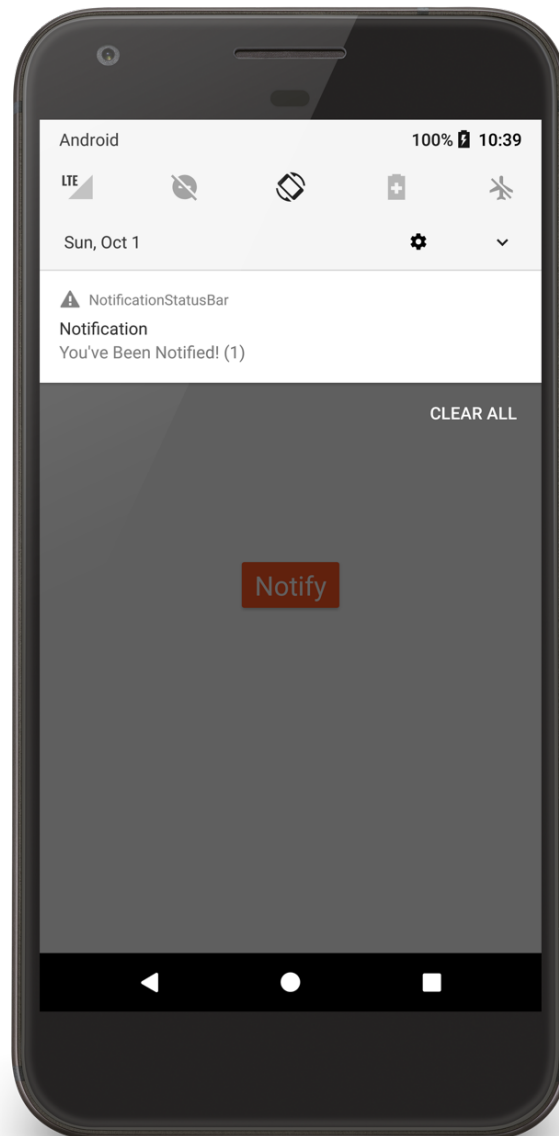
Configure the notification channel object

Submit the notification channel object to the notification manager

Notification StatusBar



Notification StatusBar



NotificationStatusBarActivity.kt

```
class NotificationStatusBarActivity : Activity() {
    companion object {
        // Notification ID to allow for future updates
        private const val MY_NOTIFICATION_ID = 1
        private const val KEY_COUNT = "key_count"
        private lateinit var mNotificationManager: NotificationManager
        private lateinit var mChannelID: String
        // Notification Text Elements
        private const val tickerText =
            "This is a Really, Really, Super Long Notification Message!"
        private const val contentTitle = "Notification"
        private const val contentText = "You've Been Notified!"
        private val mVibratePattern =
            longArrayOf(100, 200, 300, 400, 500, 400, 300, 200, 400)
        private lateinit var mSoundURI: Uri
    }
}
```

