|  |
| --- |
| Implantação do Projeto CodeIgniter 4 no XAMPP (Windows) |

Aqui está uma documentação detalhada para implantar um projeto ZIP servidor XAMPP (PHP 8.2.12) no Windows, além de como importar o banco de dados.

Pré-requisitos

1. XAMPP instalado e configurado (com Apache e MySQL).
2. PHP 8.2.12 (ou versão compatível com o CodeIgniter 4).
3. Banco de dados MySQL ou MariaDB configurado.

Passo 1: Instalando o XAMPP

1. Baixe o instalador do XAMPP para Windows

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/8.2.12/>

1. Execute o instalador e siga as instruções na tela.
2. Selecione os componentes Apache, MySQL e PHP e conclua a instalação.
3. Abra o XAMPP Control Panel e inicie os serviços Apache e MySQL.

Passo 2: Implantando o Projeto CodeIgniter 4

Descompactando o Projeto ZIP

1. Localize o arquivo ZIP do projeto CodeIgniter 4.
2. Extraia o conteúdo do arquivo ZIP para a pasta de projetos do XAMPP (C:\xampp\htdocs\).

O caminho ficará assim: (C:\xampp\htdocs\api-compras).

OBS: Para que o projeto funcione, ele deve rodar na porta 8080. Caso utilize outra porta, por favor alterar a porta na variável $baseURL em app/Config/App.php.

O XAMPP geralmente configura o Apache para escutar a porta 8080 por padrão, mas se necessário, altere o httpd.conf do Apache em C:\xampp\apache\conf\httpd.conf para a porta desejada.

Procure por Listen 8080 e altere para Listen 80.

Passo 3: Importando o Banco de Dados

1. No phpMyAdmin, clique no banco de dados recém-criado.
2. Clique na aba Importar.
3. Clique em Escolher Arquivo e selecione o arquivo .sql (o arquivo de dump do banco de dados do projeto).
4. Clique em Executar para importar o banco de dados.

Passo 4: Instalando o Postman no Windows

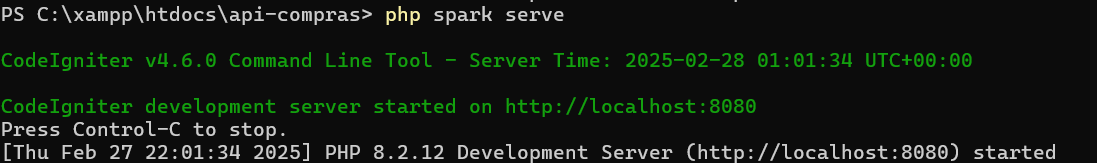
Acesse o site oficial do Postman:   
  
<https://www.postman.com/downloads/>

Execute o Instalador e siga as Instruções de Instalação.

Se tudo estiver correto, é hora do teste!

