

Postman

O que é o POSTMAN?

O Postman é um API Client que facilita a vida da pessoa desenvolvedora no processo de desenvolvimento do Backend de uma API. O Postman permite criar, compartilhar, testar e documentar APIs de forma simples e prática. Isso é feito, permitindo aos usuários criar e salvar requisições **HTTP e HTTPS** (Request) desde as mais simples até as mais complexas, bem como ler suas respostas (Response).

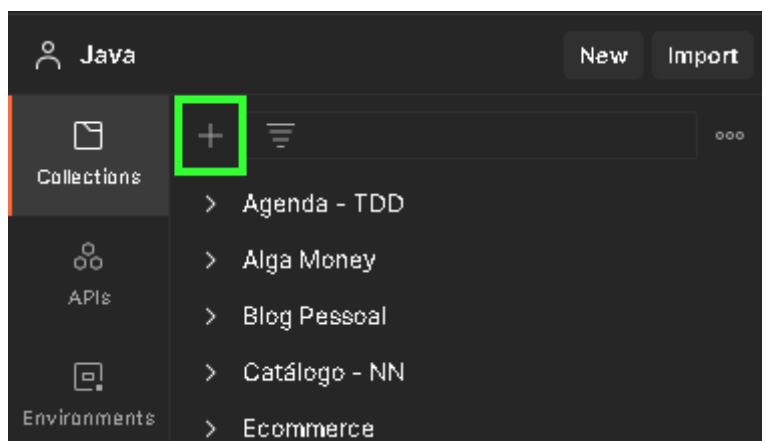
Por que utilizar o POSTMAN?

Além da praticidade de ter todos os exemplos e códigos de integração prontos, o POSTMAN é a ferramenta de testes mais utilizadas pelas equipes de desenvolvimento das grandes empresas. Dessa maneira ao realizar integrações, caso você possua duvidas, será mais rápido e simples de confirmar o que pode estar ocorrendo com o seu código.

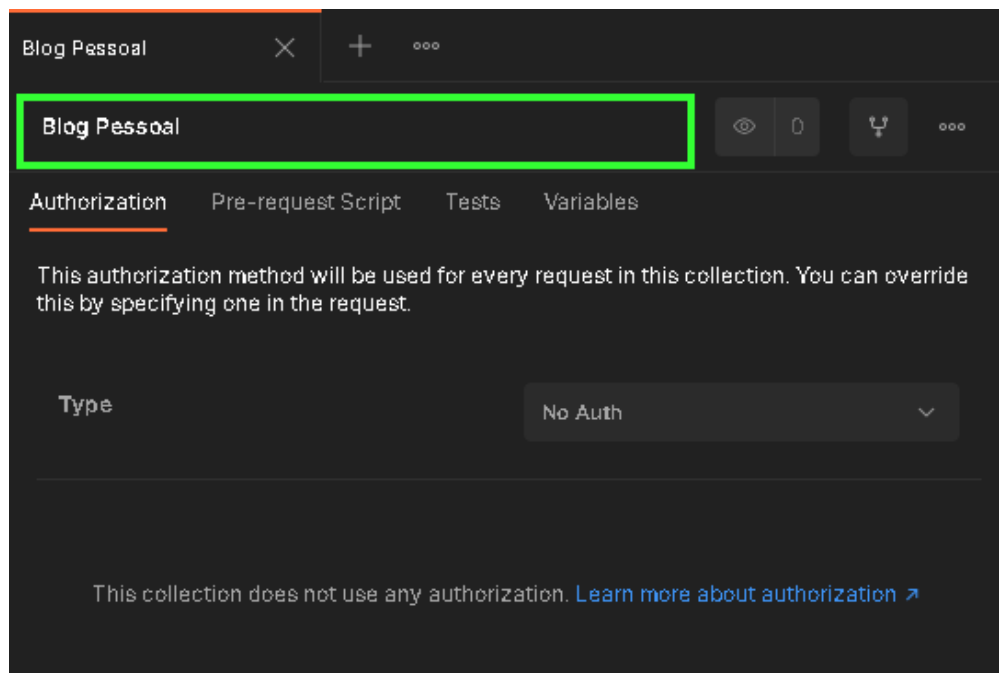
Este tutorial partirá do princípio que você já criou a conta no Postman, Instalou e Configurou o aplicativo no seu computador. Caso você não tenha feito estes passos, consulte o Guia de Instalação enviado pela Generation.

