Projeto 02 - Blog Pessoal - CRUD 08

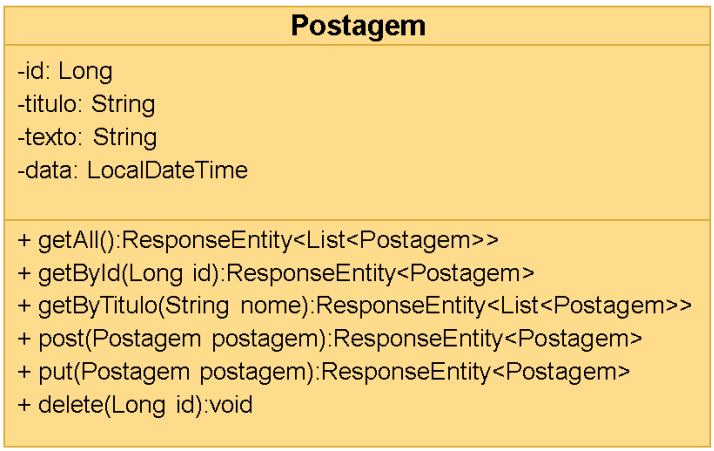
O que veremos por aqui:

1. Criar o método delete(Long id) para excluir uma Postagem no Banco de Dados

1. O Recurso Postagem

Nas etapas anteriores, construímos a Classe **PostagemController** e implementamos os Métodos:

- **getAll()** → Retorna todos os Objetos da Classe Postagem.
- **getById(Long id)** → Retorna um Objeto específico da Classe Postagem persistidos no Banco de dados. A Postagem é identificada pelo atributo id.
- **getByTitulo(String titulo)** → Retorna todos os Objetos da Classe Postagem persistidos no Banco de dados, cujo atributo titulo contenha a String enviada no parâmetro titulo do Método.
- **Método post(Postagem postagem)** → Persiste (salva) um novo Objeto da Classe Postagem no Banco de dados
- **Método put(Postagem postagem)** → Atualiza um Objeto da Classe Postagem persistido no Banco de dados.



Vamos finalizar a construção da Classe Controladora com o **Método delete(Long id)**.

🦶 Passo 01 - Criar o Método delete(Long id)

Vamos implementar o Método **delete(Long id)** na Classe Postagem Controller. Traçando um paralelo com o MySQL, seria o equivalente a instrução: `DELETE FROM tb_postagens WHERE id = id;`.

```
64
65      @ResponseStatus(HttpStatus.NO_CONTENT)
66      @DeleteMapping("/{id}")
67      public void delete(@PathVariable Long id) {
68          Optional<Postagem> postagem = postagemRepository.findById(id);
69
70          if(postagem.isEmpty())
71              throw new ResponseStatusException(HttpStatus.NOT_FOUND);
72
73          postagemRepository.deleteById(id);
74      }
75
```

Linha 65: a anotação **@ResponseStatus** indica que o Método **delete(Long id)**, terá uma Response Status específica, ou seja, quando a Resposta da Requisição for positiva, será retornado o **HTTP Status NO_CONTENT → 204**, ao invés do **HTTP Status OK → 200** como resposta padrão.

Linha 66: ** A anotação **@DeleteMapping("/{id}")** mapeia todas as Requisições **HTTP DELETE**, enviadas para um endereço específico (**Endpoint**), dentro do Recurso Postagem, para um Método específico que responderá as requisições, ou seja, ele indica que o Método **delete(Long id)**, responderá a todas as requisições do tipo **HTTP DELETE**, enviadas no endereço **http://localhost:8080/postagens/id**, onde **id** é uma **Variável de Caminho** (Path Variable), que receberá o id da Postagem que será Deletada.

Linha 67: O Método **void delete(@PathVariable Long id)** será do tipo **void** porque ele responda Requisições HTTP (HTTP Request), ao deletar uma Postagem ela deixa de existir, logo não tem nenhum tipo de retorno. Como configuramos a anotação **@ResponseStatus**, ele devolverá uma **Resposta HTTP NO_CONTENT → 204**, indicando que o Objeto deletado não existe mais. Observe que o Método possui um parâmetro do tipo **Long**, chamado **id**.

