

Criando Requisições HTTP no Insomnia

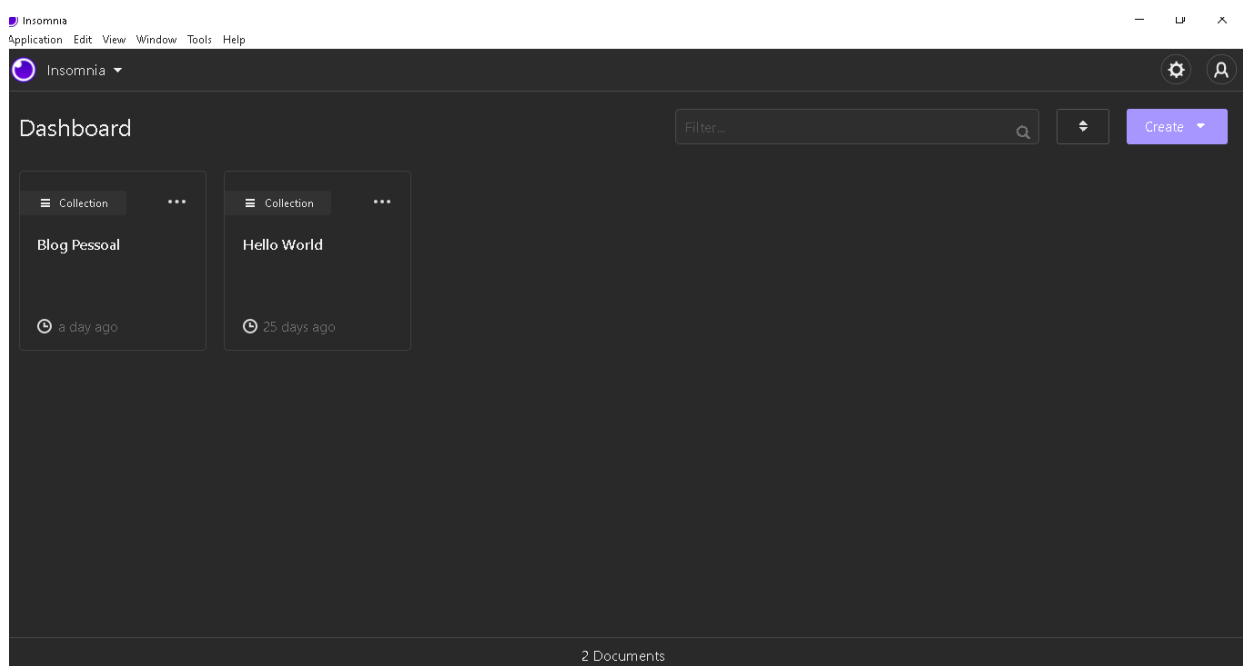
O que é o Insomnia?

O Insomnia é um API Client que facilita a vida da pessoa desenvolvedora no processo de desenvolvimento do Backend de uma API. Ele permite criar, compartilhar, testar e documentar API's 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 Insomnia?

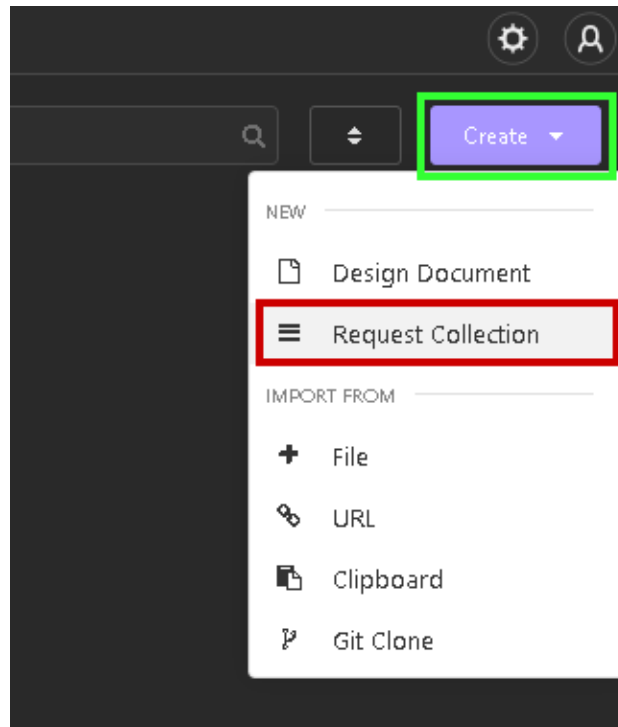
Além da praticidade de ter todos os exemplos e códigos de integração prontos, o Insomnia é uma das ferramentas de testes mais utilizadas pelas equipes de desenvolvimento das grandes empresas. Dessa maneira ao realizar integrações, caso você possua dúvidas, será mais rápido e simples de confirmar o que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

Este tutorial partirá do princípio que você já instalou o aplicativo no seu computador. Caso você não tenha feito estes passos, consulte o Guia de Instalação do Insomnia.



Criando uma Collection

1. Na janela principal do Insomnia, clique no botão **Create** e clique na opção **Request Collection**.



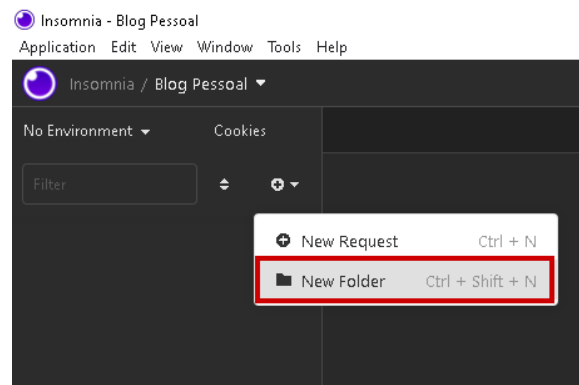
2. Na janela que será aberta, informe o nome da Collection (**Blog Pessoal**) e clique no botão **Create** para concluir.