Criando uma Collection

1. Em Collections, clique no botão +

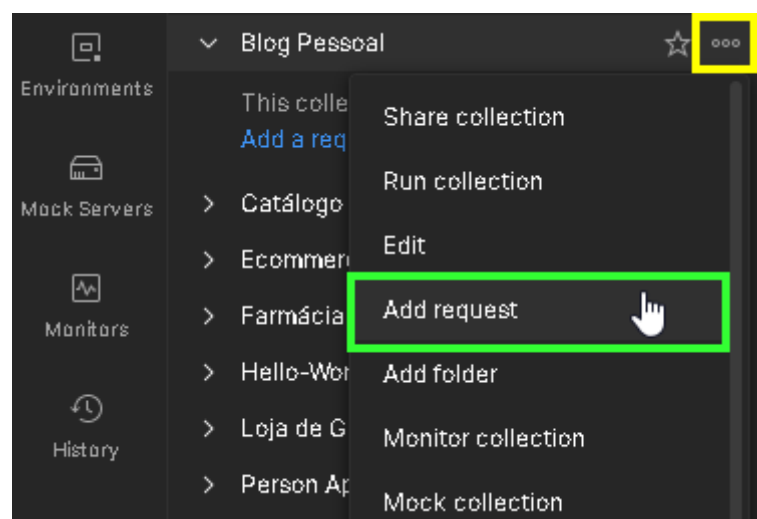


2. Na janela que será aberta, informe o nome da Collection e pressione a tecla enter para concluir

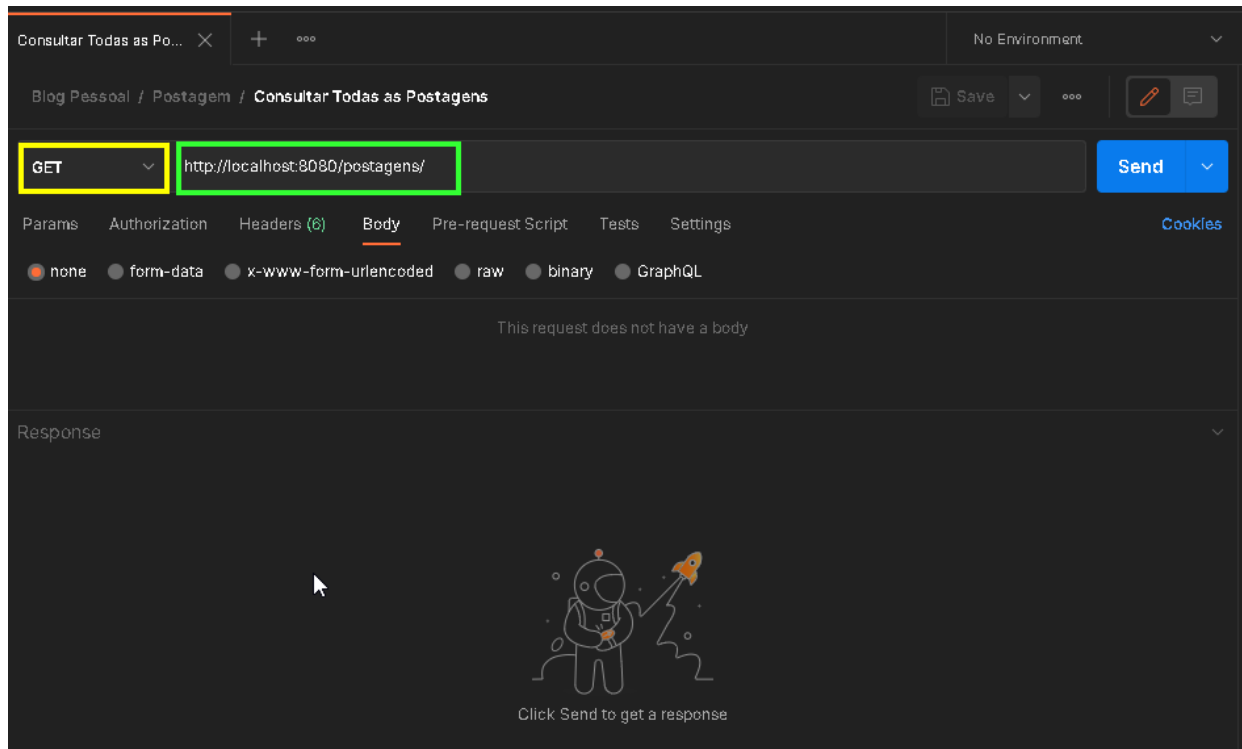


Criando a Request - Consultar todas as postagens - findAll()

1. Esta consulta retornará uma lista com todas as postagens salvas no Banco de dados.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).
3. Clique na opção **Add request** (indicado em verde na imagem abaixo)



4. Configure a requisição conforme a imagem abaixo:

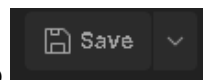


5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição. Como estamos fazendo uma consulta, utilize o verbo **GET**.

