Recording in Progress

This class is being recorded
Please turn off your video and/or video if you do not wish to be recorded
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

Show Toast

You're Toast!
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()
    }
}
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+)
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

Notify

Notify

You've Been Notified! (1)
Notification
StatusBar

Got The Intent
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
    }
}
// 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)
}
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)
    ...
}
// Define the Notification's expanded message and Intent:
val notificationBuilder = Notification.Builder(
    applicationContext, 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:
    mNotificationManager.notify(MY_NOTIFICATION_ID, notificationBuilder.build())
}
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 )")
// Define the Notification's expanded message and Intent:
val notificationBuilder = Notification.Builder(applicationContext,
mChannelID)
    .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

Firebase
Example Applications

NotificationToast
NotificationStatusBar
NotificationStatusBarWithCustomView