



Notification

Anderson Rodrigues



Intent

- O que são **Intents**?
- Tipos de **Intent**
- Utilizando **Intents**
- Como são instanciadas?



Intent

```
val intent = Intent(packageContext: this, FromNotificationActivity::class.java)
```

```
val intentActivity = Intent(App1BroadcastReceiver.ACTION)
```

```
val intent = Intent(context, CountdownService::class.java)  
intent.putExtra(EXTRA_COUNTDOWN_TIME, countDownTime)  
context.startService(intent)
```



PendingIntent

- O que é?
 - Possui uma ação associada
 - Permite a execução de uma ação
 - Activity, BroadcastReceiver, Service



Intent-filter

- Especifica os tipos de **Intents** que um componente pode responder
- Componentes:
 - Activity
 - Service
 - BroadcastReceiver
- Elementos:
 - Action
 - Category
 - Data



System Services

- Serviços a nível de sistema
- Obtidos através de um *Context*

Currently available classes are: `WindowManager`, `LayoutInflater`, `ActivityManager`, `PowerManager`, `AlarmManager`, `NotificationManager`, `KeyguardManager`, `LocationManager`, `SearchManager`, `Vibrator`, `ConnectivityManager`, `WifiManager`, `AudioManager`, `MediaRouter`, `TelephonyManager`, `SubscriptionManager`, `InputMethodManager`, `UiModeManager`, `DownloadManager`, `BatteryManager`, `JobScheduler`, `NetworkStatsManager`, `DomainVerificationManager`.

```
getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
```



Notification

- Mensagem exibida fora da aplicação
- Lembretes, mensagens, informações específicas da aplicação
- Pode levar o usuário à aplicação ou prover interação diretamente



Notification

- Alarme
- iFood, 99Food, Uber Eats
- YouTube Music, Spotify, Deezer
- WhatsApp, Facebook
- Gmail
- Google Play Store

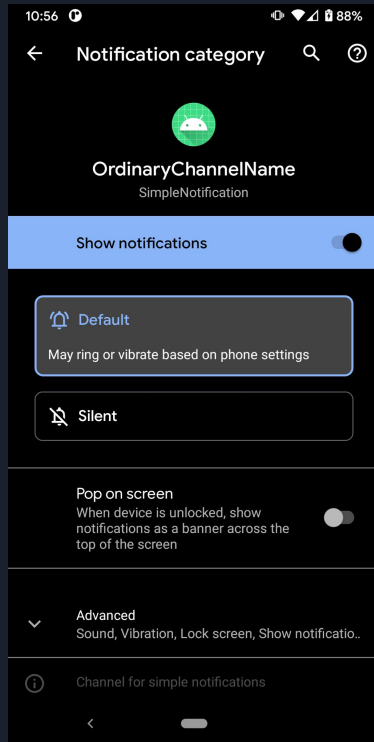
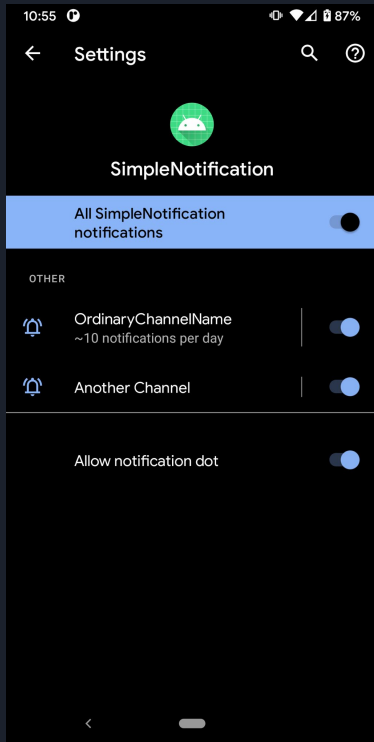


Notification

NotificationChannel

- Android 8+
- Notificações precisam estar associadas a um NotificationChannel
- Usuário tem controle sobre todos os NotificationChannels de um app

Notification





Notification

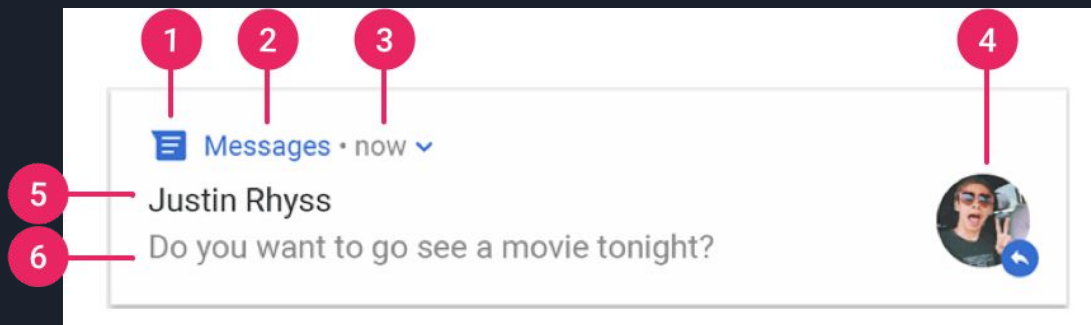
- name
- description
- importance
 - HIGH (urgent): sound + heads-up
 - DEFAULT (High): sound
 - LOW (Medium): no sound
 - MIN (Low): no sound + sem presença na status bar