6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Consultar todas as postagens** 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**.

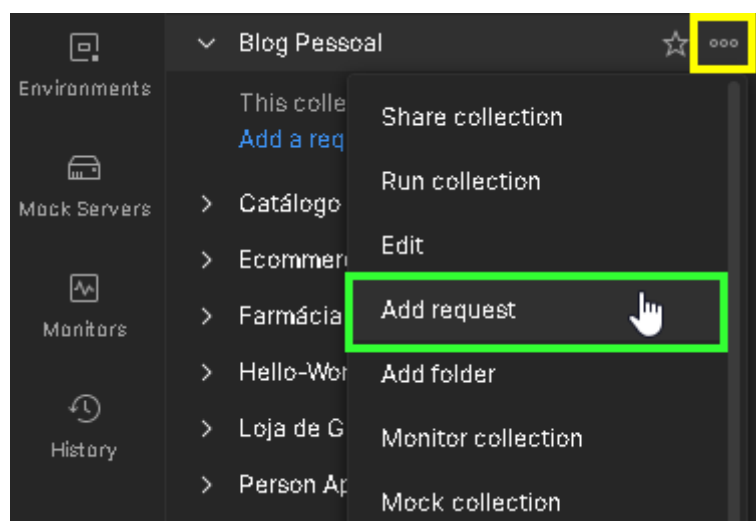
7. Salve a requisição clicando no botão



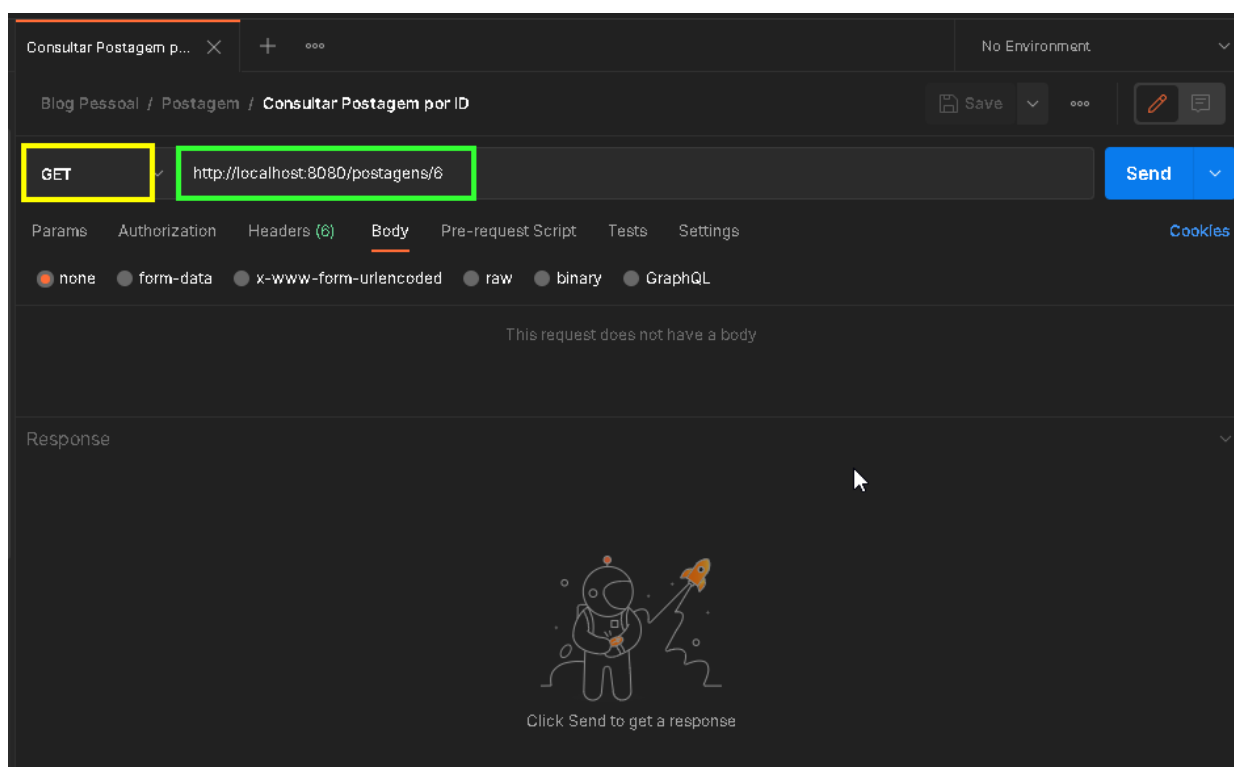
8. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.