Passo 5: Teste

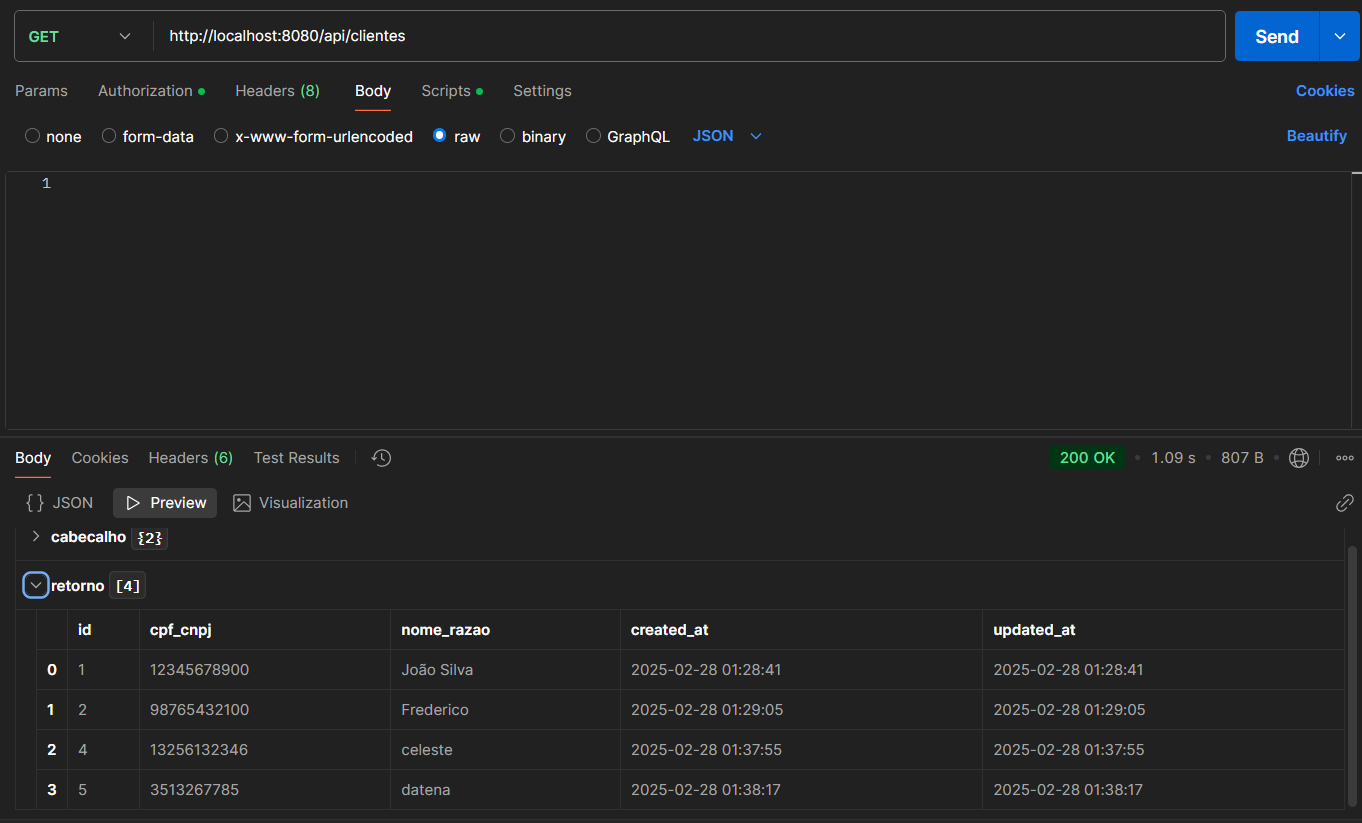
Abra o prompt de comando na localização do projeto em “C:\xampp\htdocs\api-compras” e digite o seguinte comando: php spark serve



Se o prompt retornar informações parecidas, está tudo pronto para o teste.

Testando Clientes

Utilizando a URL <http://localhost:8080/api/clientes>, você poderá ver a lista de clientes cadastrados.