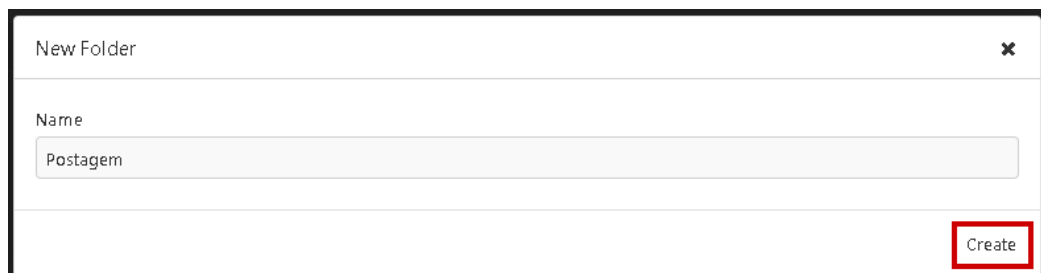
Criando a Pasta Postagem

Vamos criar dentro da **Collection Blog Pessoal** a **Pasta Postagem**, que guardará todas as requisições do **Recurso Postagem**.

1. Na **Collection Blog Pessoal**, clique no botão **+**. No menu que será aberto, clique na opção **New Folder**.

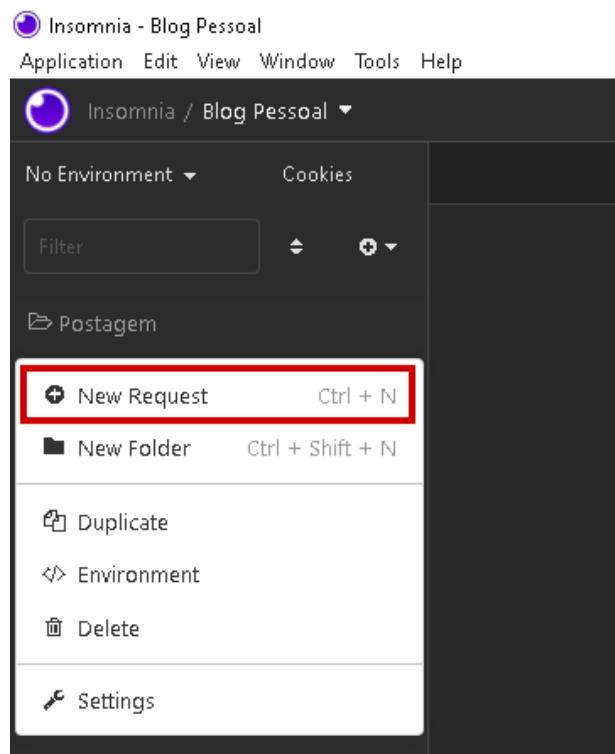


2. Na janela que será aberta, informe o nome da pasta (**Postagem**) e clique no botão **Create** para concluir.



Criando a Request - Consultar todas as postagens

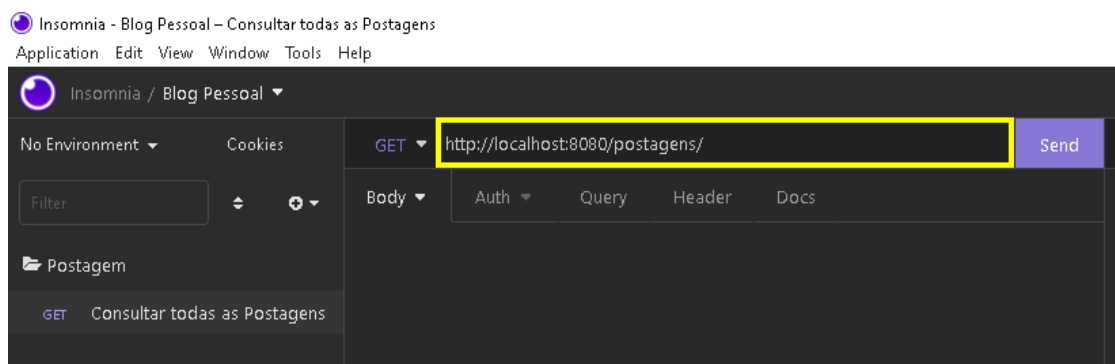
1. Clique com o botão direito do mouse sobre a **Pasta Postagem** para abrir o menu e clique na opção **New Request**.



2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**GET**), indicado na imagem na cor azul. Clique no botão **Create** para concluir.