Criando a Request - Consultar Postagem por ID - findById(id)

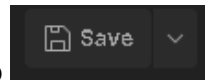
1. Esta consulta retornará uma única postagem salva no Banco de dados, conforme o Id informado.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).
3. Clique na opção **Add request** (indicado em verde na imagem abaixo)

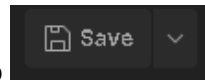


4. Configure a requisição conforme a imagem abaixo:



5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição. Como estamos fazendo uma consulta, utilize o verbo **GET**.
6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Consultar postagem por ID** 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 (/6) é a variável de caminho (**Path variable**) id. Informe o id que você deseja procurar.

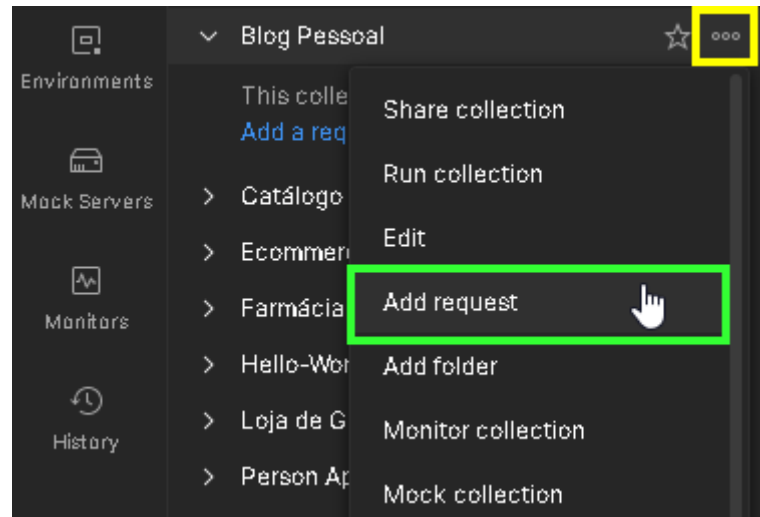


7. Salve a requisição clicando no botão .
8. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.

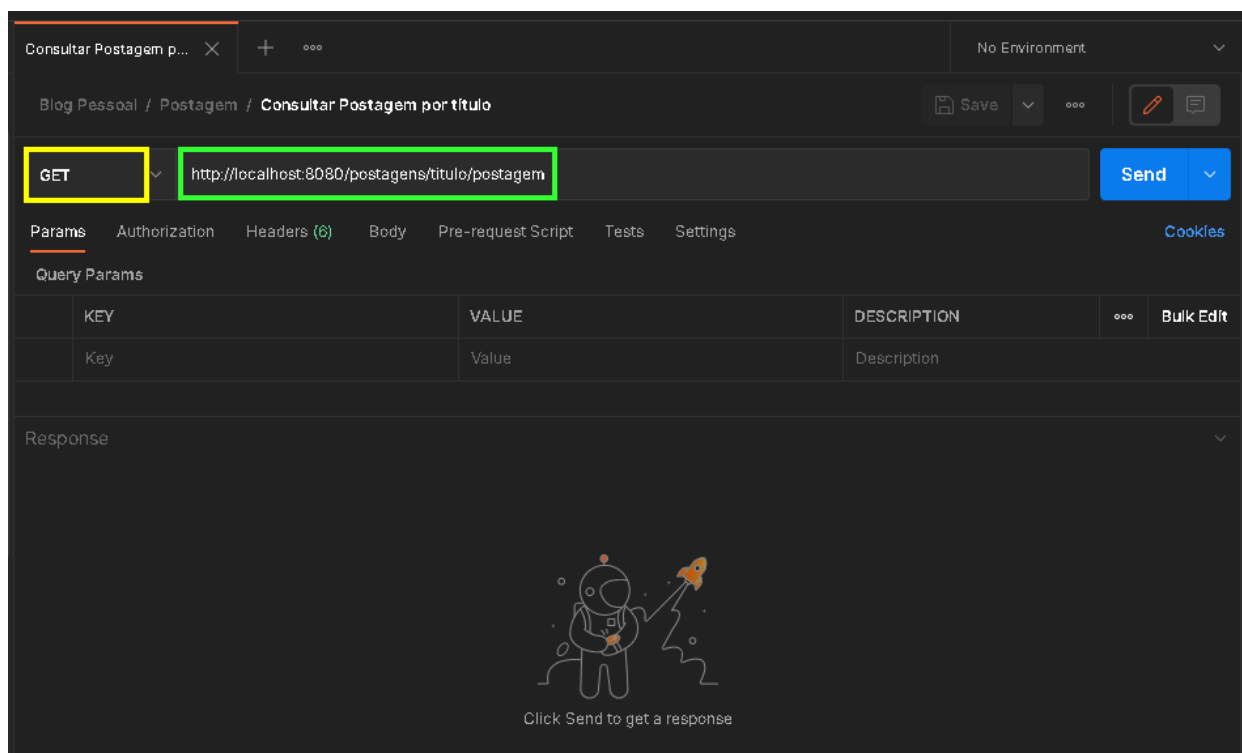
Criando a Request - Consultar Postagem por Título - `findByTituloContainingIgnoreCase(titulo)`

1. Esta consulta retornará uma lista com todas as postagens salvas no Banco de dados, que possuam no atributo Título o texto digitado na variável de caminho.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).

