

---

# Desenvolvimento WEB

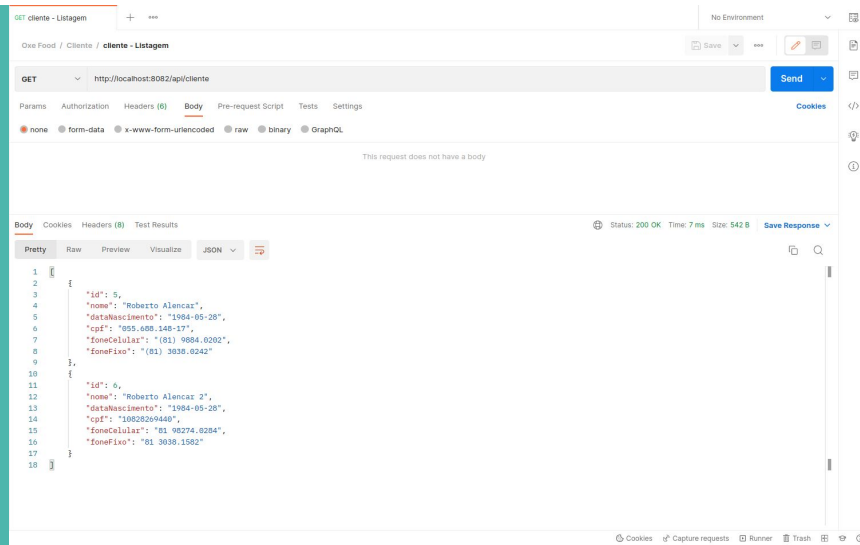
-

Full Stack Completo: Java + React

---

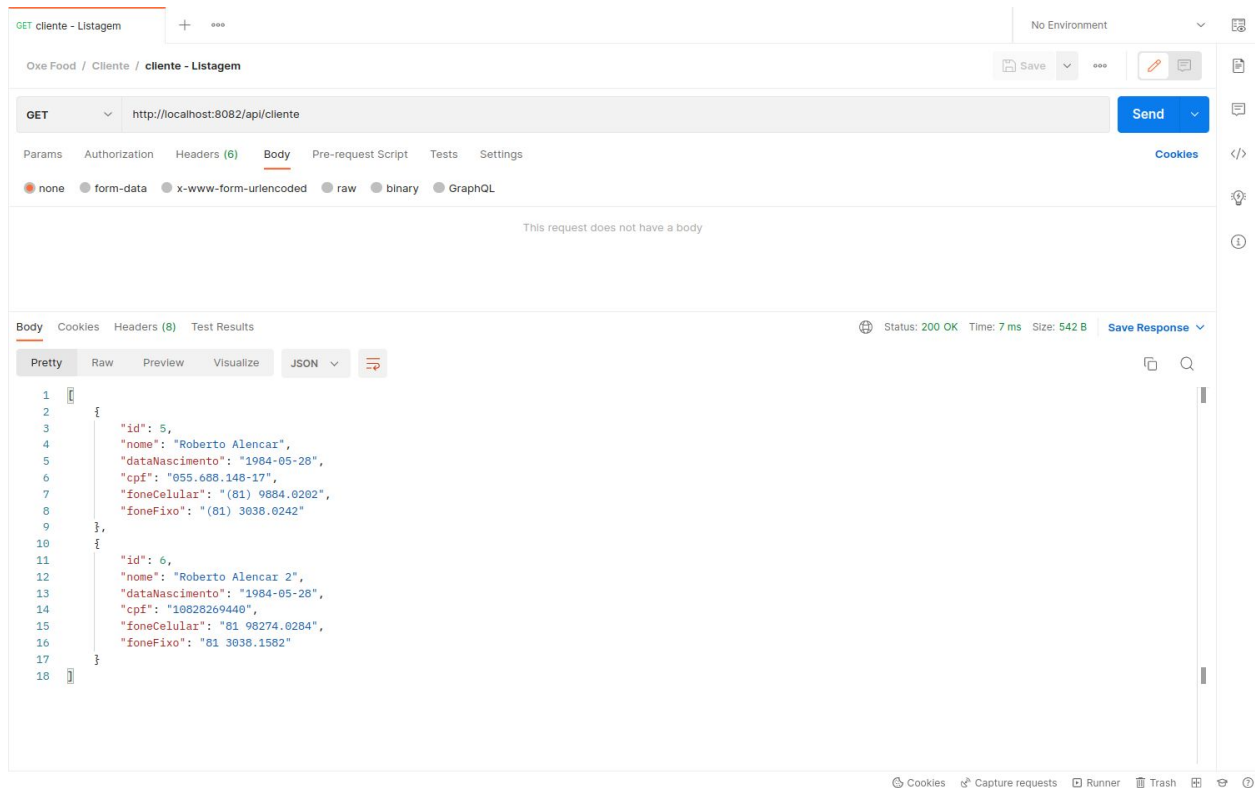
---

# Implementando a Listagem no Back-end



# Back-end :: Implementando a Listagem

**Objetivo: Listar (no Postman) todos os clientes cadastrados no banco de dados**



# Back-end :: Implementando a Listagem

1) Acrescente na classe `ClienteController` os dois métodos abaixo:

```
...
import java.util.List;
...
public class ClienteController {
    ...
    @GetMapping
    public List<Cliente> listarTodos() {
        return clienteService.listarTodos();
    }

    @GetMapping("/{id}")
    public Cliente obterPorID(@PathVariable Long id) {
        return clienteService.obterPorID(id);
    }

    ...
}
```

Esse método retorna todos os clientes cadastrados na tabela de clientes do banco de dados.

Esse método recebe um ID com parâmetro enviado na URL e retorna um objeto do tipo `Cliente`

# Back-end :: Implementando a Listagem

2) Acrescente na classe `ClienteService` os dois métodos abaixo em **negrito**:

```
...
public class ClienteService {

    ...

    public List<Cliente> listarTodos() {

        return repository.findAll();
    }

    public Cliente obterPorID(Long id) {

        return repository.findById(id).get();
    }

    ...

}
```

# Back-end :: Implementando a Listagem

## 3.1) Teste os dois métodos criados no Postman

The screenshot shows the Postman interface for a GET request to `http://localhost:8082/api/cliente`. The request is successful, returning a 200 OK status with a response time of 7 ms and a size of 542 B. The response body is displayed in the 'Body' tab, showing a JSON array of two client objects.

```
1 {
2   {
3     "id": 5,
4     "nome": "Roberto Alencar",