3. Configure a requisição conforme a imagem abaixo:

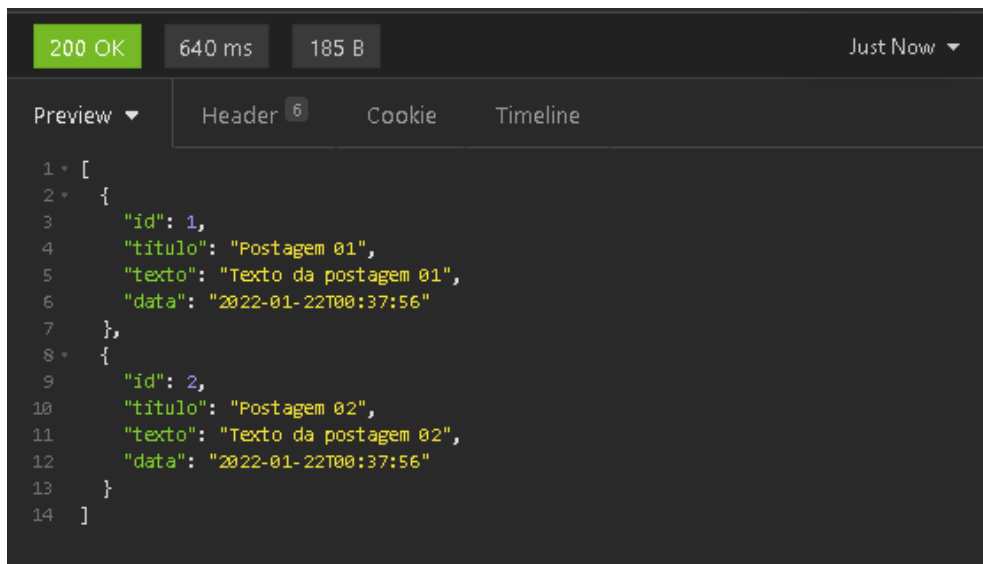


4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint) da Requisição. A requisição **Consultar Todas as postagens** foi configurada da seguinte maneira:

- 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 (<http://nomedaaplicacao.herokuapp.com>).
- A segunda parte do endereço é o **endpoint** configurado na anotação **@RequestMapping**, em nosso caso **/postagens**.

5. Para testar a requisição, com a aplicação rodando, clique no botão .

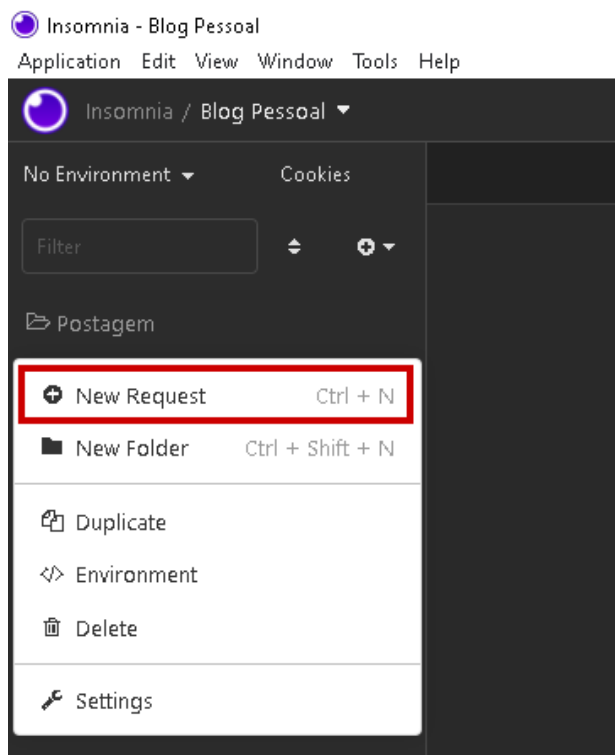
6. O resultado da requisição você confere na imagem abaixo:



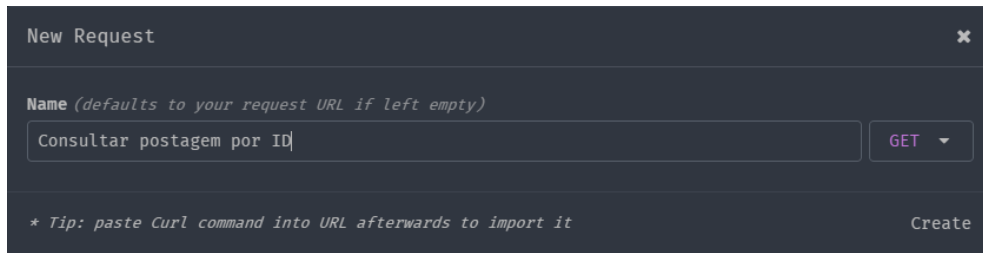
7. Observe que a aplicação além de exibir os dados de todos os Objetos da Classe Postagem persistidos no Banco de dados, no Corpo da Resposta, ela também retornará um **HTTP Status 200 → OK** (indicado em verde na imagem acima), informando que a Requisição foi bem sucedida!

Criando a Request - Consultar Postagem por 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 (**GET**). Clique no botão **Create** para concluir.



New Request

Name (defaults to your request URL if left empty)