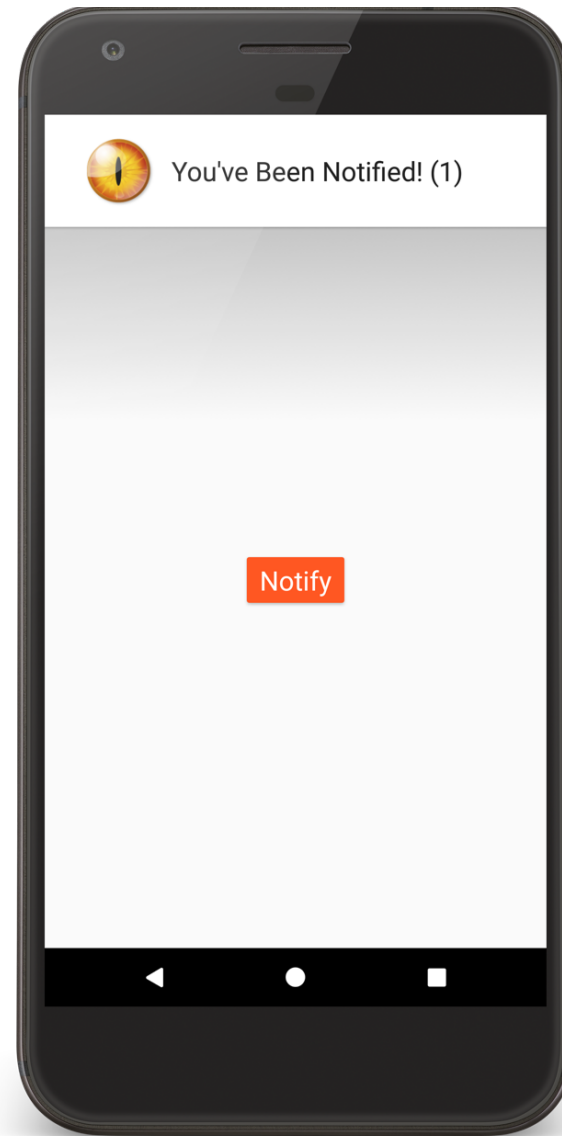
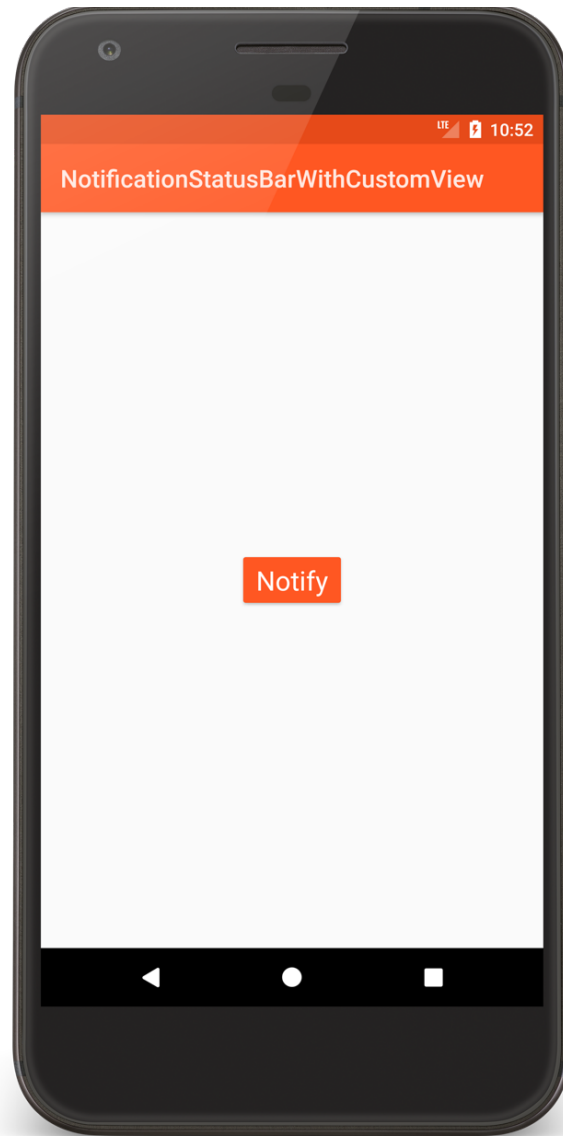

NotificationStatusBarActivity.kt

```
// Notification Count
private var mNotificationCount: Int = 0
public override fun onCreate(savedInstanceState: Bundle?) {
    ...
    mNotificationManager =
        getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    createNotificationChannel()
}
private fun createNotificationChannel() {
    ...
    val mChannel = NotificationChannel(mChannelID, name, importance)
    ...
    mNotificationManager.createNotificationChannel(mChannel)
}
```

NotificationStatusBarActivity.kt

```
fun onClick(v: View) {  
    // Define action Intent  
    val mNotificationIntent = Intent(applicationContext,  
        NotificationSubActivity::class.java).  
        setFlags(Intent.FLAG_ACTIVITY_NEW_TASK)  
    val mContentIntent = PendingIntent.getActivity(applicationContext,  
        0, mNotificationIntent, PendingIntent.FLAG_UPDATE_CURRENT)  
    ...  
}
```


Notification
StatusBar
WithCustomView



custom_notification.xml

```
<RelativeLayout ...>
  <ImageView
    android:id="@+id/notification_image"
    android:layout_width="88dp"
    android:layout_height="88dp"
    android:layout_centerVertical="true"
    android:contentDescription="@string/eye_desc_string"
    android:src="@drawable/fire_eye_alien" />
  <TextView
    android:id="@+id/notification_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_toEndOf="@id/notification_image"
    android:textAppearance=
      "@android:style/TextAppearance.Material.Large" />
</RelativeLayout>
```

NotificationStatusBarWithCustomViewActivity.kt

```
fun onClick(v: View) {  
    // Define action Intent  
    val notificationIntent = Intent(applicationContext,  
        NotificationSubActivity::class.java).  
        setFlags(Intent.FLAG_ACTIVITY_NEW_TASK)  
    val contentIntent = PendingIntent.getActivity(applicationContext,  
        0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT)  
    val contentView =  
        RemoteViews(packageName, R.layout.custom_notification)  
  
    contentView.setTextViewText(R.id.notification_text,  
        "$mContentText ( ${++mNotificationCount} )")  
}
```

NotificationStatusBarWithCustomViewActivity.kt

```
// Define the Notification's expanded message and Intent:
val notificationBuilder = Notification.Builder(applicationContext,
    channelId)
    .setTicker(mTickerText)
    .setSmallIcon(android.R.drawable.stat_sys_warning)
    .setAutoCancel(true)
    .setContentIntent(contentIntent)
    .setCustomContentView(contentView)

// Pass the Notification to the NotificationManager:
mNotificationManager.notify(MY_NOTIFICATION_ID,
    notificationBuilder.build())
}
```

Next Time

Threads and Handlers

Example Applications

NotificationToast

NotificationToastWithCustomView

NotificationStatusBar

NotificationStatusBarWithCustomView