Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Modelação e Padrões de Desenho

2018

A biblioteca movlazy disponibiliza informação detalhada sobre filmes. Os dados são obtidos a partir de uma API RESTful: <https://api.themoviedb.org>. O modelo de domínio é formado pelas entidades: SearchItem, Movie, CastItem e Actor e obedece à especificação apresentada no diagrama de classes da Figura 1.

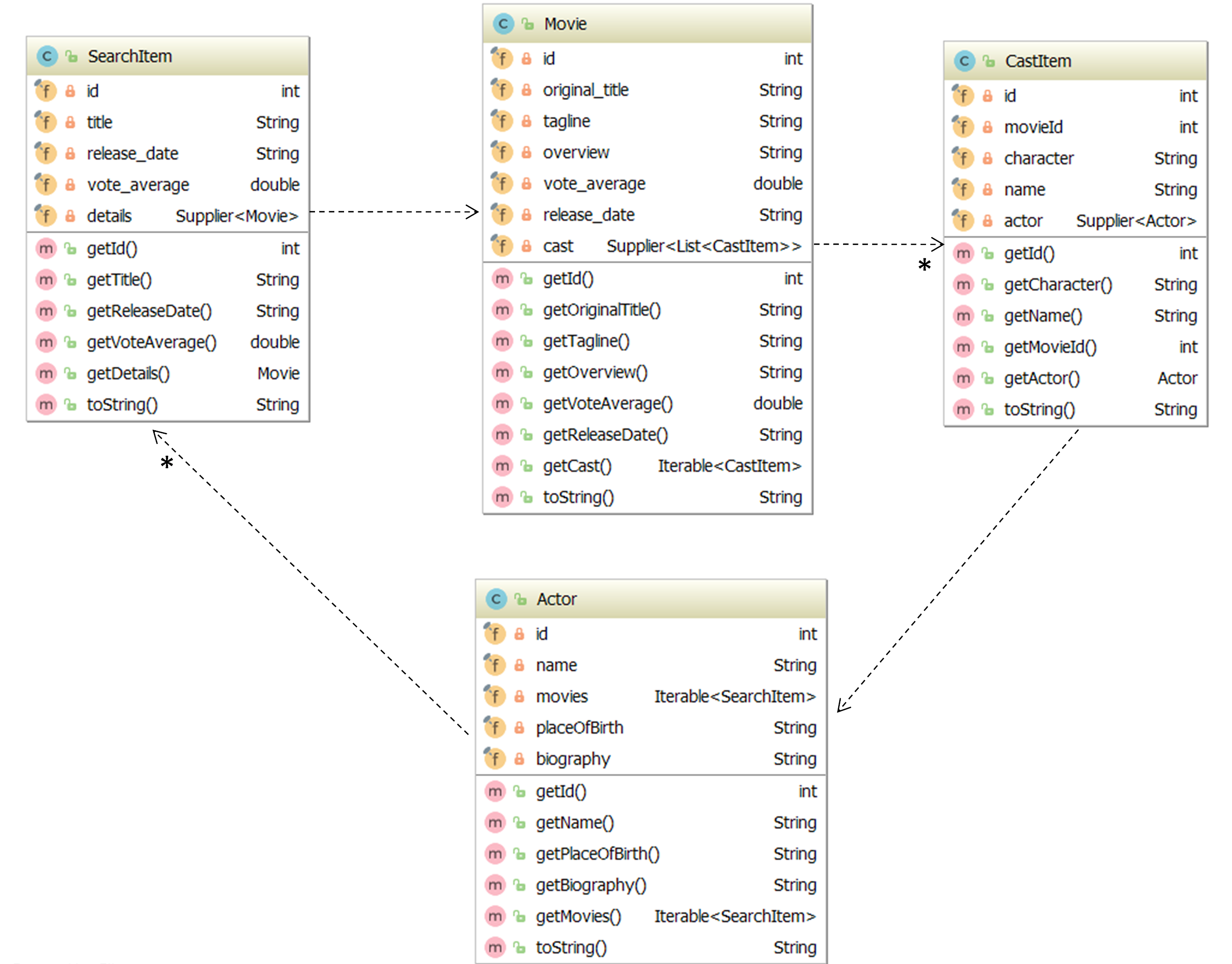
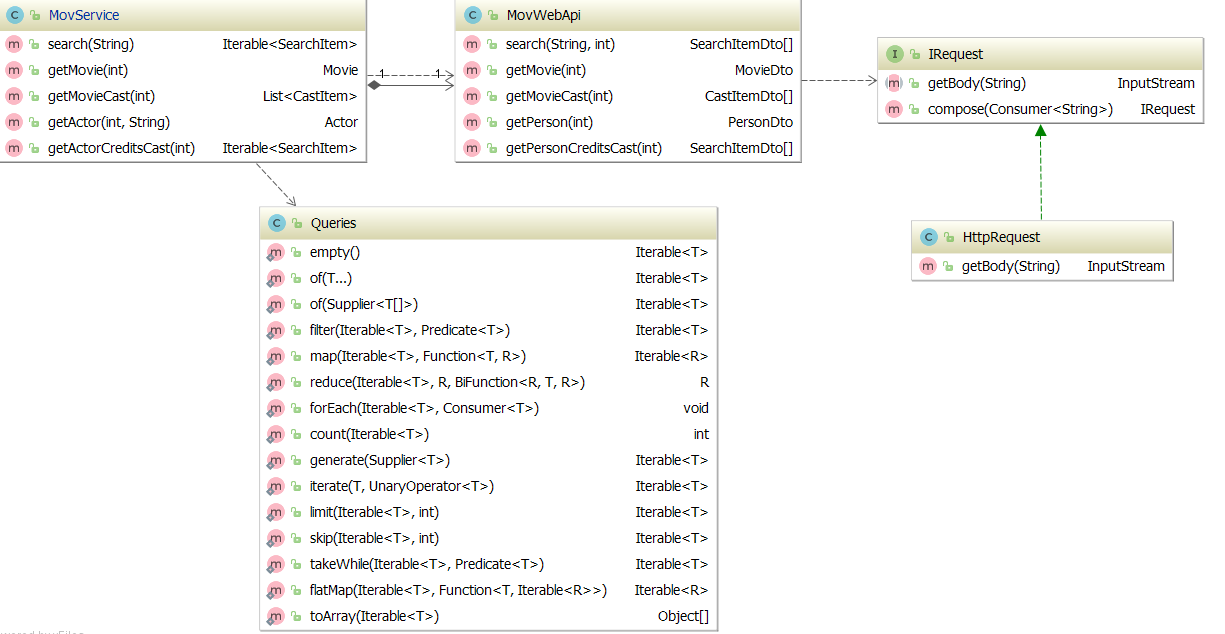