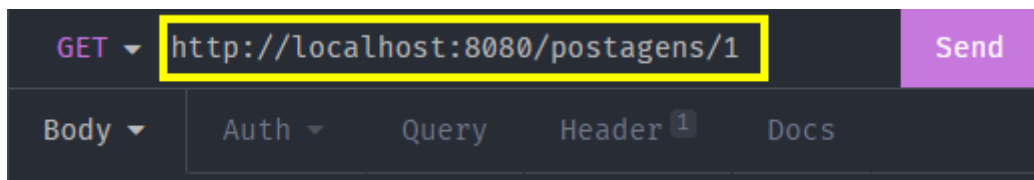
Consultar postagem por ID

GET

* Tip: paste Curl command into URL afterwards to import it

Create

3. Configure a requisição conforme a imagem abaixo:



GET

http://localhost:8080/postagens/1

Send

Body

Auth

Query

Header 1

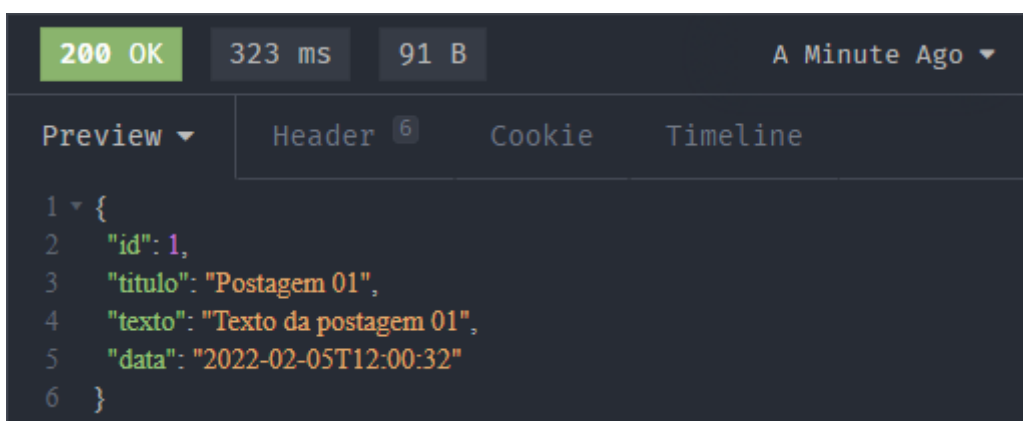
Docs

4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint) 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 (<http://nomedaaplicacao.herokuapp.com>).
- A segunda parte do endereço é o **Endpoint** configurado na anotação **@RequestMapping**, em nosso caso **/postagens**.
- A terceira parte (**/1**) é a variável de caminho (**@PathVariable**) **id**. Informe o id que você deseja procurar.

5. Para testar a requisição, com a aplicação rodando, clique no botão .

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



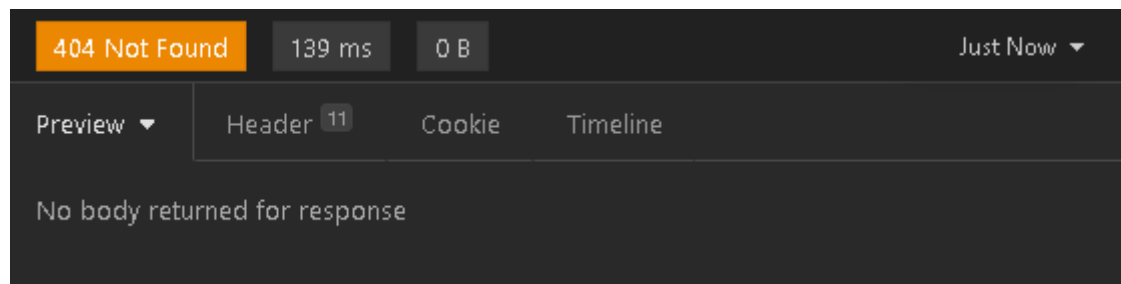
200 OK 323 ms 91 B A Minute Ago

Preview

Header 6 Cookie Timeline

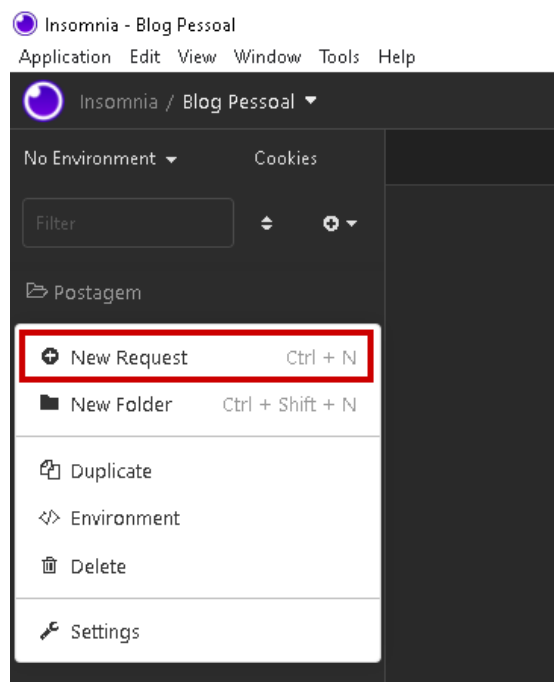
```
1 {
2   "id": 1,
3   "titulo": "Postagem 01",
4   "texto": "Texto da postagem 01",
5   "data": "2022-02-05T12:00:32"
6 }
```

7. Observe que a aplicação quando encontra o Objeto no Banco de dados, além de exibir os dados do Objeto no Corpo da Resposta, respeitando o critério informado na consulta (id 1), ela também retorna um **HTTP Status 200 → OK** (indicado em verde na imagem acima), informando que a Requisição foi bem sucedida!
8. Caso o Objeto Postagem não seja encontrado, a aplicação retornará o **HTTP Status 404 → NOT FOUND** (Não encontrado), marcado em laranja na imagem abaixo:

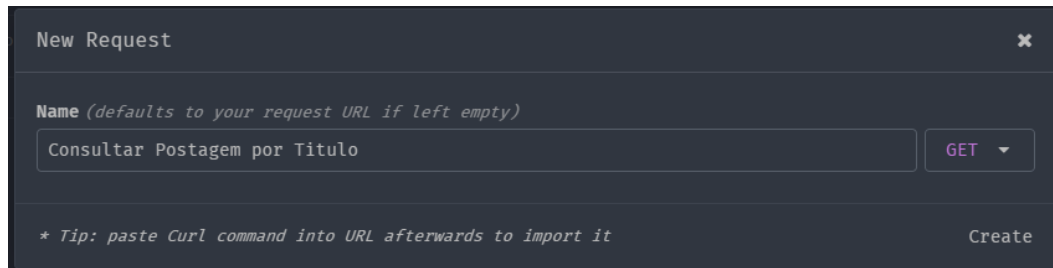


Criando a Request - Consultar Postagem por Título

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 (**GET**). Clique no botão **Create** para concluir.



