

CAPÍTULO 4

Usando o Postman

Índice

4.1 O que é o Postman e para que serve	3
4.2 Criando Coleções no Postman	3
4.3 Criando Requisições	4
4.3.1 Criando requisições GET normais	6
4.3.2 Criando requisição GET com PathVariable	7
4.3.3 Criando requisições GET com QueryParams	8
4.3.4 Fazendo POST com JSON Body	9
4.3.5 Criando uma requisição DELETE usando PathVariable	10
4.3.6 Criando uma requisição PUT para um registro	11

4.1 O que é o Postman e para que serve

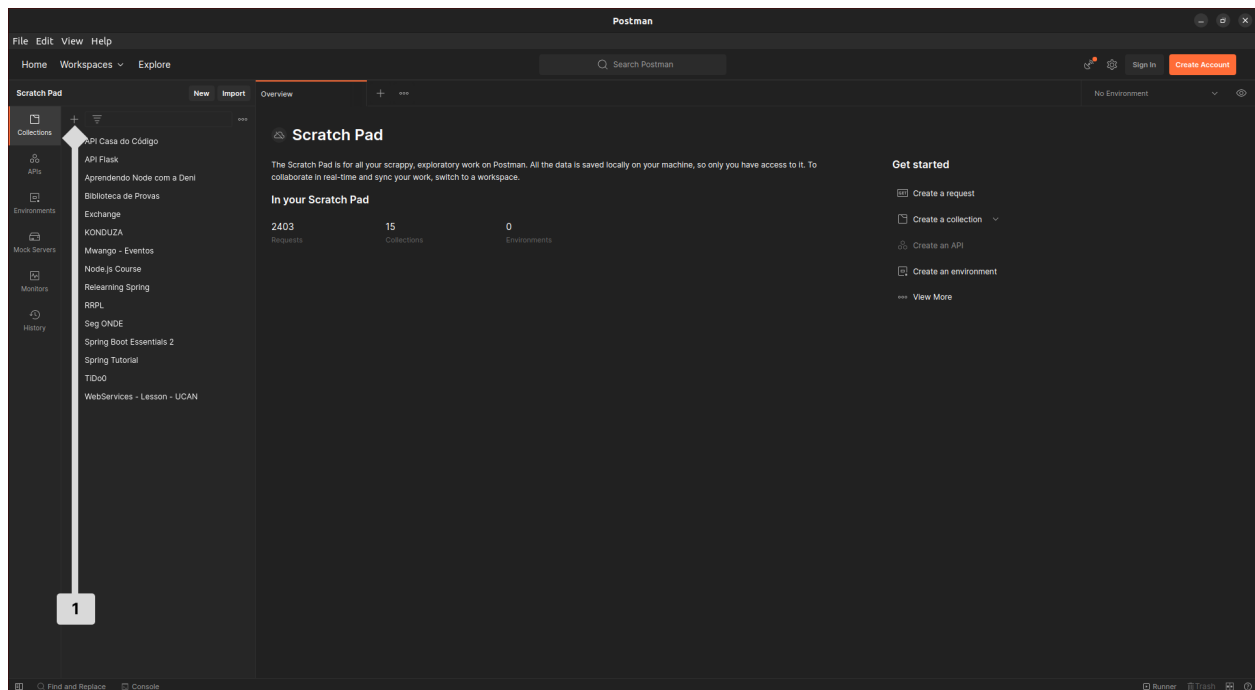
O Postman é um software robusto com diversas funcionalidades mas cujo a sua principal funcionalidade é ser o um cliente HTTP de nossos web services, em outras palavras é o software que nos permite testar os nossos endpoints da nossa API.

4.2 Criando Coleções no Postman

Um das primeiras coisas que fazemos quando abrimos o Postman é criar uma coleção, uma coleção basicamente se resume em um local onde podemos colocar todas as requisições HTTP, como se fosse uma pasta.

Aconselha-se que para cada projecto que seja necessário testar com o Postman uma coleção individual.

