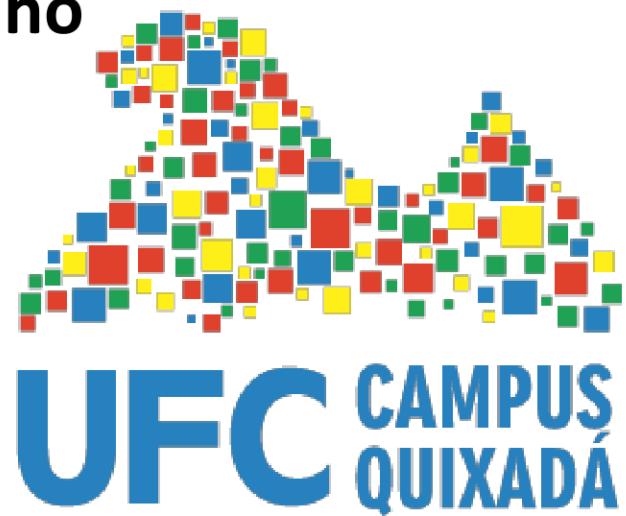


Desenvolvimento de software para Dispositivos Móveis

Services e Broadcast Events



Prof. Sidartha Carvalho



Services no Android

Services

- Service é um processamento Android que não exige uma visualização para o usuário (diferente das Activities).
- Não exige um layout.xml.
- Tipos de Service
 - **Foreground**
 - Operação que é perceptível ao usuário.
 - Exige a exibição de uma notificação do App na barra de notificações.
 - Exemplo: tocar uma música
 - **Background**
 - Operação que não é facilmente perceptível ao usuário.
 - Exemplo: enviar os dados do aplicativo para a nuvem ou compactar os dados.
 - **Bounded**
 - Serviço que está vinculado a um aplicativo e permite a comunicação entre o serviço e o app que o chamou.
 - Deve executar somente enquanto a aplicação que o criou estiver sendo executada.

Criar um Serviço: Passos

1. Criar a classe que irá implementar o serviço, deve “extends Service” e implementar os métodos onStartCommand() e onDestroy().
2. Criar a Intent para iniciar o serviço e chamar o método StartService() passando a Intent.
3. Adicionar o Service no AndroidManifest.

Criar um Serviço: Passo 1

```
public class MusicInBackgroundService extends Service {  
  
    MediaPlayer mediaPlayer;  
  
    @Nullable  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
  
    @Override  
    public int onStartCommand(Intent intent, int flags, int startId) {  
  
        mediaPlayer = MediaPlayer.create(this, Settings.System.DEFAULT_RINGTONE_URI);  
        mediaPlayer.setLooping(true);  
        mediaPlayer.start();  
  
        return START_STICKY;  
    }  
  
    @Override  
    public void onDestroy() {  
        super.onDestroy();  
  
        mediaPlayer.stop();  
    }  
}
```

Criar um Serviço: Passo 2

```
Intent startMusicIntent = new Intent(v.getContext(), MusicInBackgroundService.class);  
startService(startMusicIntent);
```

```
Intent stopMusicIntent = new Intent(v.getContext(), MusicInBackgroundService.class);  
stopService(stopMusicIntent);
```

Criar um Serviço: Passo 2

```
public void onClick(View v) {  
    if(v == btn_start){  
        Log.d("sidartha", "start music button");  
        Intent startMusicIntent = new Intent(v.getContext(), MusicInBackgroundService.class);  
        startService(startMusicIntent);  
    }else if(v == btn_stop){  
        Log.d("sidartha", "stop music button");  
        Intent stopMusicIntent = new Intent(v.getContext(), MusicInBackgroundService.class);  
        stopService(stopMusicIntent);  
    }  
}
```

Criar um Serviço: Passo 3

```
</application>
```

```
// DECLARAÇÃO DAS ACTIVITIES
```

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

```
</application>
```

Criar um Serviço: Passo 3

- O que criamos até o momento foi um Background Service.
- O bounded é o mesmo, mas com uma comunicação/troca de informações entre a aplicação que chamou e o serviço.
- `startService()` vs `startForegroundService()`
 - `startForegroundService()` deve ser usado para iniciar um serviço em primeiro plano que precisa continuar em execução mesmo em condições de poucos recursos. Isso é necessário para evitar a limitação de serviços em segundo plano em dispositivos com Android Oreo e versões posteriores.
 - Quando se utiliza `startForegroundService()`, é necessário chamar o método `startForeground()` dentro do serviço para exibir uma notificação na barra de status. Essa notificação informa ao usuário que o serviço está em execução e permite que ele interaja com o serviço conforme necessário.

Resultado

- Após clicar em Play Music, mesmo se você fechar a aplicação a música continuará a tocar.
- Isso não acontecia no nosso AnimalsApp, quando a aplicação perdia a atenção do usuário, a música era interrompida.



Eventos de Broadcast

Broadcast

- Os apps Android podem enviar ou receber mensagens de Broadcast do sistema Android e de outros aplicativo, similar ao padrão de design Publish–Subscribe.
- Essas transmissões são enviadas quando ocorrem alguns eventos interessantes.
- Exemplos: o dispositivo começa a carregar, o modo avião é ligado ou desligado, o Wifi é desconectado, etc.
- Os apps também podem enviar transmissões personalizadas, por exemplo, para notificar outros apps sobre algo que possam ser do interesse deles.
- Quando uma transmissão é enviada, o sistema a roteia automaticamente para apps que se inscreveram para receber esse tipo específico de transmissão.

Broadcast: recebendo eventos do modo avião

- Passos
 1. Criar uma classe Java que “extends BroadcastReceiver” e implementar o método `onReceive()` necessário.
 2. Escolher qual evento deseja receber dentro do `onReceive()`.
 3. Adicionar a permissão no `AndroidManifest` manualmente ou de forma dinâmica, usando código Java e `IntentFilter` (recomendado).
- Para cada evento há um nome associado
 - Mais de 200 eventos nativos do Android

<https://developer.android.com/about/versions/11/reference/broadcast-intents-30?hl=pt-br>

Broadcast: recebendo eventos do modo avião

```
public class AirplaneModeReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        //Este método é chamado toda vez que um evento de Broadcast  
é lançado  
        if (intent.getAction() != null &&  
            intent.getAction().equals(Intent.ACTION_AIRPLANE_MODE_CH  
ANGED))  
        {  
            boolean isAirplaneModeOn =  
intent.getBooleanExtra("state", false);  
  
            String msg = isAirplaneModeOn ? "Airplane Mode is ON" :  
"Airplane mode is OFF";  
  
            Toast.makeText(context,  
                ""+msg,  
                Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

Broadcast: recebendo eventos do modo avião

```
//MainActivity
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Registrar de forma dinâmica a permissão necessária
    IntentFilter intentFilter = new
IntentFilter("android.intent.action.AIRPLANE_MODE");
    AirplaneModeReceiver br = new AirplaneModeReceiver();
    registerReceiver(br, intentFilter);
}
```

Referências

- <https://developer.android.com/>

Dúvidas?



www.shutterstock.com · 744867163