New Request

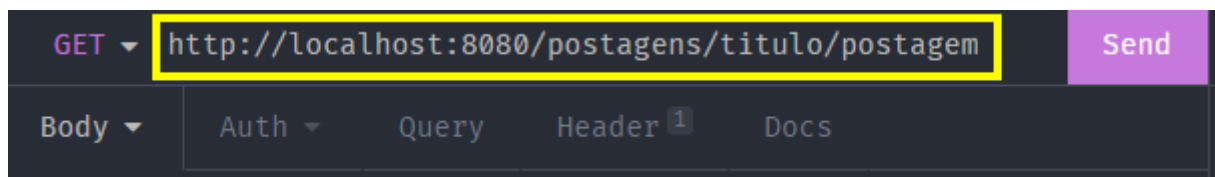
Name (defaults to your request URL if left empty)

Consultar Postagem por Titulo GET

* Tip: paste Curl command into URL afterwards to import it

Create

3. Configure a requisição conforme a imagem abaixo:



GET http://localhost:8080/postagens/titulo/postagem Send

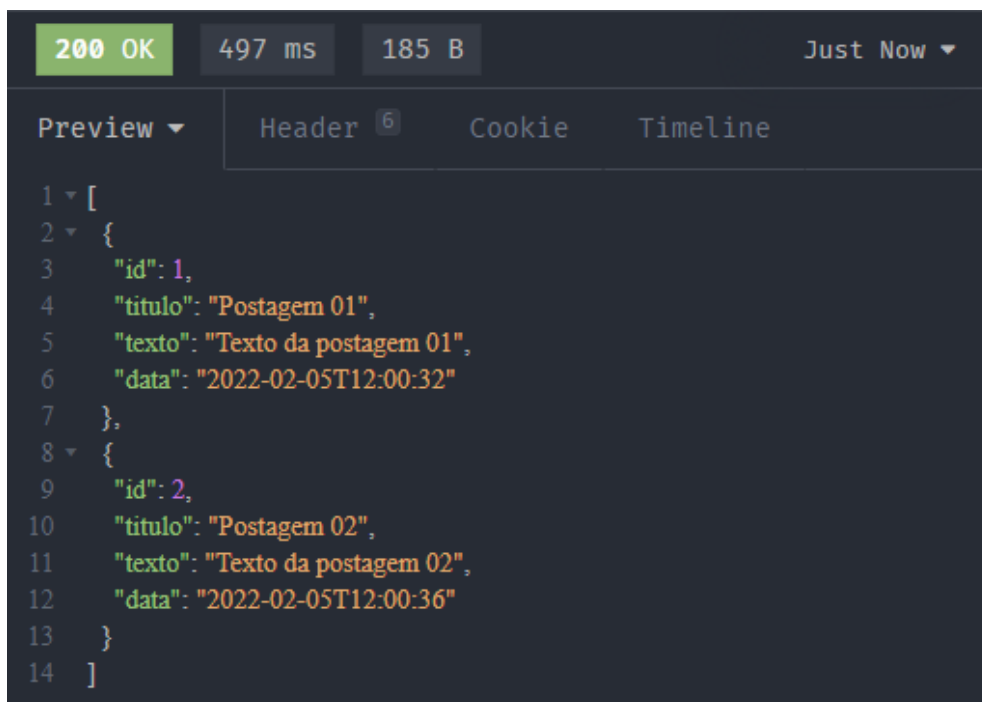
Body Auth Query Header 1 Docs

4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint) da Requisição. A requisição **Consultar Postagem por Titulo** 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 (<http://nomedaaplicacao.herokuapp.com>).
- 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. A palavra postagem é o conteúdo da variável de caminho (**@PathVariable**) titulo. Informe o texto que você deseja pesquisar.

5. Para testar a requisição, com a aplicação rodando, clique no botão .

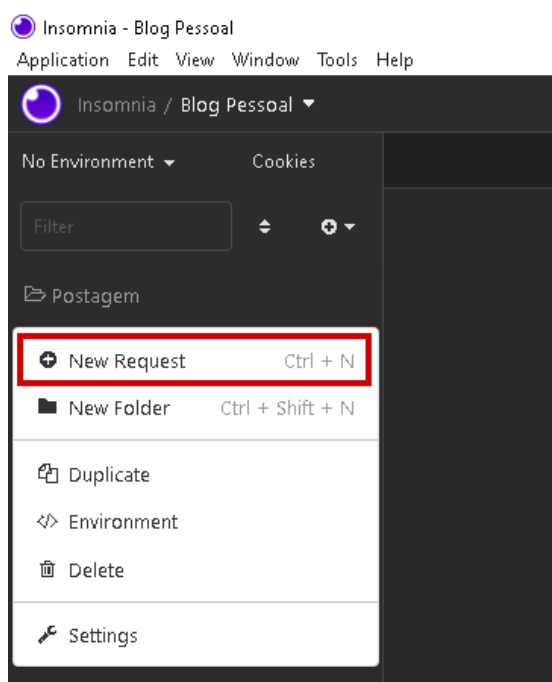
6. O resultado da requisição você confere na imagem abaixo:



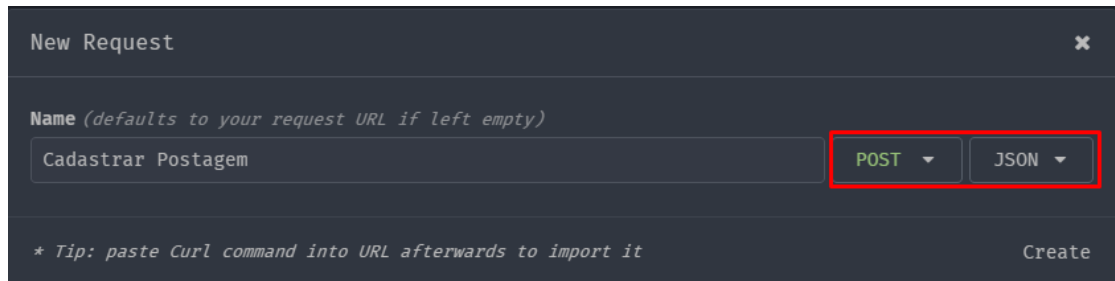
7. Observe que além de exibir no corpo da resposta todos os Objetos da Classe Postagem, respeitando o critério informado na consulta (palavra postagem), ela também retornará um **HTTP Status 200 → OK** (indicado em verde na imagem acima), informando que a Requisição foi bem sucedida!

Criando a Request - Cadastrar Postagem

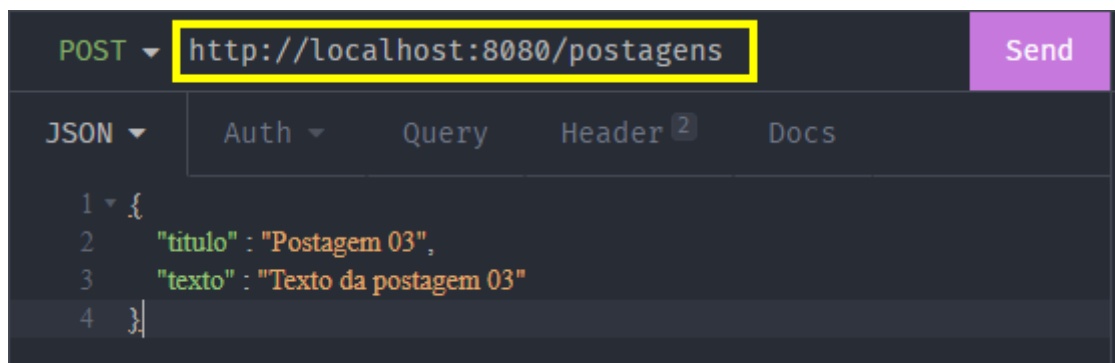
1. Clique com o botão direito do mouse sobre a **Pasta Postagem** para abrir o menu e clique na opção **New Request**.



2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**POST**). Depois de selecionar o Método **POST**, observe que será necessário informar o formato em que os dados serão enviados através do Corpo da Requisição. Selecione o formato **JSON**, como indicado na imagem abaixo em vermelho e clique no botão **Create** para concluir.



3. Configure a requisição conforme a imagem abaixo:



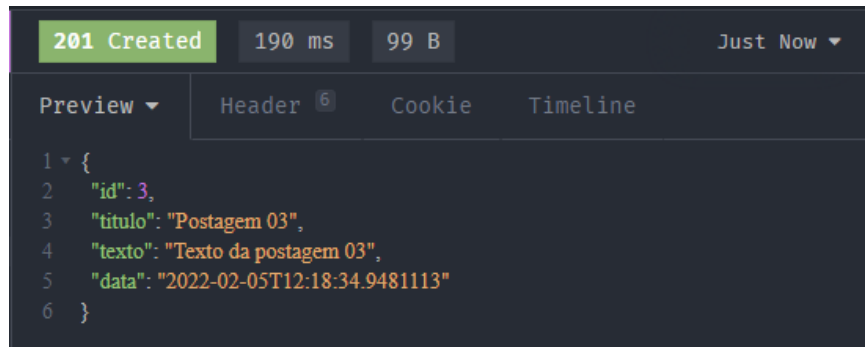
4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint) da Requisição. A requisição **Cadastrar Postagem** foi configurada da seguinte maneira:
- A primeira parte do endereço (<http://localhost:8080>) é o endereço do nosso servidor local. Quando a API estiver na nuvem, ele será substituído pelo endereço da aplicação na nuvem.
 - A segunda parte do endereço é o **endpoint** configurado na anotação **@RequestMapping**, em nosso caso **/postagens**.
5. Na guia **JSON**, precisamos inserir um **JSON** com os dados que serão inseridos na nova postagem. Lembrando que no padrão JSON: **o texto antes dos 2 pontos (:) é o atributo** da Classe e **o texto depois dos 2 pontos (:) é o dado** que será cadastrado no atributo. Os atributos são separados por vírgula, como mostra a imagem acima.



ATENÇÃO: Observe que no método **POST** não é necessário enviar os atributos **id** (será gerado pelo Banco de dados) e **data** (será enviada pelo Sistema Operacional) no **JSON**.