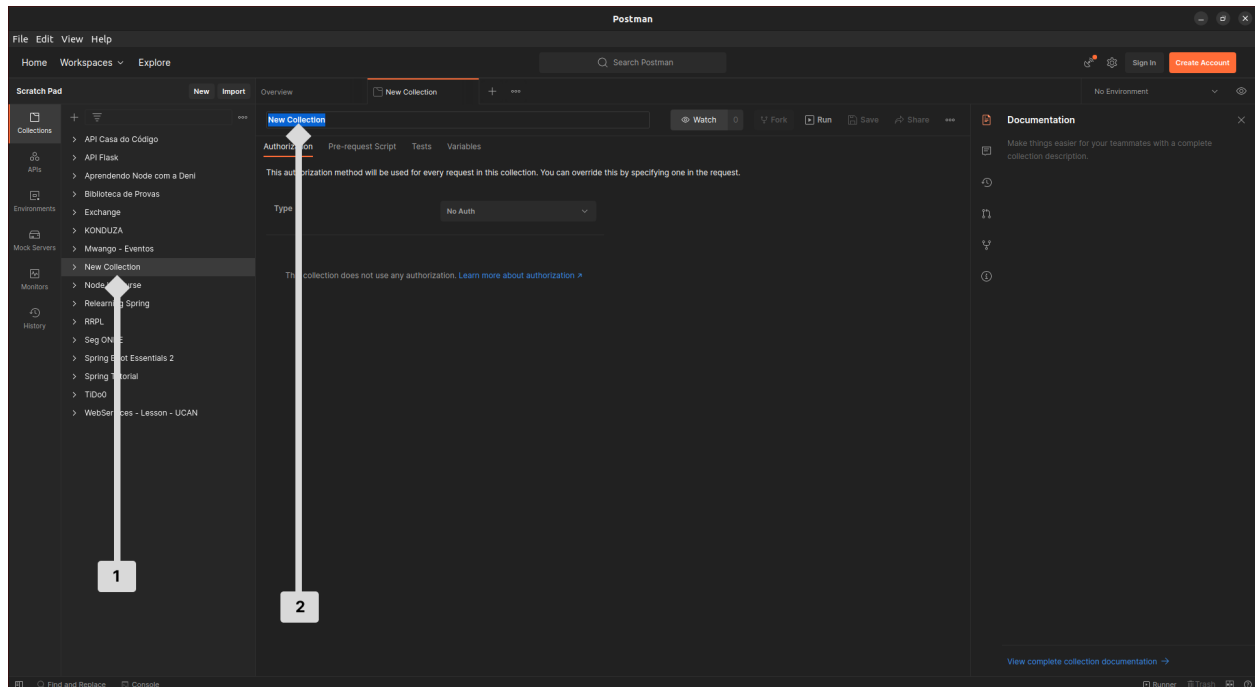
Na imagem abaixo apresentamos temos Postman e apresentamos como podemos criar uma coleção.



Descrição:

- 1 - Botão para criação de coleções quando clicado.

