

Modelação e Padrões de Desenho

LEIRT 2024

Enunciado do Trabalho 3

Data limite de entrega: 8 de junho

Objectivos: Prática de programação assíncrona com **CompletableFuture**

No âmbito do projeto `music_all` pretende-se tornar a sua API (**LastfmWebApi**) e o serviço (**MusicAllService**) assíncronos. Na API, trocando o retorno de **List<T>** por **CompletableFuture<List<T>>**. No serviço, trocando o retorno de **Stream<T>** ou **Supplier<Stream<T>>** por **CompletableFuture<Stream<T>>**. Deverão copiar o módulo `music-all-async`, parcialmente implementado, para o vosso repositório, e completar as classes **LastfmWebApi**, **MusicAllService**, **Artist** e **Album**, com base no anterior `music-all-streams`, adaptando-as de acordo com os requisitos deste enunciado.

NOTAS:

- Não poderão bloquear sobre o resultado das computações assíncronas (i.e. usar os métodos **join** ou **get** de **CompletableFuture**) com exceção dos testes unitários.
- Os métodos do serviço, ao contrário das fases anteriores, não têm de ser *lazy*, mas os métodos associados aos objetos de domínio (**Artist** e **Album**), só poderão invocar a API na invocação desses métodos e não na construção dos objetos de domínio.
- O módulo não deverá ter nenhuma dependência com nenhum outro módulo dos trabalhos anteriores.

Fase 0

1. Fazer *clone* do repositório em <https://github.com/isel-leic-mpd/music-all-students-template>
2. Copiar o módulo `music-all-async` para o vosso repositório
3. Adicionar a `settings.gradle` o novo módulo `music-all-async`.
4. Remover do vosso projeto as pastas `.idea` e `.gradle`.

Todas estas ações deverão ser realizadas não estando o vosso projeto aberto no IntelliJ.

Fase 1

1. Completar e testar a classe **LastfmWebApi**.
2. Implementar e testar (realizando testes específicos) os métodos **searchArtistPar** e **getAlbumsPar** da classe **MusicAllService**. Os métodos permitem obter em paralelo duas páginas consecutivas de artistas e álbuns, respetivamente. De forma a valorizar o trabalho, poderão opcionalmente realizar variantes dos métodos que recebam por parâmetro o número de páginas a obter em paralelo.
3. Completar e testar (realizando testes específicos) os métodos **getAlbumTracks** e **getArtistDetail** da classe **MusicAllService**.
4. Completar e testar os métodos **searchArtist** e **getAlbums** da classe **MusicAllService**. Os métodos deverão tirar partido dos métodos **searchArtistPar** e **getAlbumsPar** realizados no ponto 1, mas terão de criar uma cadeia de chamadas que permita obter até ao número de elementos (artistas ou álbuns) indicados por parâmetro.

Bom trabalho,
Jorge Martins