3. Clique na opção **Add request** (indicado em verde na imagem abaixo)



4. Configure a requisição conforme a imagem abaixo:

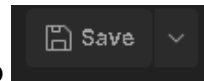


5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição. Como estamos fazendo uma consulta, utilize o verbo **GET**.

6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Consultar postagem por Título** 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 (/titulo/postagem), titulo é apenas um indicativo do conteúdo da variável que deverá ser preenchida. postagem é o conteúdo da variável de caminho (**Path variable**) titulo. Informe o texto que você deseja pesquisar no atributo titulo da tabela.

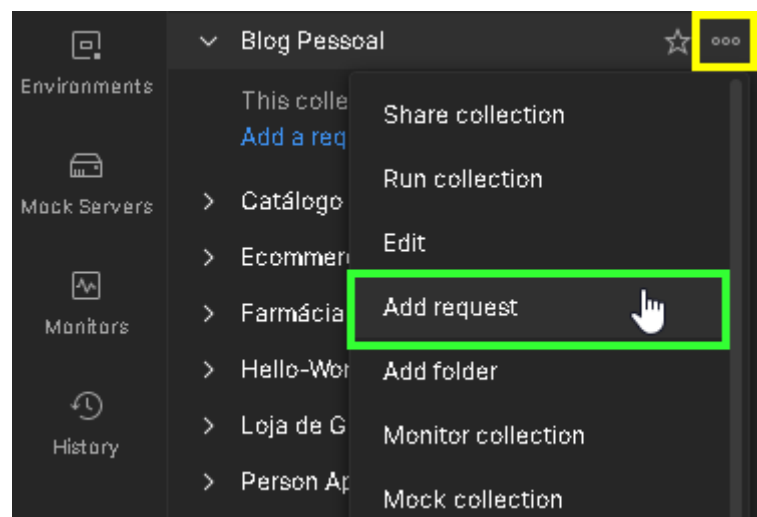


7. Salve a requisição clicando no botão

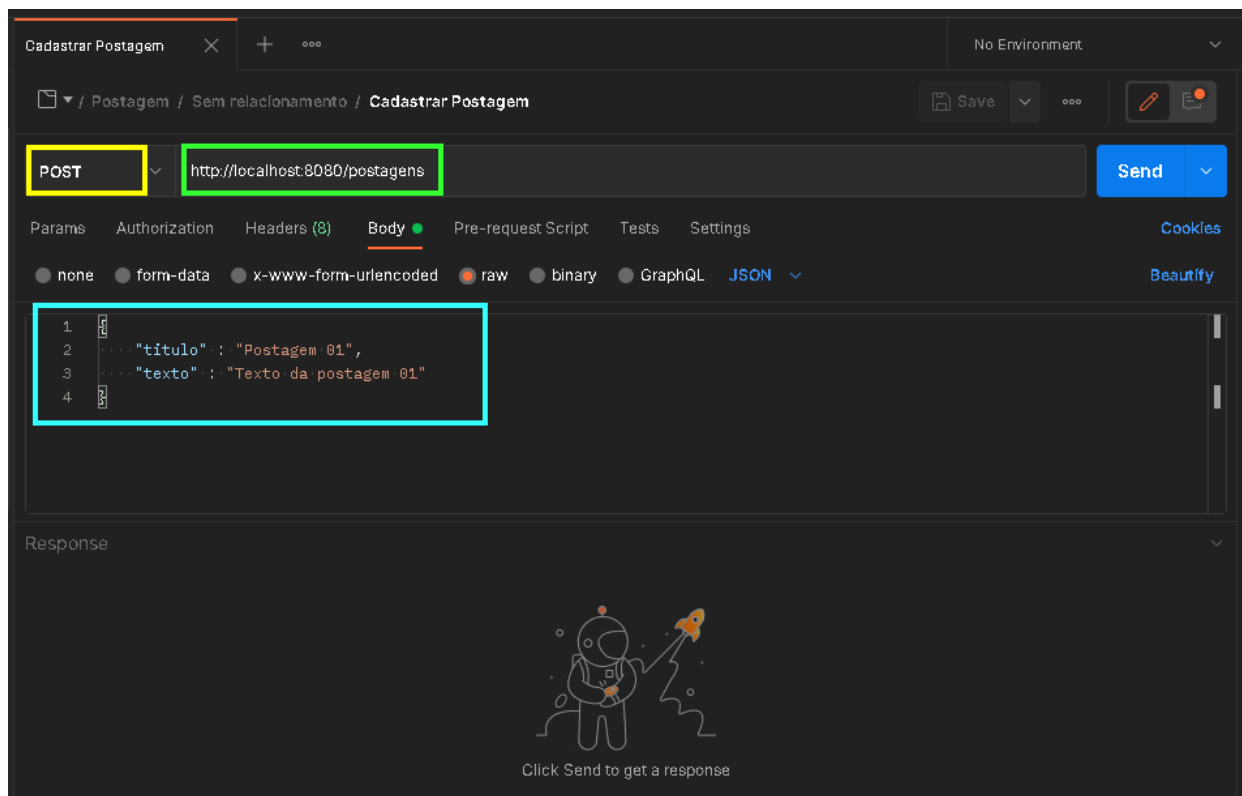
8. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.