Após criar uma coleção temos o seguinte resultado

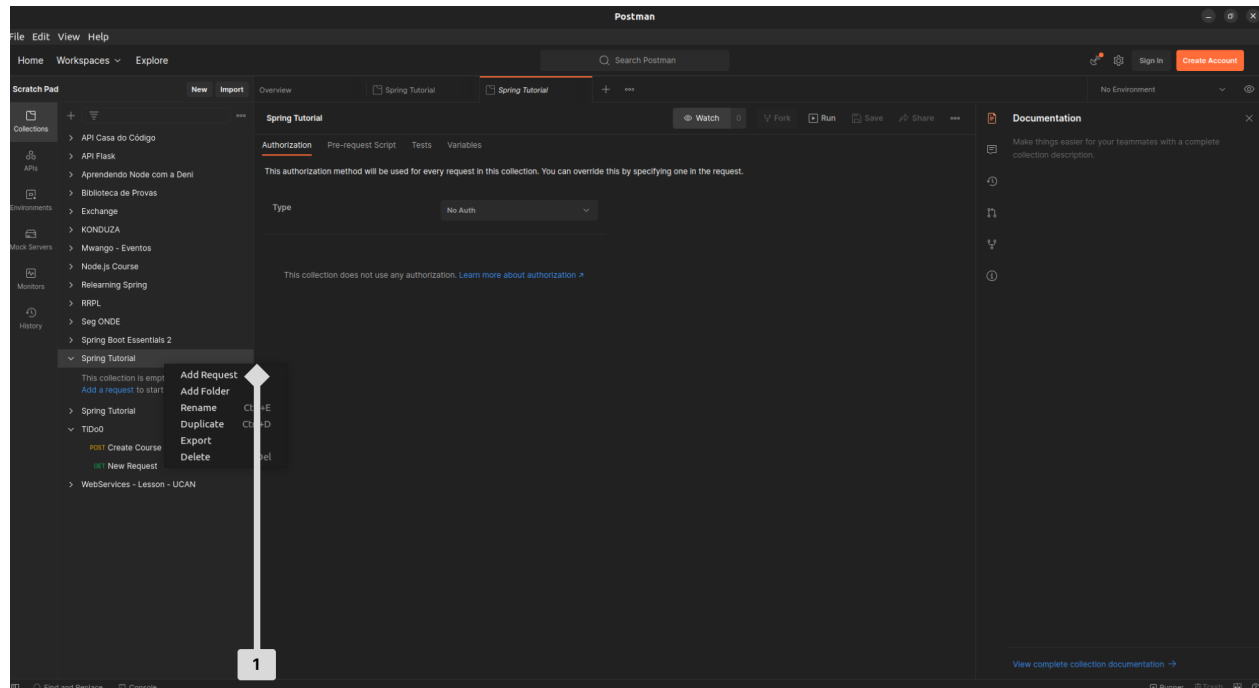


Descrição:

- 1 - Apresenta a coleção criada com o nome default que é **New Collection**.
- 2 - É o local onde podemos alterar dar o nome da coleção, por convenção o nome da coleção é sempre igual ao nome do projecto para que saibamos sempre que determinada coleção é de determinado projecto.

4.3 Criando Requisições

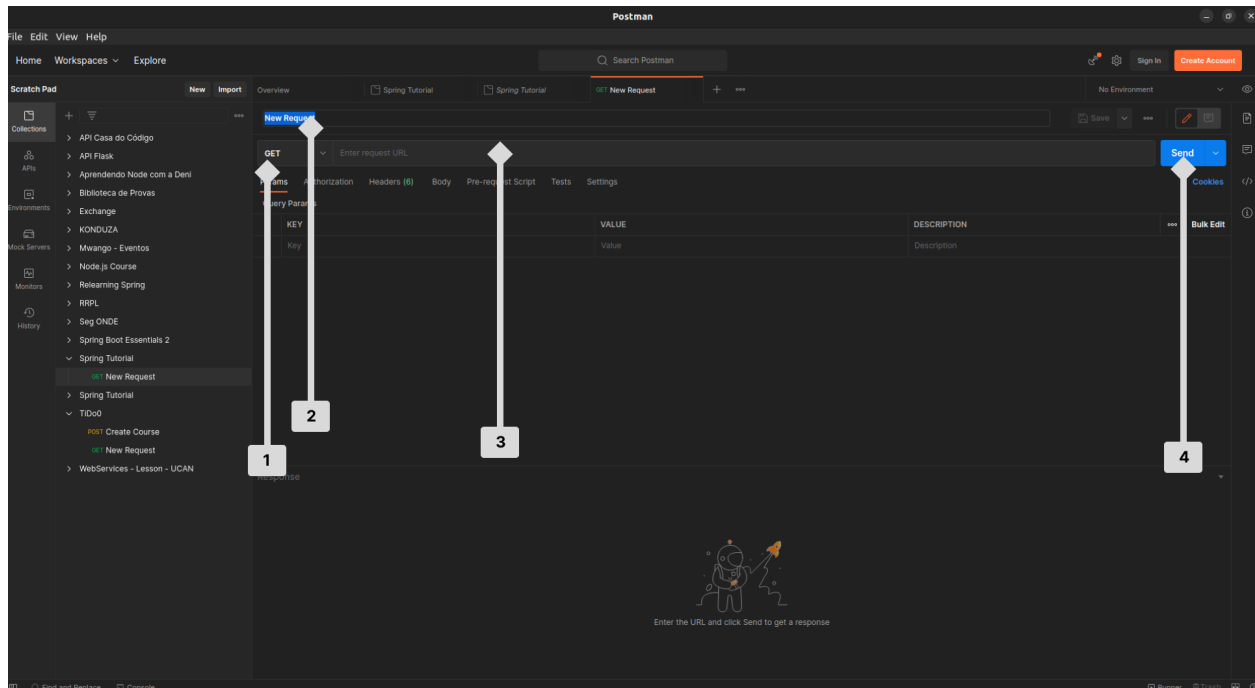
Essa é de certeza uma das partes mais importantes do Postman porque se refere a criação de requisições HTTP estas que iram nos permitir testar cada um dos nossos WebServices.