Notification

```
val importance = NotificationManager.IMPORTANCE_DEFAULT
val channel = NotificationChannel(NOTIFICATION_CHANNEL, name, importance).apply {
    this.description = description
    enableLights( lights: true)
    lightColor = Color.BLUE
    enableVibration( vibration: true)
    vibrationPattern = longArrayOf(0L, 100L, 200L, 300L)
    setShowBadge(true)
}
```

Notification



1 - smallIcon

2 - app name

3 - Time stamp by the system

4 - largeIcon

5 - ContentTitle

6 - ContentText



Notification

```
val builder = NotificationCompat.Builder(context: this, NOTIFICATION_CHANNEL)
    .setSmallIcon(R.drawable.ic_launcher_foreground)
    .setContentTitle("Title")
    .setContentText(text)
    .setAutoCancel(true)
```

```
val manager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
manager.notify(id: 0, builder.build())
```



Notification

Atividade

- Criar uma aplicação que, ao clique de um botão, lança uma notificação que contenha o texto inserido pelo usuário. (Activity + EditText + Button)



Notification

- Notificações também podem interagir com componentes
- Abrir uma Activity, lançar um broadcast, comunicar com um service

```
val intent = Intent( packageContext: this, FromNotificationActivity::class.java).apply { this: Intent
    |   flags = Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK
    | }

val pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, intent, PendingIntent.FLAG_ONE_SHOT)

    .setContentIntent(pendingIntent)
```




Notification

Antes de continuar...

```
.setStyle(NotificationCompat.BigTextStyle().bigText(text))
```



Notification

Atividade

- Criar uma segunda Activity que será chamada quando o usuário clicar na notificação
- Extra: Exibir a mensagem digitada pelo usuário na tela anterior



Notification

Action

- O que são e como funcionam?
- Não devem ter a mesma ação do clique na notificação



Notification

```
val cancelAction = NotificationCompat.Action.Builder(android.R.drawable.ic_menu_close_clear_cancel,  
    "Cancel",  
    pendingIntent  
)
```



Notification

Atividade

- Adicionar uma action à notificação. Ao clique, um *BroadcastReceiver* deve recebê-la e exibir um log



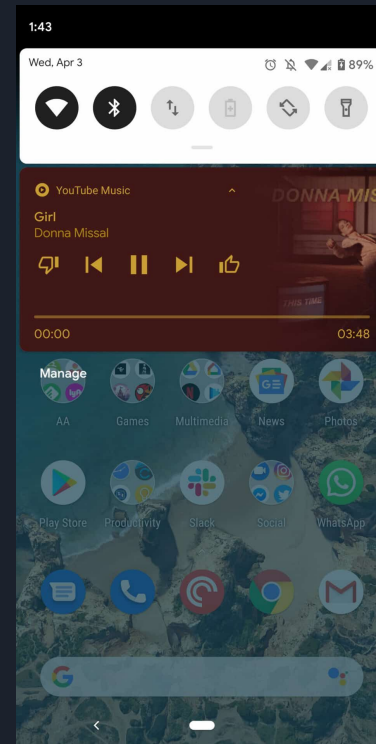
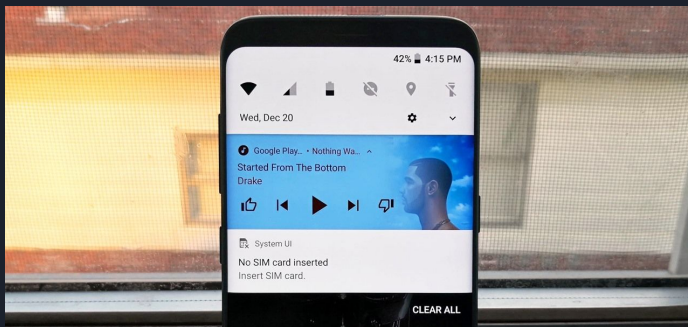
Notification

Media Notifications

- `setStyle`
- `setLargeIcon`

```
.setLargeIcon(bitmap)  
.setStyle(androidx.media.app.NotificationCompat.MediaStyle())
```

Notification





Notification

Custom Notification

- Layout customizado
- Campos diferentes dos especificados pela API
- Cuidado com as cores e backgrounds



Notification

RemoteViews

- Utilizadas em Notificações e Widgets

<https://developer.android.com/reference/android/widget/RemoteViews>



Notification

Passos para uma Custom Notification

- Criar o layout
- Instanciar via RemoteViews
- Adicionar ações (opcional)
- Definir como layout da Notification