Criando a Request - Cadastrar Postagem

1. Esta requisição irá salvar (criar) uma nova postagem no Banco de dados.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).
3. Clique na opção **Add request** (indicado em verde na imagem abaixo)



4. Configure a requisição conforme a imagem abaixo:



5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição.

Como estamos fazendo uma inserção de dados no Banco de dados, utilize o verbo **POST**.

6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Cadastrar 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**.

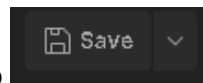
7. No item marcado em azul, precisamos inserir um JSON com os dados que serão inseridos na nova postagem. O texto antes dos 2 pontos (:) é o atributo da classe. O texto depois dos 2 pontos (:) é o dado que será cadastrado.

8. Observe que no método **POST** não é necessário enviar o atributo **id** no JSON.

9. Caso a classe **Postagem** esteja com o relacionamento **habilitado**, você deverá acrescentar no JSON o objeto da classe relacionada. Veja o exemplo abaixo:

```
1  {
2    "titulo": "Spring Teste 2",
3    "texto": "Spring Teste 2",
4    "tema": {
5      "id": 1
6    }
7  }
```

10. No item marcado em verde na imagem acima, observe que está sendo passado dentro do JSON um objeto da classe Tema chamado tema, identificado apenas pelo atributo id.