Descrição:

- **1** - Clicando com o lado direito na coleção criada, iremos ter essa pequena caixa de dialogo que devemos clicar em **Add Request**.

Após criada a requisição temos a seguinte tela, que pode parecer complicada mais é super simples e iremos explicar cada um dos campos apresentados.



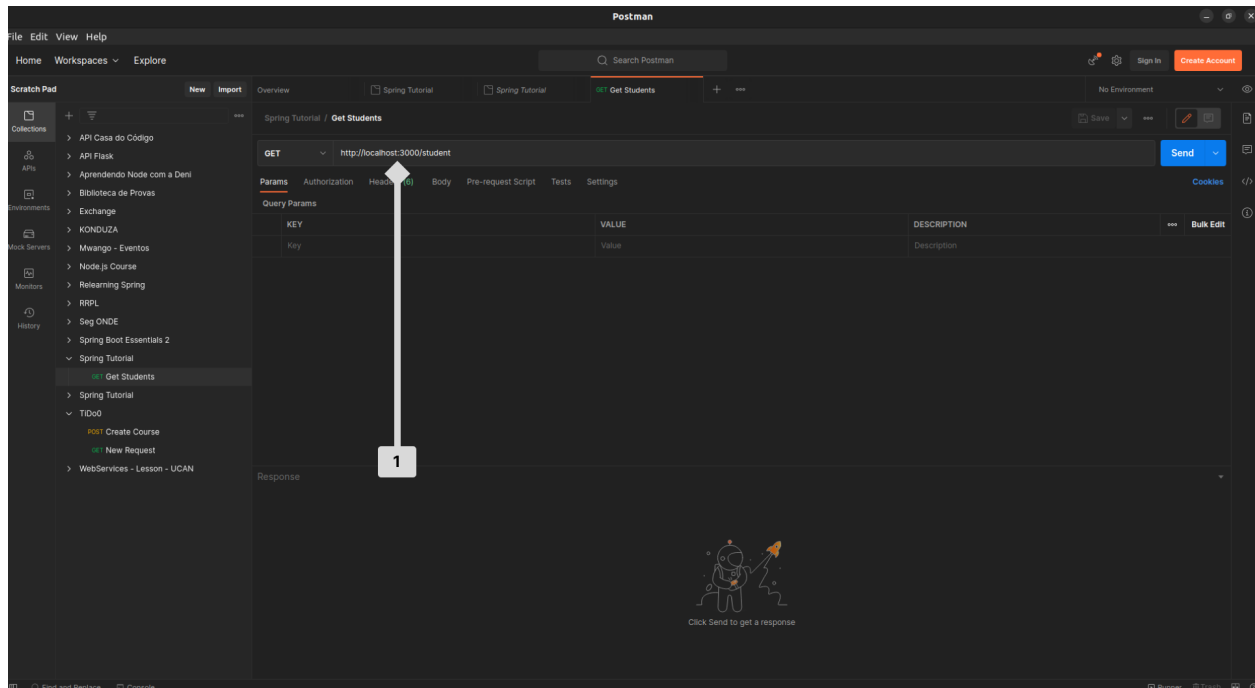
Descrição:

- **1** - É o local onde indicamos de qual verbo HTTP será aquela determinada requisição.
- **2** - Local onde determinamos o nome que daremos a requisição
- **3** - Especificamos o endpoint do WebService a ser testado, exemplo:
- **4** - Botão que envia a requisição criada para o nosso servidor que roda os nossos WebServices.

4.3.1 Criando requisições GET normais

Uma requisição do tipo GET normal, sem parâmetros, sem pathVariables e sem segurança é das mais simples que podem existir e é muito fácil criar.

A primeira coisa que devemos fazer é criar uma requisição, como ensinado na página anterior e partimos para a sua personalização.