@PathVariable Long id: Esta anotação insere o valor enviado no endereço do endpoint, na Variável de Caminho **{id}**, no parâmetro do Método **delete(Long id)**;

Exemplo:

http://localhost:8080/postagens/1

Neste exemplo, o parâmetro **Long id**, do Método delete(Long id), receberá o valor 1 (Id que será procurado em tb_postagens)

ATENÇÃO: ***Por questões de boas práticas e legibilidade do código, a Variável de Caminho e o Parâmetro do Método delete devem possuir o mesmo nome.***

Linha 68: Optional<Postagem> postagem = postagemRepository.findById(id);: Cria um **Objeto Optional da Classe Postagem** chamado **postagem**, que receberá o resultado do método findById(id). Como o Método pode retornar um Objeto Nulo, utilizaremos o Optional para evitar o erro **NullPointerException**. Ao ao invés de utilizarmos o map com as Expressões Lambda, utilizaremos o Optional.

Linhas 69 e 70: Através da estrutura condicional if, checamos se o Objeto postagem está vazio (**postagem.isEmpty()**). Se estiver, geramos um HTTP Status **NOT FOUND → 404** (Não Encontrado!) e como estamos utilizando um Objeto da **Classe ResponseStatusException** (throw new ResponseStatusException(HttpStatus.NOT_FOUND);), as próximas linhas do Método serão ignoradas.

Linha 73:: Executa o Método padrão da Interface JpaRepository deleteById(Long id) e retorna o **HTTP Status NO_CONTENT → 204**, HTTP Status padrão do Método.


[Documentação: *@DeleteMapping*]


[Documentação: *@PathVariable*]


[Documentação: *ResponseEntity*]


[Documentação: *HttpStatus*]


[Documentação: *.deleteById()*]


[Documentação: *Optional*]

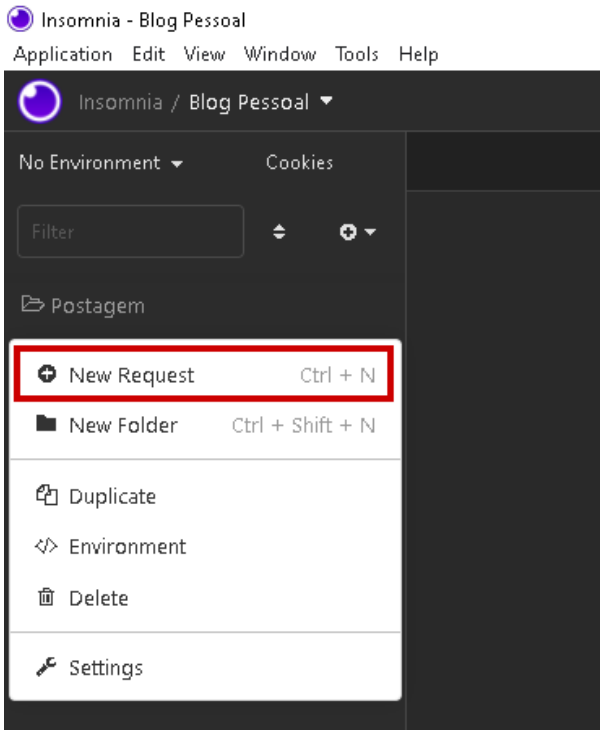
Para concluir, não esqueça de Salvar o código (**File → Save All**) .