5     "dataNascimento": "1984-05-28",
6     "cpf": "055.688.148-17",
7     "foneCelular": "(81) 9884.0202",
8     "foneFixo": "(81) 3038.0242"
9   },
10  {
11    "id": 6,
12    "nome": "Roberto Alencar 2",
13    "dataNascimento": "1984-05-28",
14    "cpf": "10828269440",
15    "foneCelular": "81 98274.0284",
16    "foneFixo": "81 3038.1582"
17  }
18 }
```

The bottom of the interface shows a status bar with icons for Cookies, Capture requests, Runner, Trash, and other utility functions.

# Back-end :: Implementando a Listagem

## 3.2) Teste os dois métodos criados no Postman

The screenshot shows the Postman interface for a GET request. The request is named "cliente - getID" and is directed to the URL "http://localhost:8082/apl/cliente/5". The response status is "200 OK" with a time of "9 ms" and a size of "398 B". The response body is displayed in JSON format, showing a client record with the following details:

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "id": 5,
3   "nome": "Roberto Alencar",
4   "dataNascimento": "1984-05-28",
5   "cpf": "055.688.148-17",
6   "foneCelular": "(81) 9884.0202",
7   "foneFixo": "(81) 3038.0242"
8 }
```

The interface also shows tabs for Params, Authorization, Headers (7), Body, Pre-request Script, Tests, and Settings. The Body tab is currently selected, showing the response in Pretty, Raw, Preview, Visualize, and JSON formats. The JSON format is selected, displaying the response body.

# Dúvidas





# Exercício



Implemente a listagem e a consulta por ID (para o teste no Postman) de:

- Cliente
- Produto
- Entregador

The background features two large, solid green abstract shapes. One is a semi-circle on the left side, and the other is a more complex, organic shape on the right side.

Obrigado !