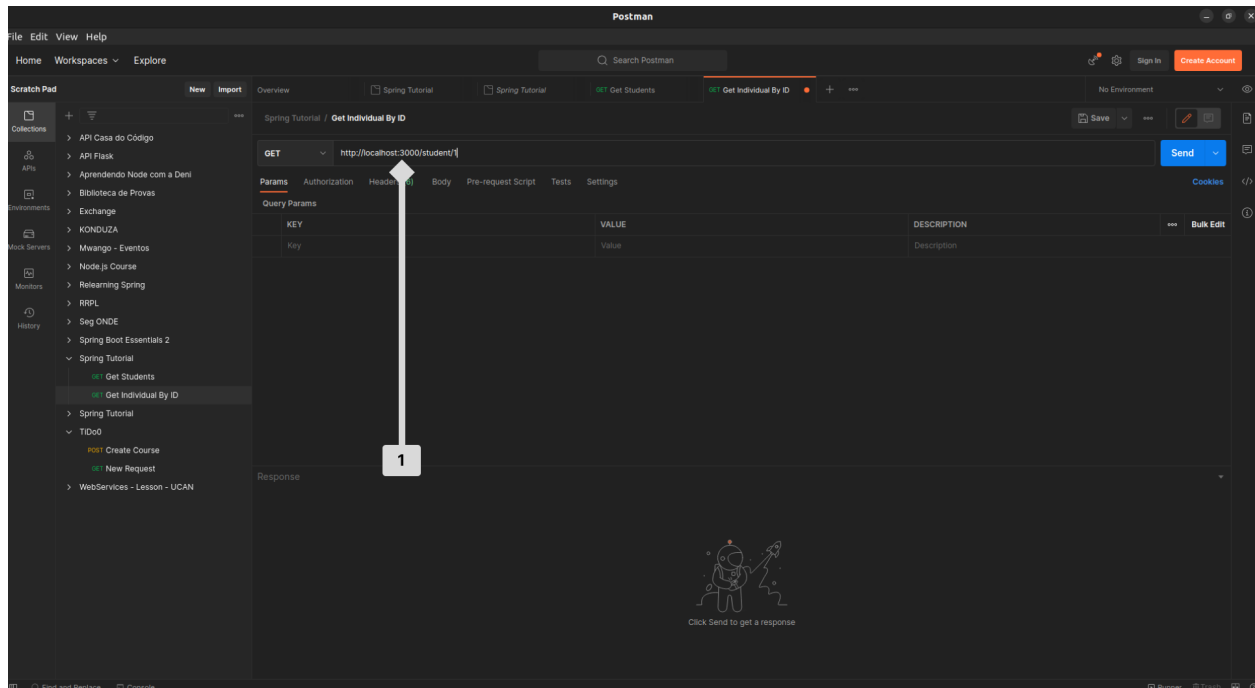
Descrição:

- **1** - Descrevemos o endpoint do WebService que será testado com o verbo HTTP do tipo GET: <http://localhost:8080/student>

4.3.2 Criando requisição GET com PathVariable

Criamos esse tipo de requisições quando pretendemos passar como informação pela url do endpoint, por exemplo o id de um estudante que a url para o WebService ficaria como: <http://localhost:8080/student/21099>.

O elemento da url negrito indica a PathVariable que é o id do estudante, sendo que será feita uma busca pelo estudante pelo id: 21099.



Descrição:

- 1 - Indicamos o endpoint a ser testado que é: <http://localhost:8080/student/1>

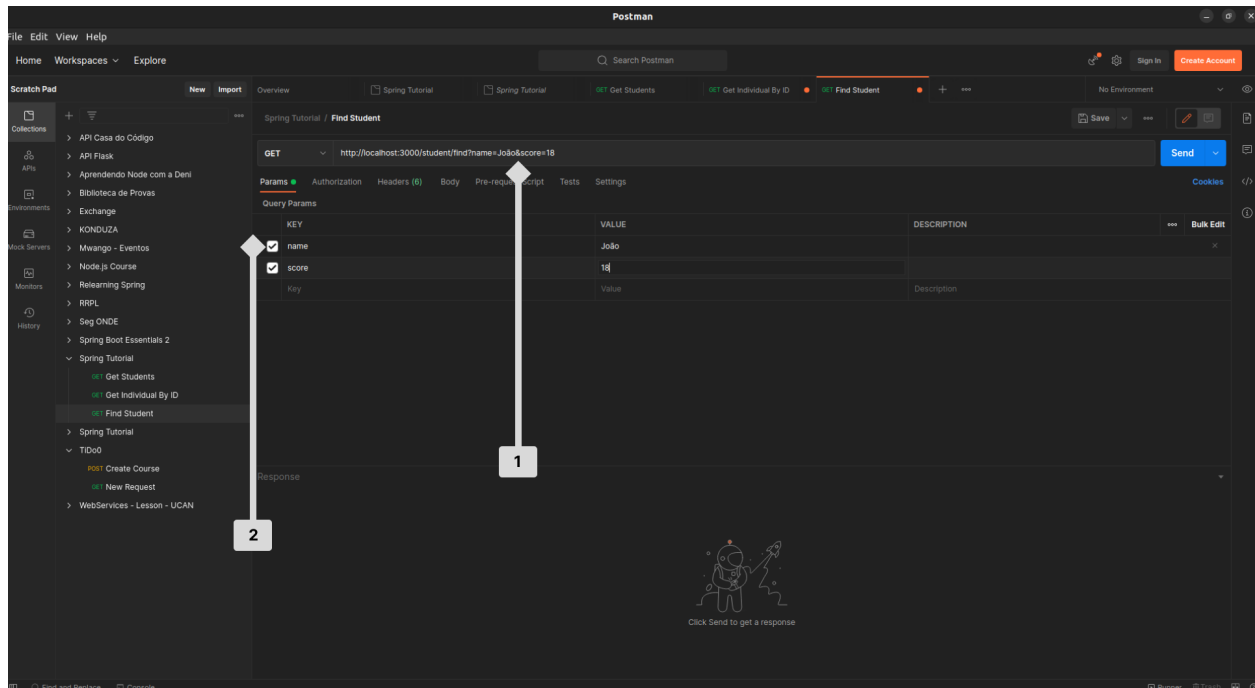
4.3.3 Criando requisições GET com QueryParams