Figura 1

As classes do modelo de domínio estão implementadas no módulo movlazy. As relações entre as entidades de domínio são mantidas da seguinte forma:

* Todas as relações são *lazy*, ou seja o pedido à fonte de dados só é feito quando consultada a propriedade que fornece os objectos de domínio, e.g. getDetails(), getCast(), getActor() e getMovies(). Cada uma destas propriedades é suportada num campo de um tipo com semântica *lazy*, e.g. Supplier ou Iterable.
* As instâncias de Movie, CastItem e Actor são mantidas num cache da instância de MovService cuja estrutura já está definida nos campos movies, cast e actors.
* Não faça cache das instâncias de SearchItem.



As informações da fonte de dados são obtidas a partir dos seguintes *endpoints* da API RESTful <https://api.themoviedb.org>:

* [*https://developers.themoviedb.org/3/search/search-movies*](https://developers.themoviedb.org/3/search/search-movies)
* <https://developers.themoviedb.org/3/movies>
* <https://developers.themoviedb.org/3/movies/get-movie-credits>
* [*https://developers.themoviedb.org/3/people*](https://developers.themoviedb.org/3/people)

A API RESTful: <https://api.themoviedb.org> tem [limite de pedidos](https://developers.themoviedb.org/3/getting-started/request-rate-limiting). Para não ultrapassar esse limite é controlado o número de pedidos através de **RateLimiter** do Guava.

De modo a que biblioteca movlazy possa usar IO não-bloqueante, a interface IRequest tem o seu método getBody()a retornar CompletableFuture<String>, devendo as implementações desta interface estar em conformidade.

Da mesma forma o modelo de domínio, MovWebApi e MovService oferecem uma API assíncrona. Ou seja, métodos e *getters* retornam CompletableFuture<T> e CompletableFuture<Stream<T>>.

Excepção ao método search() de MovService que retorna Supplier<Stream<SearchItem>>. Neste caso, o método search() de MovService desencadeia os pedidos HTTP de todas as páginas na chamada ao search() e cujo resultado é agregado num CompletableFuture<Stream<SearchItem>> *lazy* produzida por um Supplier.

As estruturas de dados para *cache* de movies, cast e actors são do tipo ConcurrentHashMap e armazenam como valor um ComplatableFuture.

Devido à natureza reactiva do modelo de domínio o número de pedidos HTTP realizado na obtenção de um objecto poderá ser **maior** que o realizado com movlazy.

A aplicação Web usa a tecnologia [VertX](https://vertx.io/docs/vertx-web/java/#_using_vert_x_web) com *handlers* assíncronos e disponibiliza as seguintes páginas:

1. Listagem de filmes com um determinado nome recebido por *query-string*. Cada filme tem um *link* para a sua página de detalhes.
2. Detalhes de um filme incluindo um *link* para a página de listagem de créditos desse filme.
3. Listagem dos créditos de um filme. Cada crédito tem um *link* para os detalhes dessa pessoa.
4. Detalhes de uma pessoa incluindo um link para a página de listagem de filmes dessa pessoa.
5. Listagem de filmes de uma pessoa.

As páginas anteriores são acessíveis através dos seguintes caminhos (*paths*):

1. /movies?name=…
2. /movies/:id
3. /movies/:id/credits
4. /person/id
5. /person/id/movies

A aplicação web nunca bloqueia (não faz join() e nem get()) na obtenção de um resultado de um CompletableFuture.

As páginas de listagens são retornadas no corpo da reposta HTTP em modo *chunked* (response.setChunked(true)).