

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Modelação e Padrões de Desenho
2017

A biblioteca vibehub disponibiliza informação detalhada sobre concertos (Event) e pesquisa de recintos (Venue) numa cidade. Os dados são obtidos a partir de duas API RESTful distintas: <https://api.setlist.fm> e <http://www.last.fm/api>. A primeira API disponibiliza operações de pesquisa de recintos por cidade e listagem dos concertos ocorridos num dado recinto (e.g. Coliseu dos Recreios). A segunda API fornece detalhes sobre artistas (Artist) e músicas (Track). O modelo de domínio é formado pelas entidades: Venue, Event, Artist e Track e obedece à especificação apresentada no diagrama de classes da Figura 1. Neste, as propriedades identificadas por P representam *getters* (métodos sem parâmetros com prefixo get).

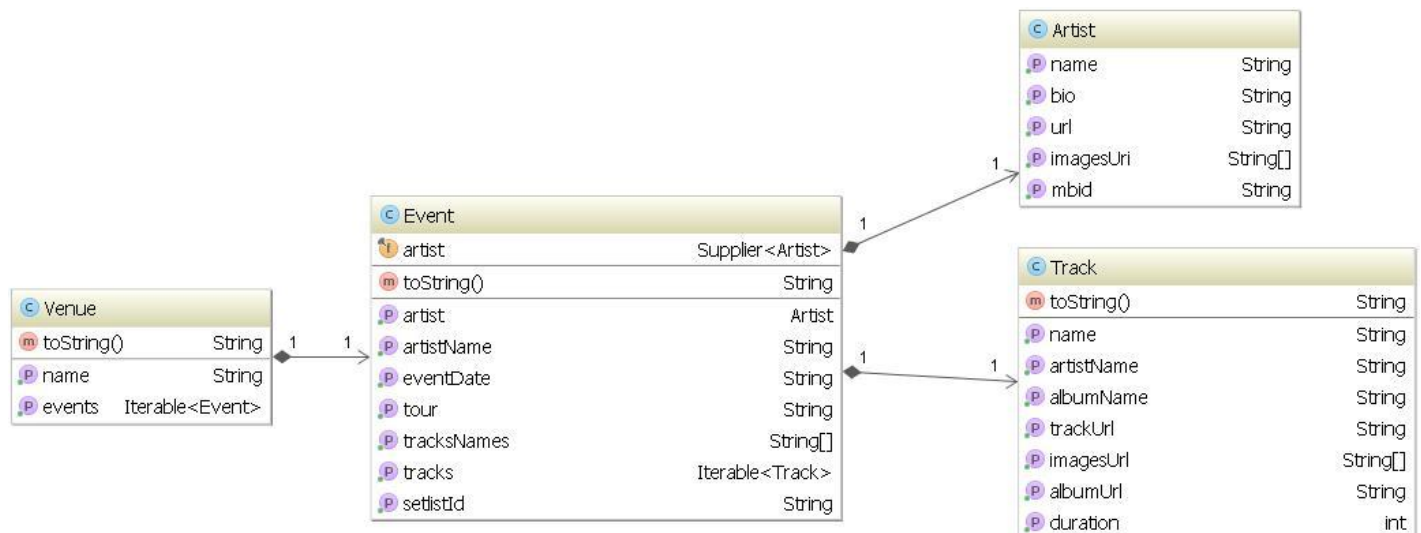
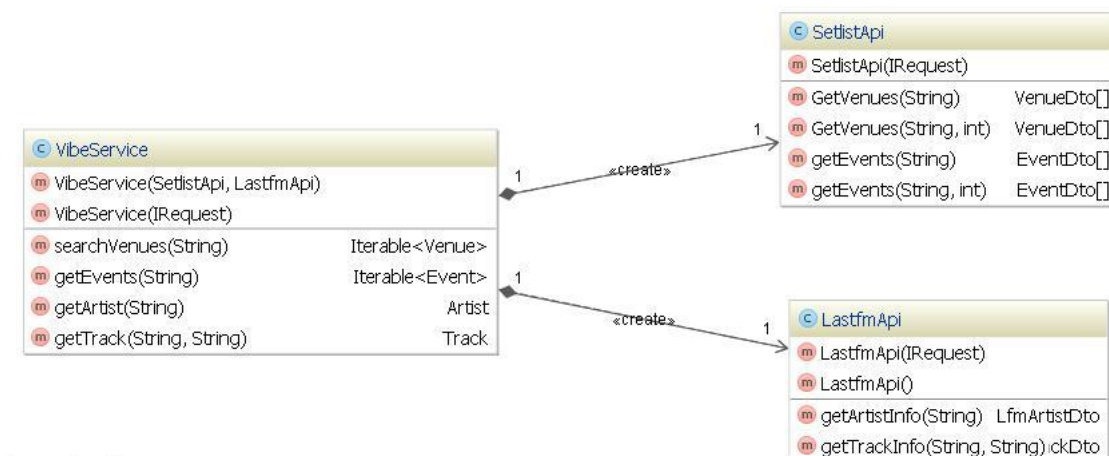


Figura 1

Todas as relações entre as entidades de domínio são implementadas de forma **lazy**. Por exemplo, a informação detalhada de um concerto (Event) só deve ser pedida à API setlist.fm apenas quando forem obtidos os elementos da sequência dada pela propriedade events de uma instância de Venue. Da mesma forma, a informação de um artista (Artist) só é pedida à API last.fm apenas quando for consultada a propriedade artist. Após haver um pedido, os próximos acessos à mesma propriedade não devem provocar novos pedidos, devendo os dados entretanto obtidos ser retornados.

Os pedidos HTTP a cada uma das APIs RESTful são mediados por uma classe específica para cada uma das APIs (SetListApi e LastFmApi). A classe VibeService é responsável pela instanciação e ligação entre as entidades de domínio.



No caso da SetListApi a informação é retornada de forma paginada, através dos métodos getVenues() e getEvents(). Nestes, o segundo parâmetro de tipo inteiro indica o número da página pretendida. Nestes casos, a

sequência retornada pelos métodos `searchVenues()` e `getEvents` de `VibeService` deve iterar de forma *lazy* sobre todas as páginas disponíveis.

Existe uma implementação de `IRequest` que mantém em cache (memória) os resultados dos pedidos HTTP que já tenham sido efectuados.

A biblioteca `vibehub` usa como fontes de dados os seguintes *endpoints*:

- <https://api.setlist.fm/docs/rest.0.1.search.venues.html> -- pesquisa de recintos numa cidade
- <https://api.setlist.fm/docs/rest.0.1.venue.venueId.setlists.html> -- lista de concertos num dado recinto
- <http://www.last.fm/api/show/artist.getInfo> -- detalhes de um artista (identificado pelo ID musicbrainz)
- <http://www.last.fm/api/show/track.getInfo> -- detalhes de uma música (identificado pelo nome da música e nome do artista)

Todos os resultados das APIs são desserializados automaticamente através biblioteca `Gson` para instâncias de classes pré definidas (DTOs), com excepção da propriedade `sets` e `set` dos dados retornados por `venue.venueId.setlists` cuja informação terá que ser obtida a partir de `JsonElement`.

A aplicação Web disponibiliza páginas para:

- Pesquisa de recintos por localidade.
- Listagem de eventos de um recinto.
- Detalhes de um evento com links para o artista e a listagem de músicas.
- Detalhes de um artista.
- Listagem de músicas de um evento.