

# Relatório da Atividade Prática

## Letícia Roberta Oliveira Souto nº14

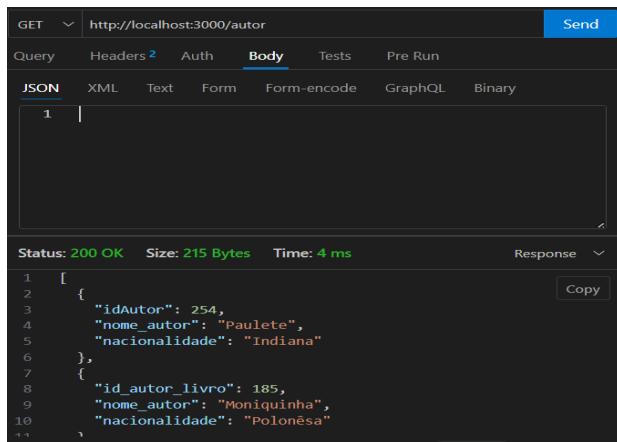
---

### Entidade Autor:

#### 1.Get

Para o Get exibir as informações da tabela autor foi necessário indicar na codificação do app.get o caminho que o navegador deveria fazer para mostrar tudo o que havia na tabela. O caminho é inserido após o a porta do servidor, através do nome da entidade, se o código estiver funcionando ao enviar esse caminho na navegação irá aparecer o id do autor, nome e nacionalidade, como é o caso do print abaixo.

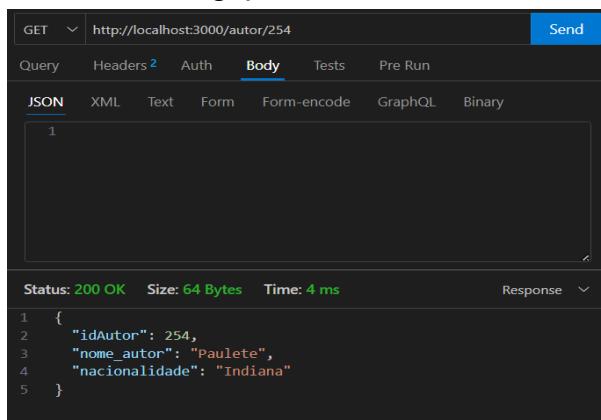
No código o status 200 indica que a comunicação está sendo realizada efetivamente e o json mostra as informações.



```
GET http://localhost:3000/autor
Status: 200 OK Size: 215 Bytes Time: 4 ms
[{"idAutor": 254, "nome_autor": "Paulete", "nacionalidade": "Indiana"}, {"idAutor": 185, "nome_autor": "Moniquinha", "nacionalidade": "Polonesa"}]
```

#### 2.Get com id:

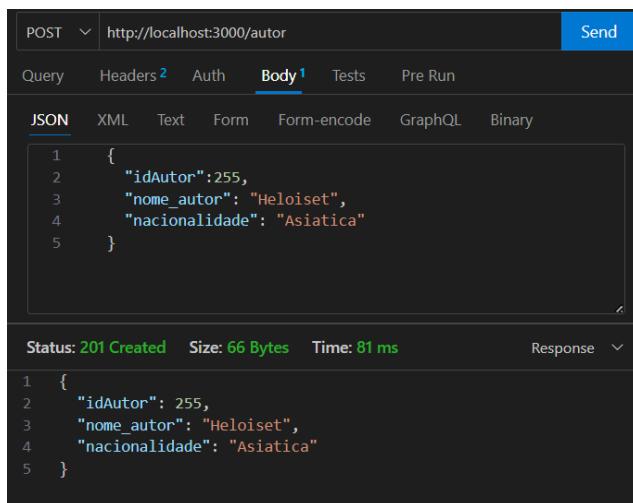
Para o get exibir informações de um autor específico foi necessário chamar a função de buscar autor, que é identificado através do id imposto a ele, esse id dentro do comando app.get é o parâmetro utilizado nessa busca. E da mesma forma que o get anterior o status vai indicar a comunicação e o json exibirá as informações do autor cujo id foi inserido no caminho da navegação.



```
GET http://localhost:3000/autor/254
Status: 200 OK Size: 64 Bytes Time: 4 ms
{
  "idAutor": 254,
  "nome_autor": "Paulete",
  "nacionalidade": "Indiana"
}
```