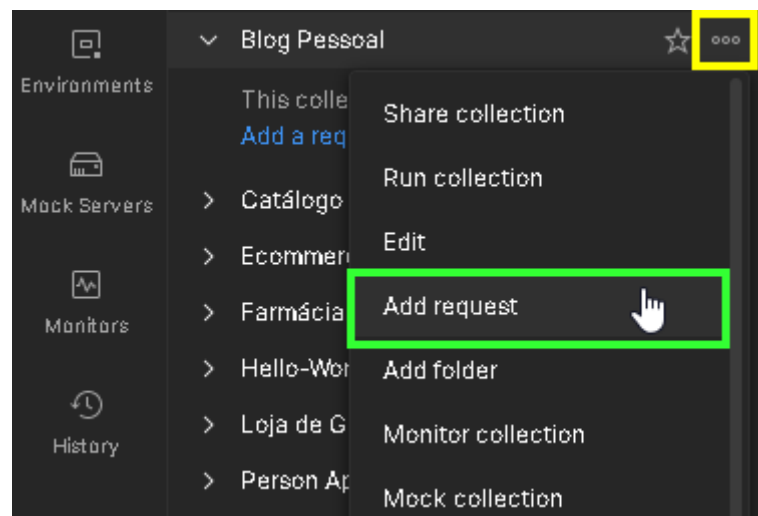


11. Salve a requisição clicando no botão

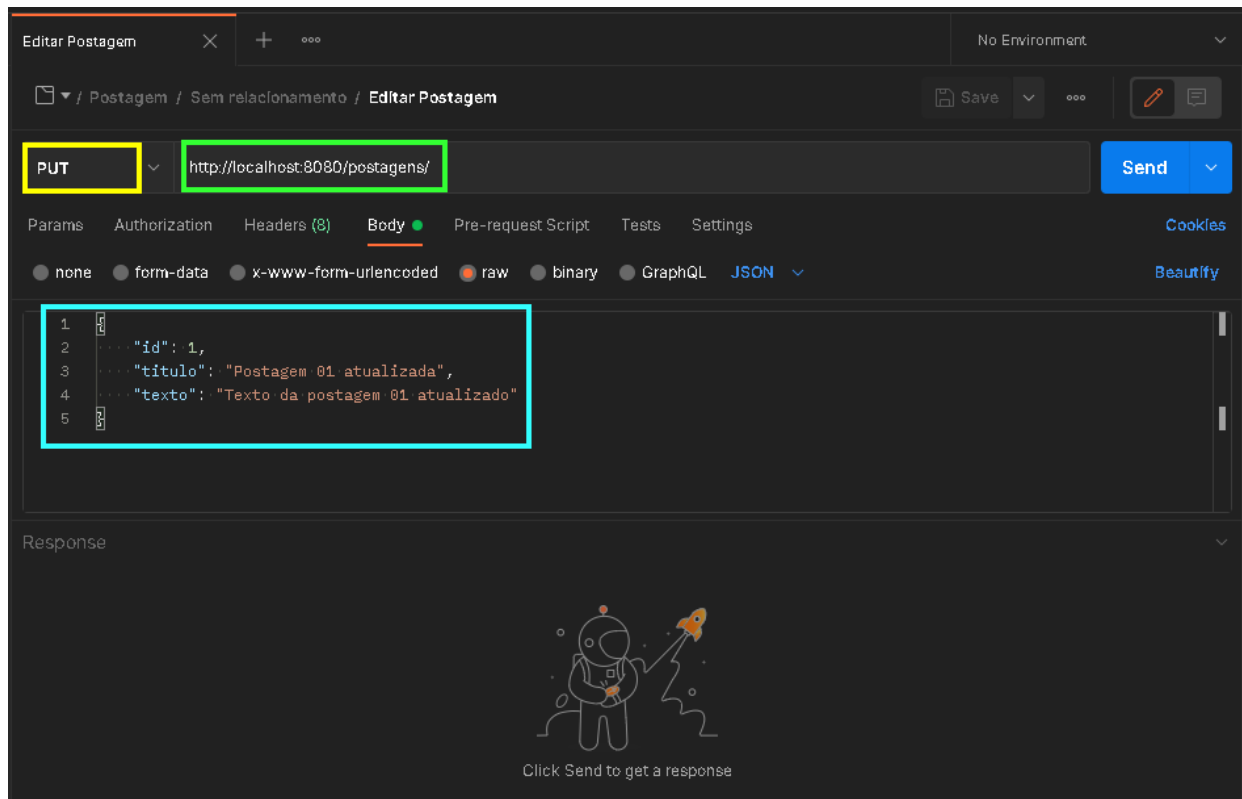
12. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.

Criando a Request - Editar Postagem

1. Esta requisição irá editar (atualizar) uma nova postagem no Banco de dados.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).
3. Clique na opção **Add request** (indicado em verde na imagem abaixo)



4. Configure a requisição conforme a imagem abaixo:



5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição. Como estamos fazendo uma inserção de dados no Banco de dados, utilize o verbo **PUT**.

6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Editar 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**.

7. No item marcado em azul, precisamos inserir um JSON com os dados que serão atualizados na postagem. O texto antes dos 2 pontos (:) é o atributo da classe. O texto depois dos 2 pontos (:) é o dado que será cadastrado.

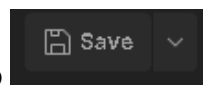
8. Observe que no método **PUT** é necessário enviar o atributo **id** no JSON para identificar a postagem que será atualizada.

9. Caso a classe **Postagem** esteja com o relacionamento **habilitado**, você deverá acrescentar no JSON o objeto da classe relacionada. Veja o exemplo abaixo:

```
1  {
2    "id": "1",
3    "titulo": "Minha Postagem",
4    "texto": "Texto atualizado",
5    "tema": {
6      "id": 1
7    }
8  }
```

10. No item marcado em verde na imagem acima, observe que está sendo passado dentro do JSON um objeto da classe Tema chamado tema, identificado apenas pelo atributo id.

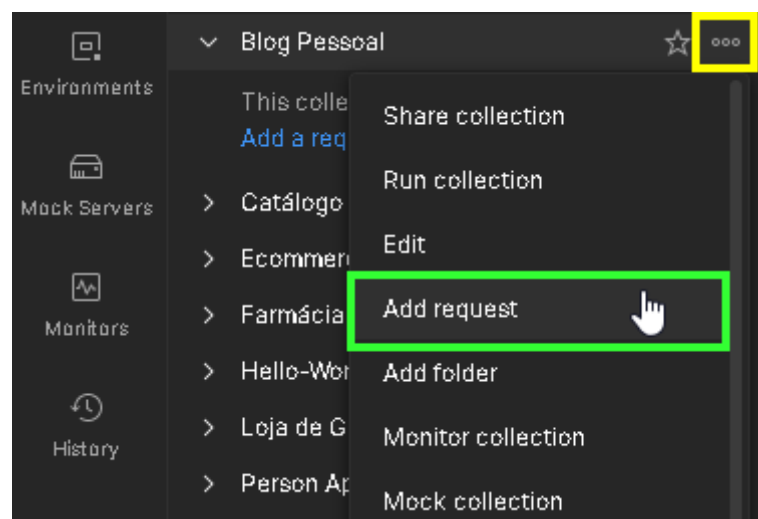
11. Salve a requisição clicando no botão