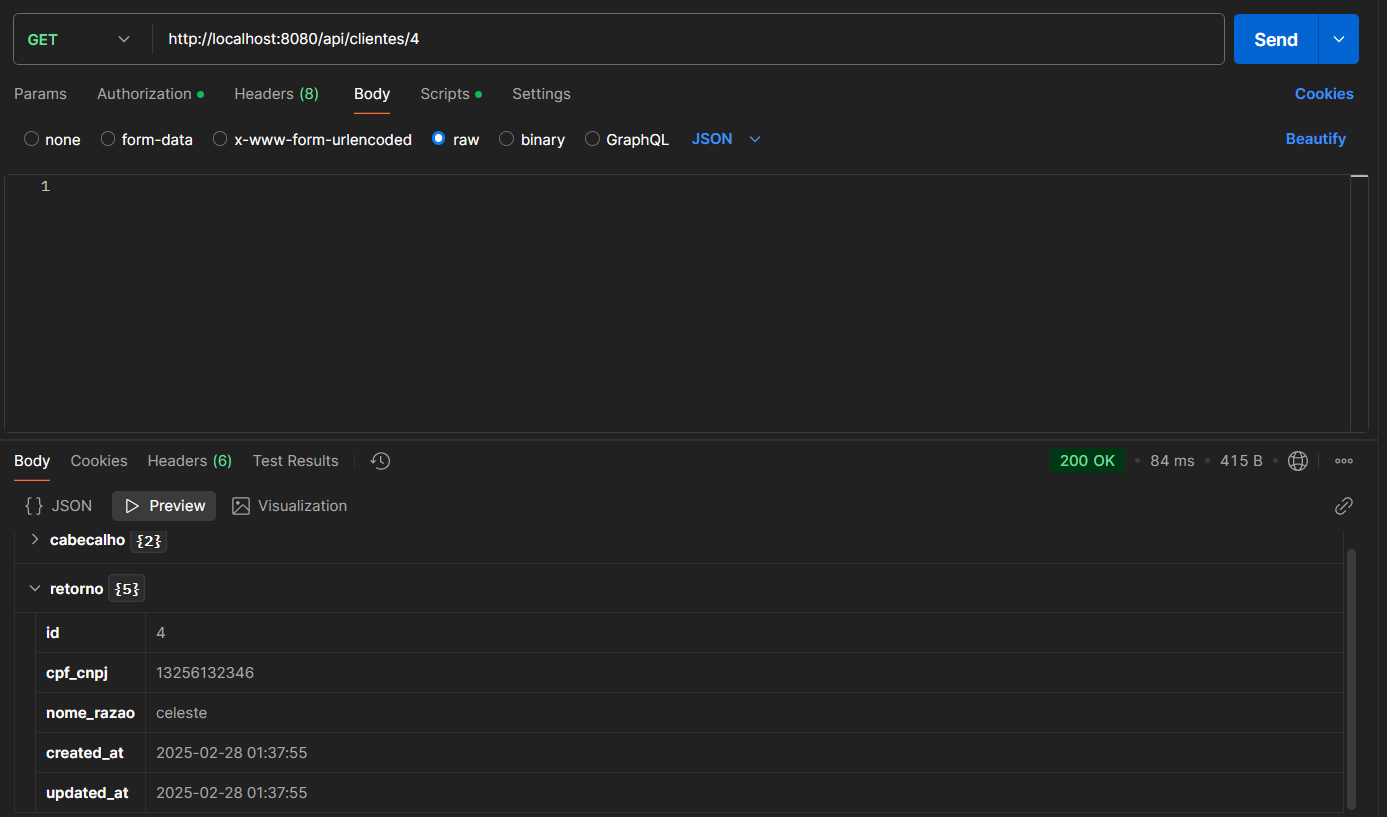


GET

No Postman, verifique se está no formato correto:

* Método: GET
* URL: <http://localhost:8080/api/clientes/4>
* Headers => Content-Type: application/json

Ao clicar em SEND, ele deve retornar o cliente com ID 4. Como no exemplo abaixo.



POST

No método POST, será usado o Body(JSON) para criar clientes no banco.

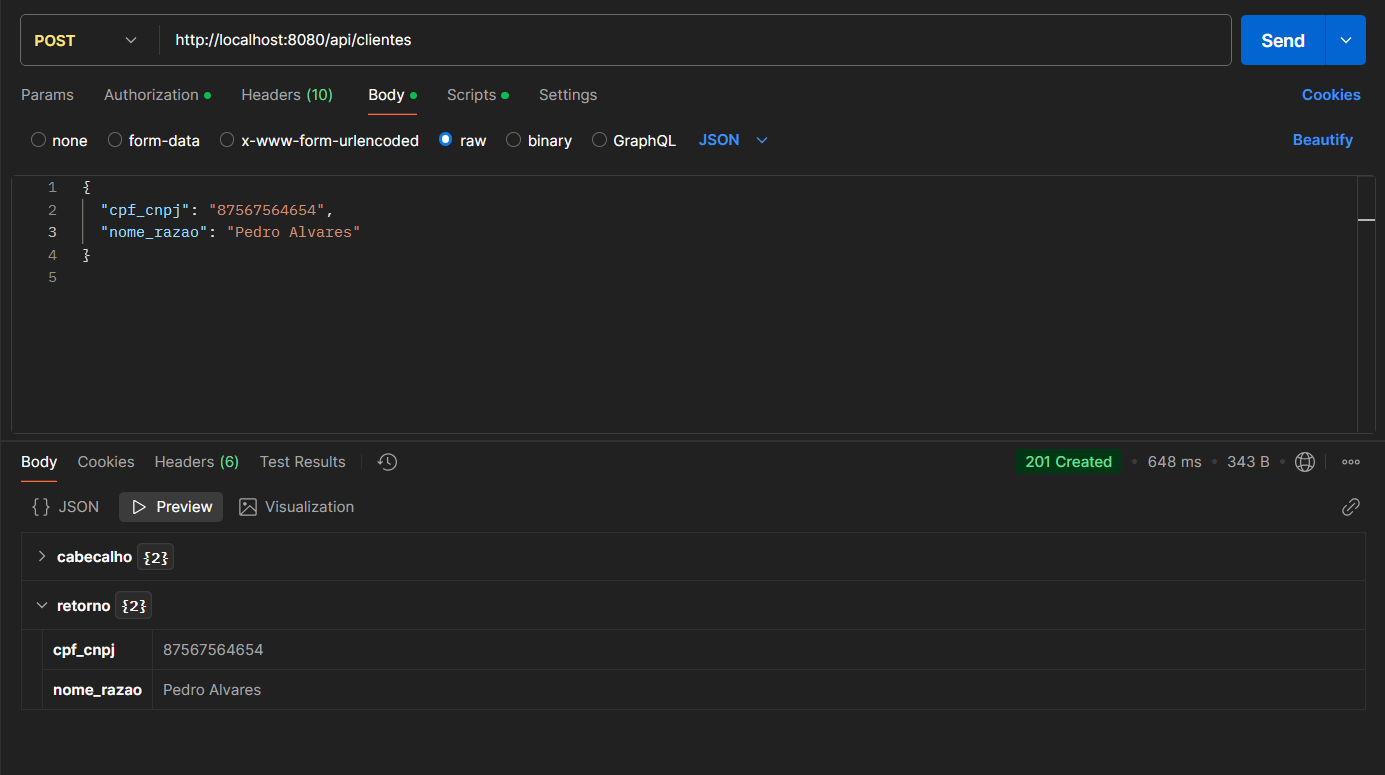
* Método: POST
* URL: <http://localhost:8080/api/clientes>
* Headers => Content-Type: application/json
* Body (JSON RAW)