Esse tipo de requisição é usada quando queremos passar parâmetros pela url, lembrando que esses parâmetros normalmente são enviados para fazer uma filtragem da resposta que teremos no nosso WebService e algumas informações adicionais.

Por exemplo se queremos procurar por estudantes que o seu nome inicial com **João** e que a sua nota foi **18** podemos passar isso como parâmetro para o nosso webservice da seguinte forma: <http://localhost:8080/student?name=João&score=18> .

Desta forma criamos a nossa requisição com os seus devidos parâmetros.

Parâmetros HTTP são sempre enviados depois da terminação com o sinal de ? e depois é feita um modelo de parâmetro e valor (parametro=valor) e essa combinação de parâmetros é sempre separada com & .

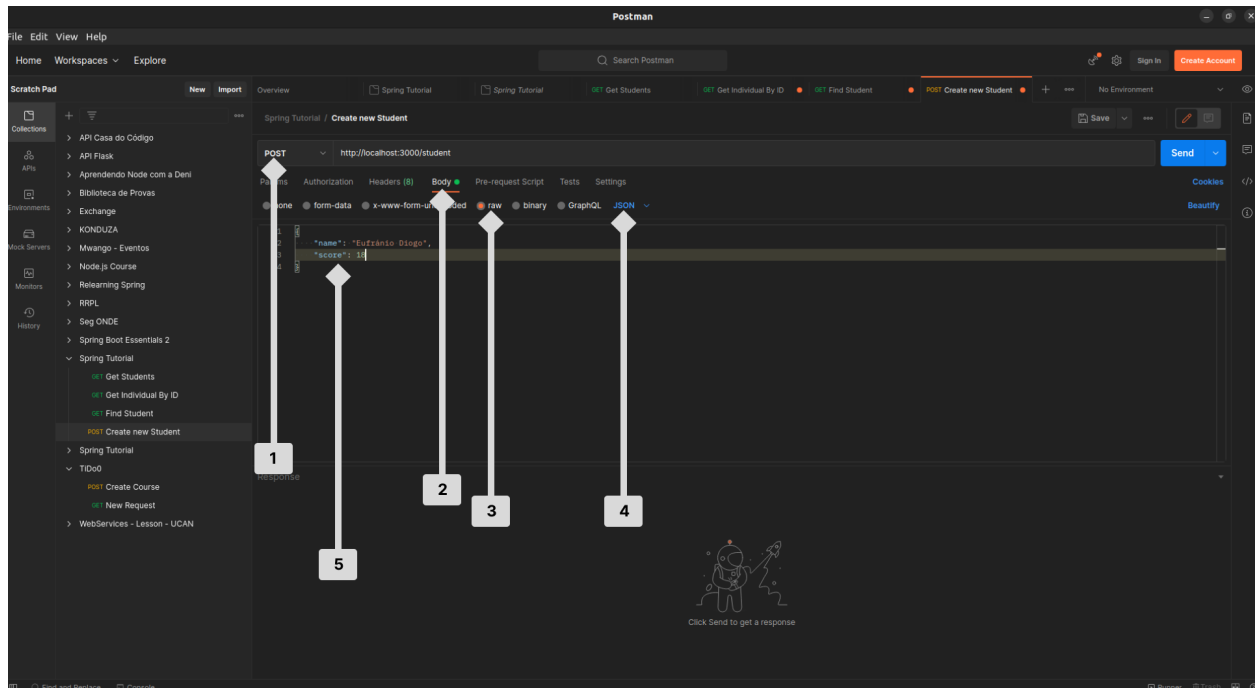


Descrição:

- **1** - Endpoint do webservice: <http://localhost:8080/student?name=João&score=18>
- **2** - Formulário onde colocamos o nome do parâmetro, valor e ainda descrição deste parâmetro. Para ter acesso a esse formulário precisamos de estar na tab **Params** demarcada como activa e com uma bolinha verde.

4.3.4 Criando requisição POST com JSON Body

Uma das tarefas que realizaremos inúmeras vezes é criação de requisições POST com corpo usando JSON (JavaScript Object Notation) que é um formato simples para especificarmos o corpo de uma entidade ou modelo na nossa aplicação.

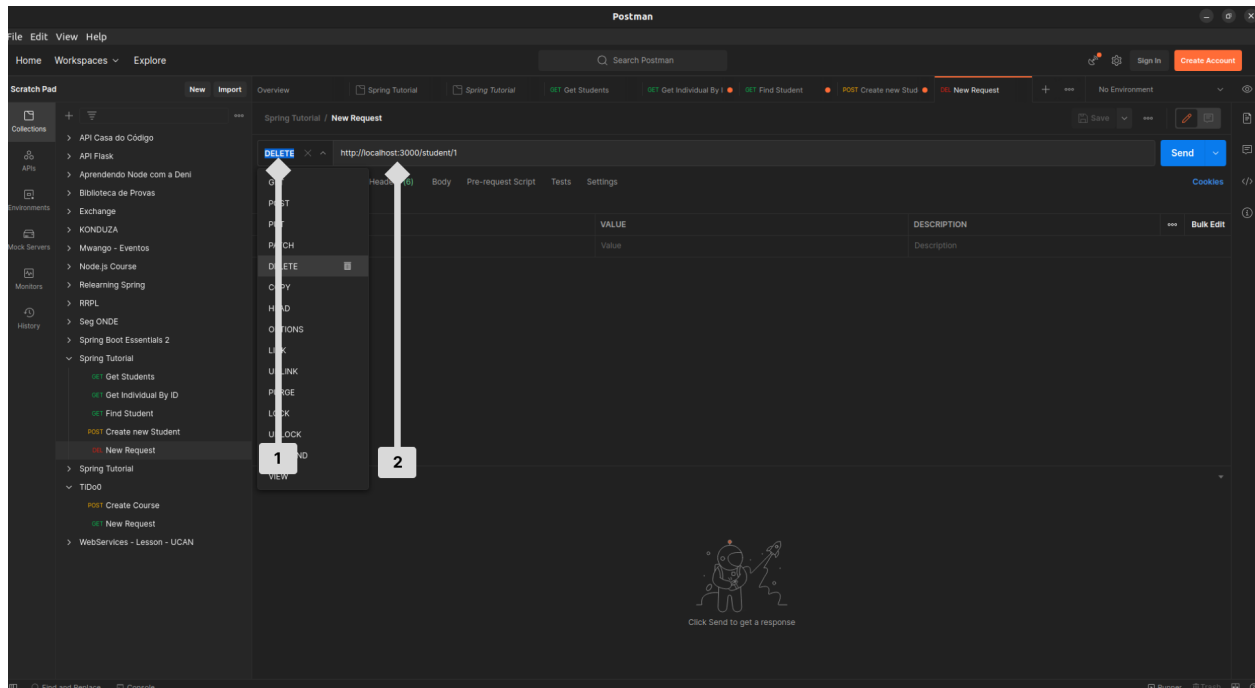


Descrição:

- **1** - Alteramos o verbo HTTP para POST.
- **2** - Entramos para a tab Body para podermos enviar um corpo na requisição.
- **3** - Escolhemos a opção raw, porque iremos enviar um corpo simples para o nosso webservice, sem nenhum ficheiro, arquivo binário e outros.
- **4** - Selecionamos o tipo de de corpo simples queremos no nosso caso **JSON**
- **5** - Declaramos o corpo json sendo que passamos um objecto com dois atributos que são **name** e **score** sendo ele o body. Att: Esse corpo por termos definido que o nosso corpo simples seria do tipo JSON devemos enviar ele como é formato JSON, caso contrário podemos sofrer erros desnecessários.