### 3.Post

Para o post adicionar um informação à entidade autor, foi necessário indicar que seria executado um push (adição) dentro da entidade autor. Ademais, foi necessário ir em json dentro do body e adicionar as informações desejadas, para testar se estava funcionando, no entanto essas informações tem que completar os requisitos já existentes dentro da entidade, no caso de autor é o id, nome e nacionalidade



POST  Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

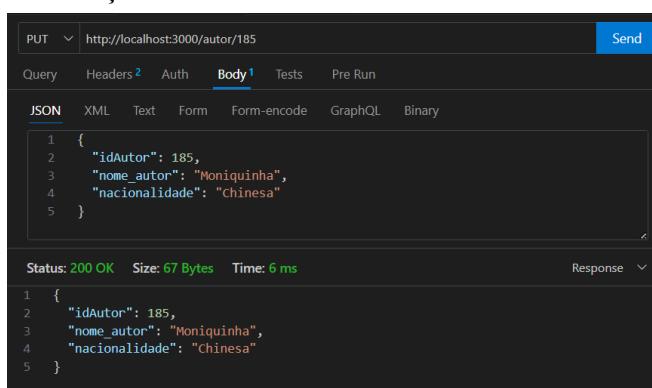
```
1 {
  "idAutor":255,
  "nome_autor": "Heloiset",
  "nacionalidade": "Asiatica"
}
```

Status: 201 Created Size: 66 Bytes Time: 81 ms Response

```
1 {
  "idAutor": 255,
  "nome_autor": "Heloiset",
  "nacionalidade": "Asiatica"
}
```

### 4.Put

Para o put ser executado foi necessário chamar a função de buscar autor, com o parâmetro sendo o id autor, assim ao incluir o idAutor no caminho da navegação e executar, o mesmo seria modificado, logo, as informações deveriam aparecer com as devidas modificações na exibição da tabela. Para alterar essas informações assim como no post é necessário ir até o json do body e escrever o atributo e a alteração desejada, por exemplo, no print abaixo, a informação alterada foi a nacionalidade do autor.



PUT  Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

```
1 {
  "idAutor": 185,
  "nome_autor": "Moniquinha",
  "nacionalidade": "Chinesa"
}
```

Status: 200 OK Size: 67 Bytes Time: 6 ms Response

```
1 {
  "idAutor": 185,
  "nome_autor": "Moniquinha",
  "nacionalidade": "Chinesa"
}
```

### 5.Delete

Para o delete ser executado também foi necessário chamar a função de buscar autor, com o parâmetro sendo o id do autor, assim, ao incluir o idAutor no caminho da navegação e executar, o mesmo seria excluído, e não deveria aparecer na exibição da tabela, como é o caso do print abaixo.

The screenshot shows a Postman interface with a 'DELETE' request to 'http://localhost:3000/autor/254'. The 'Query' tab is selected. The response status is '200 OK', size is '71 Bytes', and time is '3 ms'. The response body is a JSON array containing one object:

```
1 [  
2 {  
3   "idAutor": 185,  
4   "nome_autor": "Moniquinha",  
5   "nacionalidade": "Polonesa"  
6 }  
7 ]
```

## 6.Dificuldades Encontradas

Eu tive bastante dificuldade ao testar os códigos, além de pequenos desvios de atenção ao nome dos atributos dentro das entidades. Precisei criar todo o arquivo novamente, por conta do arquivo que eu estava usando estar corrompido, após reescrever o código consegui testar melhor os comandos, os quais deram certo. No entanto, demorei para conseguir executar o put, até descobrir que o erro era de digitação. Além disso, quando eu estava testando os comandos no arquivo corrompido, alguns deram certo pois eu não estava usando o servidor da pasta correta e sim de todos os arquivos, assim, o servidor não conseguia identificar os erros dos comandos.