6. Para testar a requisição, com a aplicação rodando, clique no botão .

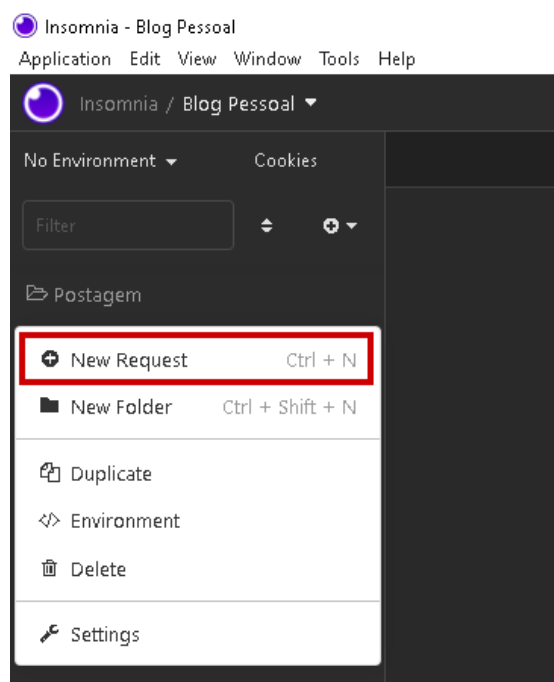
7. O resultado da requisição você confere na imagem abaixo:



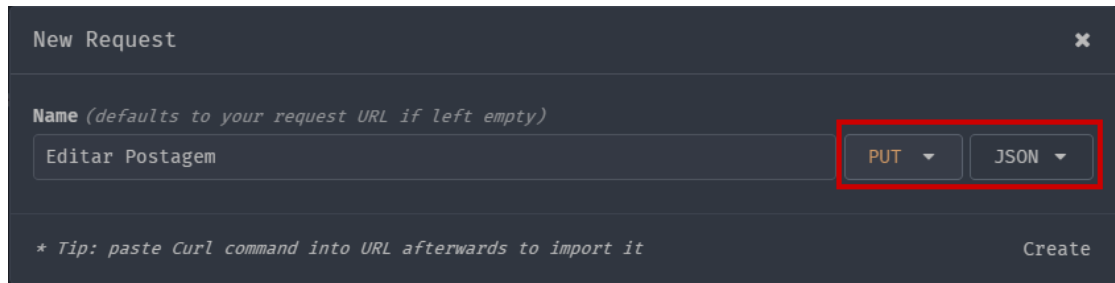
8. Observe que além do Objeto Postagem que foi persistido no Banco de dados com o **id** e a **data**, ela também retorna um **HTTP Status 201 → CREATED** (indicado em verde na imagem acima). Este Status indica que a Requisição foi bem sucedida!

Criando a Request - Editar Postagem

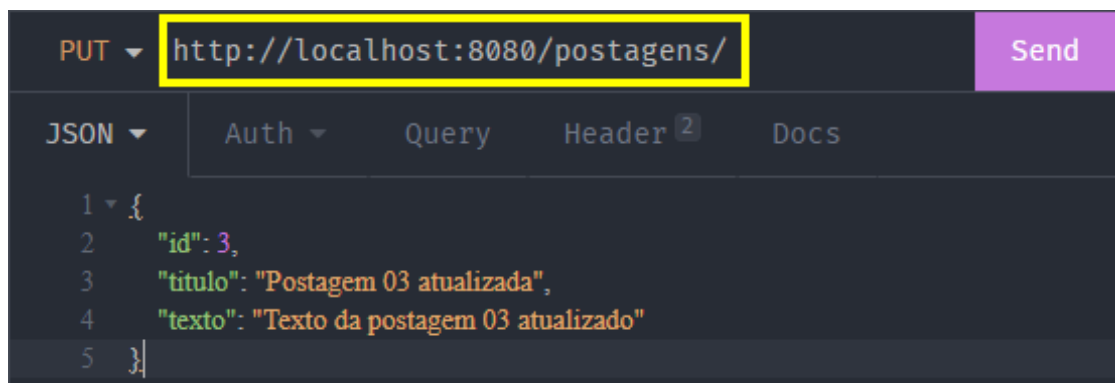
1. Clique com o botão direito do mouse sobre a **Pasta Postagem** para abrir o menu e clique na opção **New Request**.



2. Na janela que será aberta, informe o nome da requisição e o Método HTTP que será utilizado (**PUT**). Depois de selecionar o Método **PUT**, observe que será necessário informar o formato em que os dados serão enviados através do Corpo da Requisição. Selecione o formato **JSON**, como indicado na imagem abaixo em vermelho e clique no botão **Create** para concluir.




3. Configure a requisição conforme a imagem abaixo:



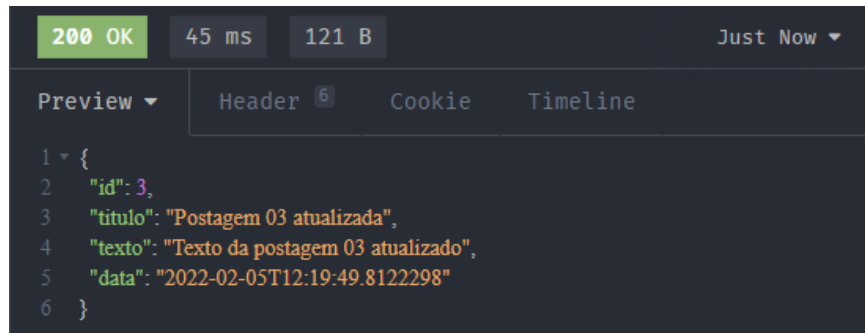
4. No item marcado em amarelo na imagem acima, informe o endereço (endpoint) da Requisição. A requisição **Editar Postagem** foi configurada da seguinte maneira:
- A primeira parte do endereço (<http://localhost:8080>) é o endereço do nosso servidor local. Quando a API estiver na nuvem, ele será substituído pelo endereço da aplicação na nuvem.
 - A segunda parte do endereço é o **endpoint** configurado na anotação **@RequestMapping**, em nosso caso **/postagens**.
5. Na guia **JSON**, precisamos inserir um **JSON** com os dados que serão inseridos na nova postagem. Lembrando que no padrão JSON: **o texto antes dos 2 pontos (:)** é o **atributo** da Classe e **o texto depois dos 2 pontos (:)** é o **dado** que será cadastrado no atributo. Os atributos são separados por virgula, como mostra a imagem acima.