4.3.5 Criando uma requisição DELETE usando PathVariable

Após termos aprendido a criar requisições do tipo GET usando PathVariable se torna muito simples criar requisições DELETE com PathVariable porque a única coisa que muda é o verbo HTTP.

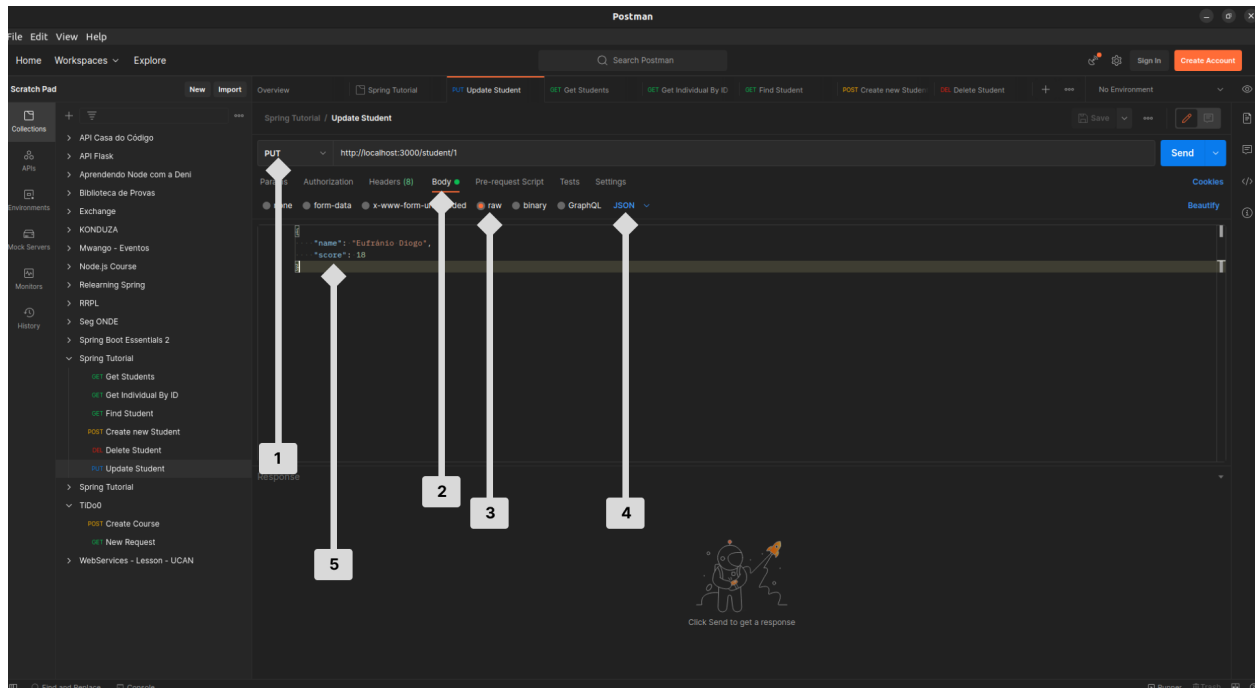


Descrição:

- 1 - Selecionamos o verbo HTTP que nesse caso será DELETE
- 2 - Descrevemos a url do webservice, indicando por exemplo o id de um estudante que a url para o WebService ficaria como: <http://localhost:8080/student/21099>.

4.3.6 Criando requisição PUT com JSON Body

Essa requisição é bem parecida com a requisição do tipo POST com Body, sendo que a sua única diferença é no corpo da requisição e o verbo HTTP utilizado que ao invés de ser POST mudamos para PUT.



Descrição:

- **1** - Alteramos o verbo HTTP para PUT.
- **2** - Entramos para a tab Body para podermos enviar um corpo na requisição.
- **3** - Escolhemos a opção raw, porque iremos enviar um corpo simples para o nosso webservice, sem nenhum ficheiro, arquivo binário e outros.
- **4** - Selecionamos o tipo de de corpo simples queremos no nosso caso **JSON**
- **5** - Declaramos o corpo json com as propriedades que pretendemos atualizar e enviamos. Att: Esse corpo por termos definido que o nosso corpo simples seria do tipo JSON devemos enviar ele como é formato JSON, caso contrário podemos sofrer erros desnecessários.