[Código fonte do projeto]

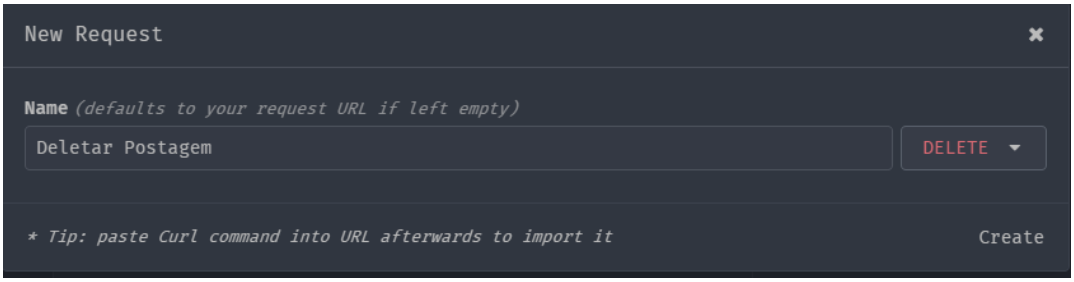
 Passo 02 - Testar no Insomnia

Agora vamos criar a Requisição para o **delete(Long id)**:

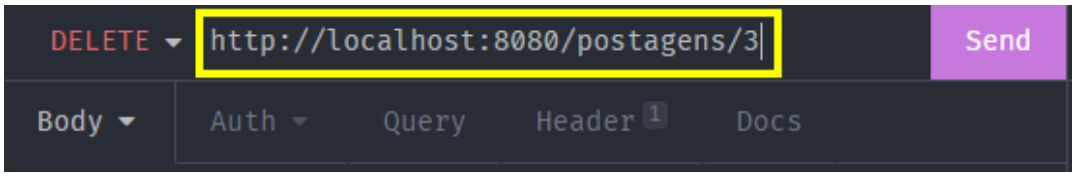
1. Clique com o botão direito do mouse sobre a **Pasta Postagem** para abrir o menu e clique na opção **New Request**.

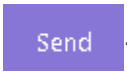


2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**DELETE**). Clique no botão **Create** para concluir.

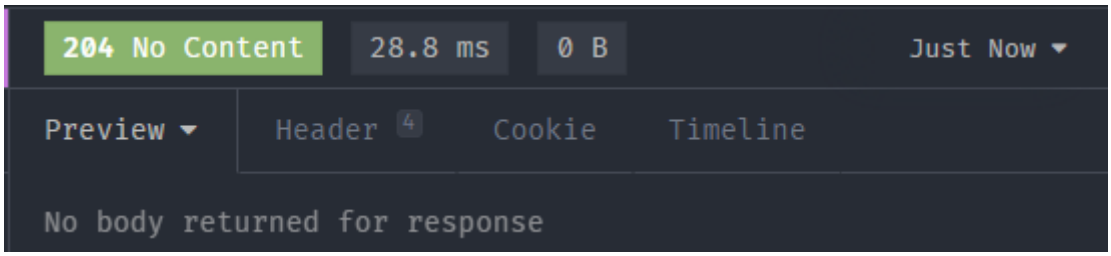


3. Configure a requisição conforme a imagem abaixo:



4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint)) da Requisição. A requisição **Deletar Postagem** foi configurada da seguinte maneira:
    - A primeira parte do endereço (http://localhost:8080) é o endereço do nosso servidor local. Quando a API estiver na nuvem, ele será substituído pelo endereço da aplicação na nuvem.
    - A segunda parte do endereço é o **endpoint** configurado na anotação ***@RequestMapping***, em nosso caso **/postagens/**.
    - A terceira parte (**/2**) é a varável de caminho (**@PathVariable**) id. Informe o id da postagem que você deseja apagar.
5. Para testar a requisição, com a aplicação rodando, clique no botão 

.
6. O resultado da requisição você confere na imagem abaixo:



7. Observe que a aplicação retorna apenas um **HTTP Status 204 → NO_CONTENT** (indicado em verde na imagem acima). Este Status indica que a Requisição foi bem sucedida!, o Objeto foi apagado e o seu conteúdo não existe mais.
8. Caso o Objeto não seja encontrado, a aplicação retornará o **HTTP Status 404 → NOT FOUND** (Não encontrado), como mostra a figura abaixo (marcado em laranja).