Notification

 notification_download.xml

```
val downloadLayout = RemoteViews(context.packageName, R.layout.notification_download)
```

```
.setStyle(NotificationCompat.DecoratedCustomViewStyle())  
.setCustomContentView(downloadLayout)
```



Notification

Atividade

- Criar uma Custom Notification



Notification

Groups

- Agrupa notificações por grupo
- Instagram DM, Requisições de Amizade, Marcações
- Diferentes comportamentos API > 24 (Android Nougat 7.0)
 - Summary automático pelo sistema



Notification

```
builder.setGroup(GROUP_NAME_1)
```

★ **Note:** The behavior of the group summary may vary on some device types such as wearables. To ensure the best experience on all devices and versions, always include a group summary when you create a group.



Notification

```
private fun createSummary() {  
    val summary = NotificationCompat.Builder( context: this, NOTIFICATION_CHANNEL)  
        .setContentTitle("Summary of your notifications")  
        .setStyle(NotificationCompat.InboxStyle()  
            .addLine( cs: "Theres a new band in town")  
            .setBigContentTitle("New Messages")  
            .setSummaryText("Just another message")  
        )  
        .setContentText("You've got new messages")  
        .setSmallIcon(R.drawable.ic_launcher_foreground)  
        .setGroup(GROUP_NAME_1)  
        .setGroupSummary(true)  
        .build()  
  
    val manager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager  
    manager.notify( id: 12, summary)  
}
```



Notification

Conversation

- Assim com as Actions, permite interação do usuário sem o app estar aberto
- Permite input de textos



Notification

```
val action = NotificationCompat.Action.Builder(android.R.drawable.ic_menu_send,  
    "Reply",  
    getPendingIntent(context))  
    .addRemoteInput(getRemoteInput(context))  
    .build()
```

```
return RemoteInput.Builder(KEY_REPLY_CONTENT).run {  
    setLabel("Reply")  
    build() ^run  
}
```

```
val value = RemoteInput.getResultsFromIntent(intent).getCharSequence(NotificationHelper.KEY_REPLY_CONTENT)
```



Notification

Atividade

- Criar uma notificação que, ao ser respondida, abre uma Activity e exibe o texto digitado



Notification

Bubbles

- Disponível a partir do Android 11 (API level 30)
- Permite que o usuário selecione qual notificação se torna uma Bubble
- Flutua sobre outros contextos
- BubbleMetadata



Notification

Android 11 +

- Usar o *MessagingStyle*
- Ter um *Shortcut* associado

Android <= 10

- Usar o *MessagingStyle* com uma *Person* associada
- Ser notificação de um *ForegroundService* com categoria *CALL* + *Person* associada
- App em *foreground* quando a notificação for enviada



Notification

Shortcut

- Long-press no ícone da aplicação
- Atalhos para alguma funcionalidade
- Instagram, Chrome, YouTube Music



Notification

```
val shortcutIntent = Intent(context, MainActivity::class.java)
shortcutIntent.flags = Intent.FLAG_ACTIVITY_CLEAR_TOP
shortcutIntent.action = Intent.ACTION_VIEW

val shortcut = ShortcutInfoCompat.Builder(context, id: "0")
    .setLocusId(LocusIdCompat(id: "0"))
    .setActivity(ComponentName(context, MainActivity::class.java))
    .setShortLabel("Cesar School")
    .setIcon(IconCompat.createWithResource(context, R.drawable.cover_012))
    .setLongLived(true)
    .setIntent(shortcutIntent)
    .setPerson(partner)
    .build()
```

```
ShortcutManagerCompat.addDynamicShortcuts(context, listOf(shortcut))
```



Notification

```
val bubble = NotificationCompat.BubbleMetadata.Builder(pendingIntent, IconCompat.createWithResource(context, R.drawable.cover_012))  
    .setDesiredHeight(600)  
    .setAutoExpandBubble(true)  
    .build()
```

```
val builder = NotificationCompat.Builder(context, NOTIFICATION_CHANNEL)  
    .setSmallIcon(R.drawable.ic_launcher_foreground)  
    .setBubbleMetadata(bubble)  
    .setContentIntent(pendingIntent)  
    .setShortcutId("0")  
    .setLocusId(LocusIdCompat(id: "0"))  
    .setCategory(NotificationCompat.CATEGORY_MESSAGE)  
    .addPerson(partner)  
    .setStyle(NotificationCompat.MessagingStyle(partner))
```



Notification

```
<activity android:name=".ui.MainActivity"  
    android:resizeableActivity="true"  
    android:allowEmbedded="true"  
    android:documentLaunchMode="always">
```




Notification

Atividade:

- Criar uma aplicação de chat que, ao receber uma nova mensagem, exibe esta mesma mensagem em uma notificação. Além disso, deve possibilitar ao usuário a ação de resposta. A resposta deve ser enviada para o servidor.



Extras

- LiveData
- Flow
- FCM
- ViewModel