{

"cpf\_cnpj": "12345678900",

"nome\_razao": "João Silva"

}



UPDATE

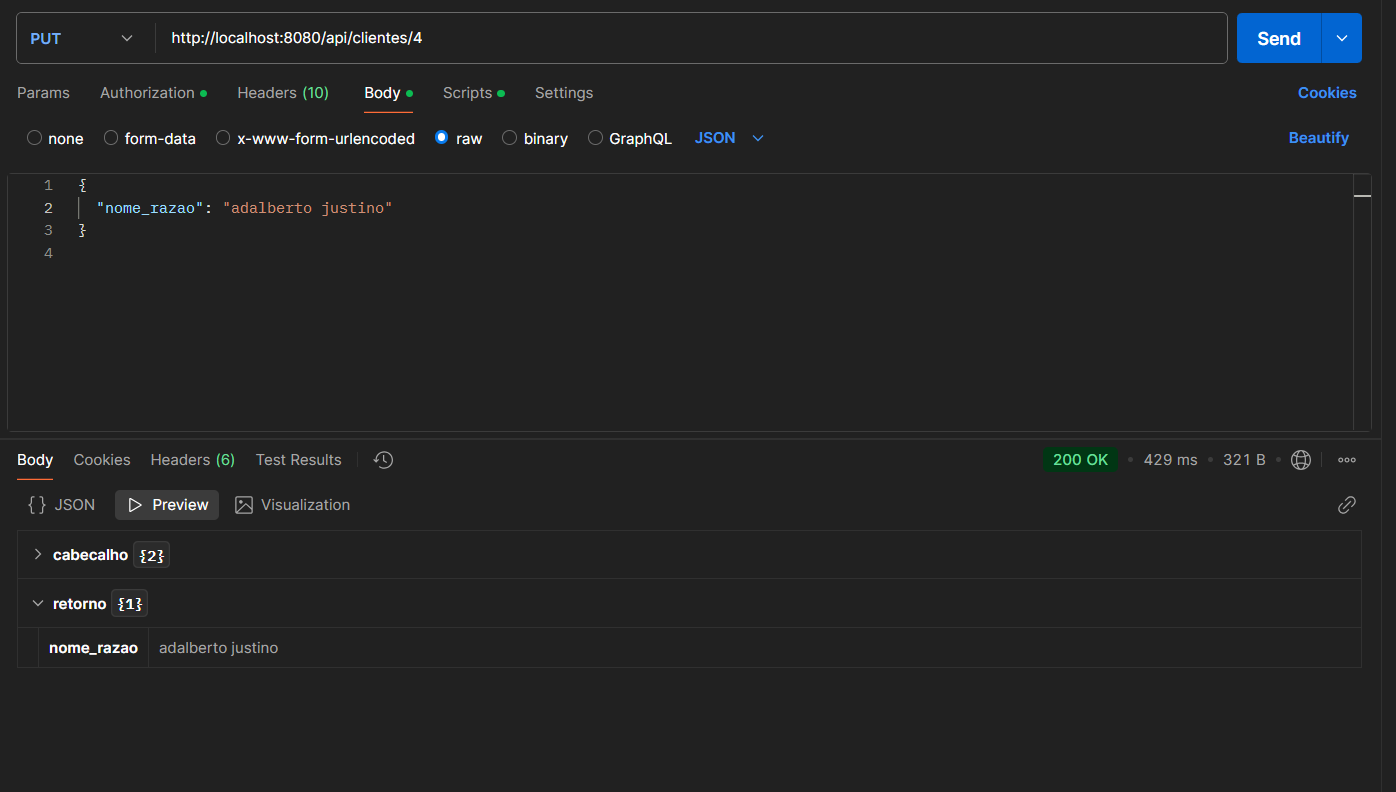
No método PUT, será usado para atualizar clientes já existentes no banco. O número na URL representa o ID do Cliente.