ATENÇÃO: Observe que no método PUT é necessário enviar o atributo **id** no JSON para identificar a Postagem que será atualizada. A data não precisa ser enviada.

6. Para testar a requisição, com a aplicação rodando, clique no botão .

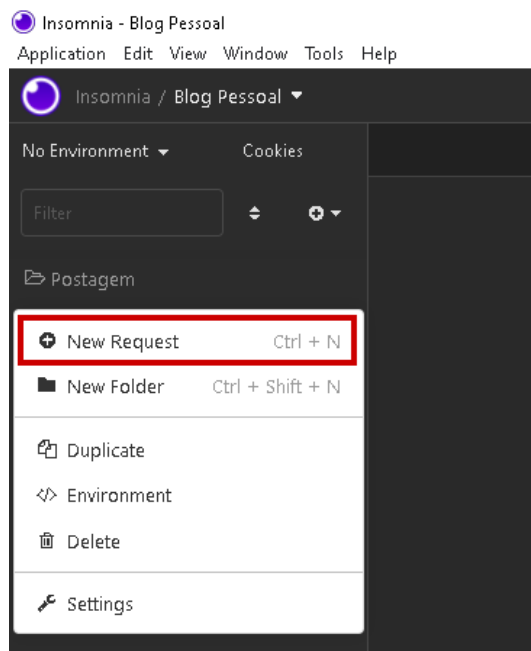
7. O resultado da requisição você confere na imagem abaixo:



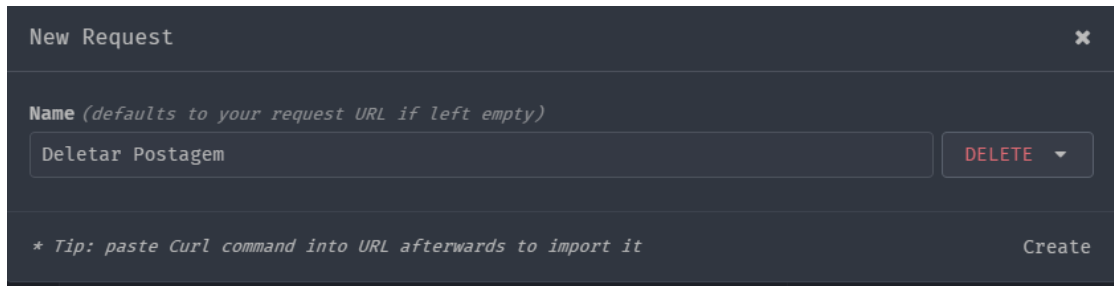
8. Observe que além de retornar o Objeto Postagem atualizado, também será retornado um **HTTP Status 200 → OK** (indicado em verde na imagem acima). Este Status indica que a Requisição foi bem sucedida!

Criando a Request - Deletar Postagem

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.



New Request

Name (defaults to your request URL if left empty)

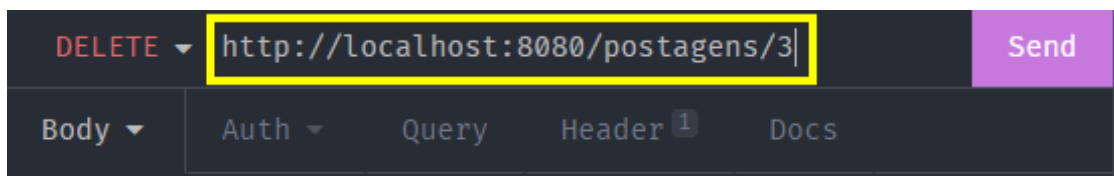
Deletar Postagem

DELETE

* Tip: paste Curl command into URL afterwards to import it

Create

3. Configure a requisição conforme a imagem abaixo:



DELETE

http://localhost:8080/postagens/3

Send

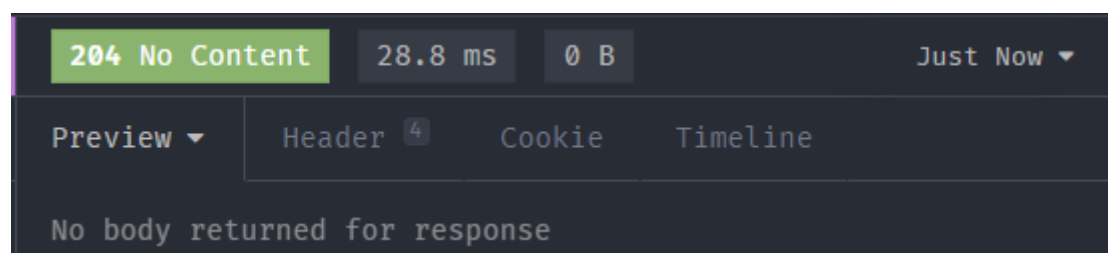
Body Auth Query Header 1 Docs

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 variável 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:



204 No Content

28.8 ms

0 B

Just Now

Preview Header 4 Cookie Timeline

No body returned for response

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).