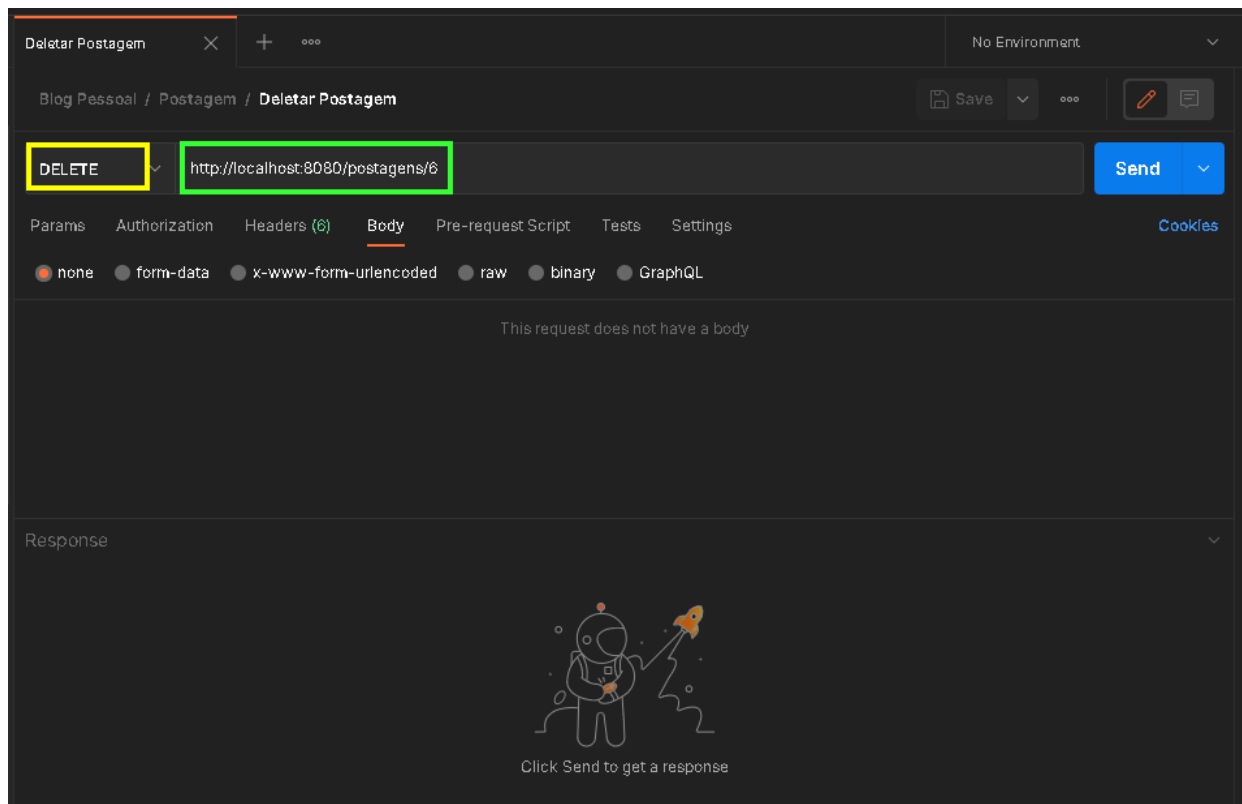
12. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.

Criando a Request - Deletar Postagem

1. Esta apagará uma única postagem salva no Banco de dados, conforme o Id informado.
2. Na Collection criada, clique nos 3 pontinhos ao lado do nome da Collection para abrir o menu (indicado em amarelo na imagem abaixo).
3. Clique na opção **Add request** (indicado em verde na imagem abaixo)



4. Configure a requisição conforme a imagem abaixo:



5. No item marcado em amarelo na imagem acima, informe o verbo da Requisição. Como estamos fazendo uma consulta, utilize o verbo **DELETE**.

6. No item marcado em verde na imagem acima, informe o destino (caminho) da Requisição. A requisição **Consultar postagem por ID** 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 (/6) é a variável de caminho (**Path variable**) **id**. Informe o id da postagem que você deseja apagar.

7. Salve a requisição clicando no botão .

8. Para testar a requisição, com a aplicação rodando, clique no botão **Send**.