

Carregando o Livro e evitando LazyInitializationException

Crie uma nova classe chamada `LivroDetalheBean` no pacote `br.com.casadocodigo.loja.beans`, anote a classe com `@Model` e também injete nela o `LivroDao`.

Nessa classe `LivroDetalheBean` crie os atributos `livro` do tipo `Livro` e `id` do tipo `Integer`. Crie também o método `carregaDetalhe` sem parâmetro e sem retorno. Nele vamos chamar no `livroDao` o método `buscarPorId` passando o atributo `id` que criamos a pouco e recebendo como retorno um `Livro` que vamos atribuir ao atributo `livro` que também acabamos de criar.

Seu Eclipse deve estar mostrando um erro pois o método `buscarPorId` ainda não existe no `LivroDao`, por isso, abra a classe `LivroDao` e nela, crie o método `buscarPorId` que recebe um `Integer id` e retorna um `Livro`. Nesse método, vamos fazer a **Planned Query** que vimos no vídeo:

```
String jpql = "select l from Livro l join fetch l.autores where l.id = :id";
```

Passe a *query* para o `EntityManager` e retorne um `singleResult` que será o `Livro`.

Apenas para didática, vamos colocar o `EntityManager` como `PersistenceContextType.EXTENDED` e consequentemente a classe `LivroDao` deve virar um **EJB Stateful** com `@Stateful`.

Faça *Full Publish* e teste o detalhe do livro. Tudo deve estar funcionando corretamente.