



Services

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



Services

Qual a definição de:

- Activity
- BroadcastReceiver
- Service



Services

- Responsável por executar operações de longa duração
- Executa suas ações em background
- Nem sempre permite interação com o usuário
- Não é um processo à parte (a não ser que especificado via Manifest)
- Não é uma thread



Services

```
class CountdownService: Service() {
```

```
    override fun onBind(intent: Intent?): IBinder? {  
        return null  
    }  
}
```




Services

```
override fun onCreate() {  
    super.onCreate()  
}
```

```
override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {  
    return START_STICKY  
}
```



Services

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



Services

- onStartCommand
 - Intent
 - Flags
 - START_FLAG_REDELIVERY -> START_REDELIVER_INTENT
 - START_FLAG_RETRY
 - startId -> gerado pelo sistema para cada chamada



Services

- onStartCommand
 - START_STICKY -> sistema tentará recriar o serviço -> Intent será null
 - START_NOT_STICKY -> não há recriação -> startService
 - START_REDELIVER_INTENT -> service recriado com a última intent válida



Services

Assim como outros componentes...

```
<service android:name=".service.CountDownService" />
```



Services

Atividade:

- Criar um service



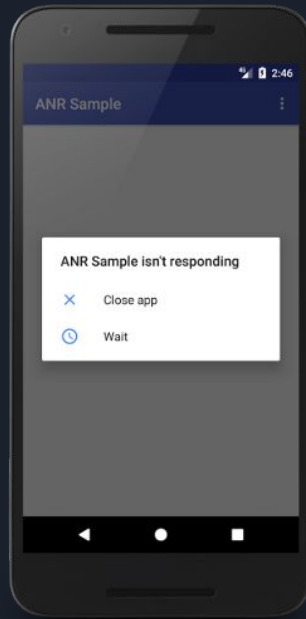
Service

Atividade:

- Criar uma aplicação que, ao clique de um botão, inicia um serviço. Ao ser iniciado, o Service deve imprimir 5 mensagens de log com um intervalo de 10 segundos entre elas



Services



<https://developer.android.com/topic/performance/vitals/anr>



Services

ANR

- Application is Not Responding
- Activity não respondeu a eventos de entrada nos últimos 5 segundos
- BroadcastReceiver não terminou sua execução no tempo especificado



Services

- Service não é uma thread
- Executa suas ações na thread principal da aplicação (UI Thread)
- Como evitar problemas de ANR?



Services

- Thread
- AsyncTask = '{
- Coroutines
- Handler



Services

Tipos de Services:

- Started
 - Background
 - Foreground
- Bound



Services

Background Services

- Não há interação com o usuário
- Rodam em background
- Têm limitação de tempo (Android 8.0+, API level 26+)
 - <https://developer.android.com/about/versions/oreo/background>



Services

Foreground Services

- Regras de execução de threads também se aplica
- Necessita, obrigatoriamente, de uma notificação
- Notificação está presente até o fim da execução



Services

- Exemplos:
 - Boose
 - Waze
 - WhatsApp -> Verificando mensagens
 - Serviços do sistema



Service

```
val intent = Intent(context, CountdownService::class.java)  
ContextCompat.startForegroundService(context, intent)
```

```
val intent = Intent(context, CountdownService::class.java)  
context.stopService(intent)
```




Services

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

```
override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {  
  
    startForeground( id: 112, NotificationHelper.getForegroundNotification( context: this))
```

```
stopForeground( flags: 0)
```



Services

- stopForeground (dentro do Service)
- Flags:
 - STOP_FOREGROUND_DETACH
 - STOP_FOREGROUND_REMOVE



SERVICES

Atividade:

- Mude o seu Service de background para foreground
- Lembre da mudanças na forma de iniciá-lo



Services

Bound Services

- Modo de iniciar diferente
- Atrelado ao componente
- Cliente-servidor
- IPC -> AIDL



Service



deezer



Spotify®



YouTube Music



Music



Services

Para Services do tipo Bound (exclusivamente)

- Não utilizamos o `startService/startForeground`
- Não lidamos com Intents pelo `onStartCommand`
- Acesso direto a métodos (não é necessário enviar dados via Intent)
- Suporta múltiplas conexões



Services

```
override fun onBind(intent: Intent?): IBinder {  
    return binder  
}
```

```
inner class CountdownBinder: Binder() {  
    fun getService(): CountdownService = this@CountDownService  
}
```



Services

```
private var mCountDownService: CountdownService? = null

private val mConnection = object: ServiceConnection {
    override fun onServiceConnected(name: ComponentName?, service: IBinder?) {
        val binder = service as CountdownService.CountdownBinder
        mCountDownService = binder.getService()
    }

    override fun onServiceDisconnected(name: ComponentName?) {
        mCountDownService = null
    }
}
```




Services

```
override fun onStart() {  
    super.onStart()  
  
    val intent = Intent(packageContext: this, CountdownService::class.java)  
    bindService(intent, mConnection, Context.BIND_AUTO_CREATE)  
}
```

```
override fun onStop() {  
    super.onStop()  
  
    mCountDownService?.setCountdownListener(null)  
    unbindService(mConnection)  
}
```



Services

Atividade:

- Transforme seu Service em um Bound Service e implemente os métodos ao invés de utilizar dados via Intent



Services

É possível ter um Service do tipo Bound e do tipo Started ao mesmo tempo?



Services

- Services podem ser Bound e Started
- Implementar tanto o `onBind` quanto o `onStartCommand`
- Caso o `unbind` seja chamado, o Service para a não ser que uma chamada do tipo `startService` tenha sido feita



IntentService



This class was deprecated in API level 30.

IntentService is subject to all the [background execution limits](#) imposed with Android 8.0 (API level 26). Consider using [WorkManager](#) or [JobIntentService](#), which uses jobs instead of services when running on Android 8.0 or higher.



IntentServices

- Subtipo de Service
- Assíncrono (sem threads, coroutines e afins)
- Assim que termina a execução o IntentService é terminado



IntentServices

All requests are handled on a single worker thread – they may take as long as necessary (and will not block the application's main loop), but only one request will be processed at a time.



IntentServices

```
class CountdownService: IntentService(name: "CountDownService")
```

```
    override fun onHandleIntent(intent: Intent?)
```




IntentServices

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

```
val intent = Intent(context, CountdownService::class.java)  
context.stopService(intent)
```



IntentServices

Atividade:

- Criar uma aplicação que, ao clique de um botão, inicia um serviço. Ao ser iniciado, o IntentService deve imprimir 5 mensagens de log com um intervalo de 10 segundos entre elas