* Método: PUT
* URL: <http://localhost:8080/api/clientes/4>
* Body (JSON - RAW):

{

"nome\_razao": "Adalberto justino"

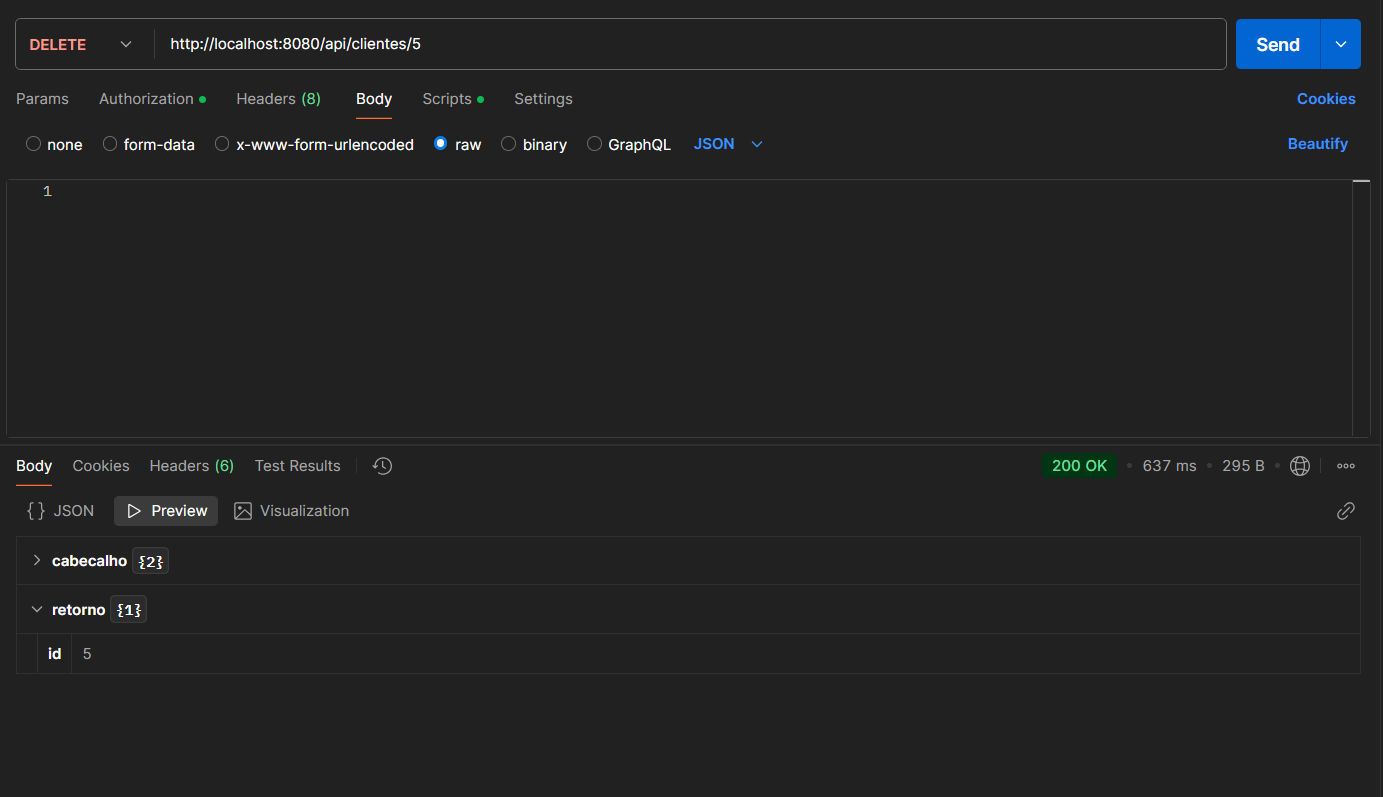
}



DELETE

Usando o delete, ao clicar em SEND, ele irá deletar o cliente com ID 1.

* Método: DELETE
* URL: <http://localhost:8080/api/clientes/5>



Esse é o desafio 1 totalmente funcional. O desafio 2 tentei implementar, mas está apresentando alguns erros e optei por deixar apenas o desafio 1. Porém caso haja a oportunidade, apresentarei funcionando.