

Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de Computadores  
Programação em Dispositivos Móveis  
2019

Android App *GeniuZ* para apresentação de informações da Last.fm Web API.

Utilização de RecyclerView com implementação de ArtistAdapter para dados só em memória (i.e. Array<ArtistDto>)

Simulação de obtenção dos dados sobre LastfmWebApiMock para efeitos de testes unitários.

Utilização de ViewModel como forma para manutenção de estado em reconfiguração.

- Activity --> ViewModel --> LastfmWebApi

ArtistsViewModel instanciado por um ViewModelProvider.Factory.

- ViewModelProviders.of(this, LasftfmViewModelProviderFactory(application as GeniuzApp))

Application -- Classe base para manter o estado global da aplicação, e.g GeniuzApp -- para manter uma instância de LastfmWebApi.

Manifesto: <application android:name=".GeniuzApp" ... />`

Contexto para o problema de operações de IO, e.g. parsing ou outro trabalho de background não poderem ocupar a UI thread.

Observação do resultado de executar IO na main thread. Exemplo: pedido HTTP bloqueante via URL (LastfmWebApiBlocking) – NetworkOnMainThreadException.

Play tracks from external storage:

- i. Use MediaPlayer without Service => stops if it looses foreground
- ii. Use MediaPlayer in a Service => stops if the application is destroyed
- iii. Use MediaPlayer in a *foreground* Service => doesn't stop

- *Component* -> startService(Intent) -> onStartCommand(Intent)
- *Component* -> stopService(Intent) -> onDestroy()
- *Component* -> startForegroundService *to run in Foreground*:
  - i. <uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />
  - ii. *Service* must call -> startForeground(id, Notification)
    - otherwise: Context.startForegroundService() did not then call Service.startForeground()