

INSTRUÇÕES DE USO

COMO FUNCIONA?

Todas as rotas utilizam o método **POST**, e precisam o usuário passe informações no formato **JSON** no corpo da requisição. Existem diversos programas que realizam isso, estarei utilizando o **POSTMAN** para explicar

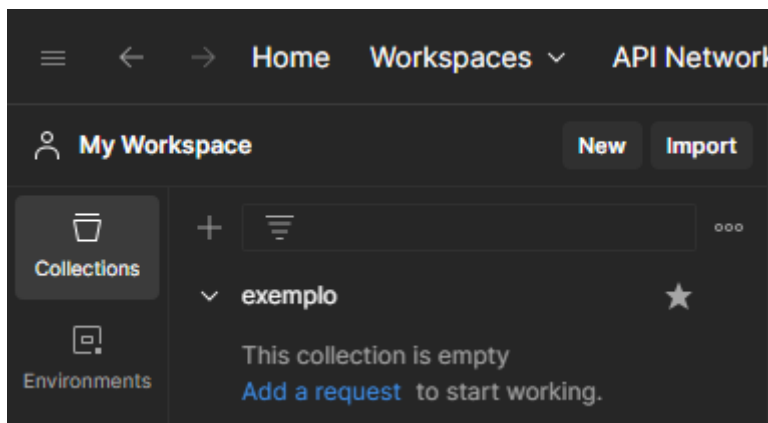
PRIMEIROS PASSOS

OBS: Caso não tenha ficado claro no readme, você precisa baixar o mysql e já criar a database onde o projeto irá rodar. As tabelas serão criadas automaticamente, então não precisa se preocupar com isso.

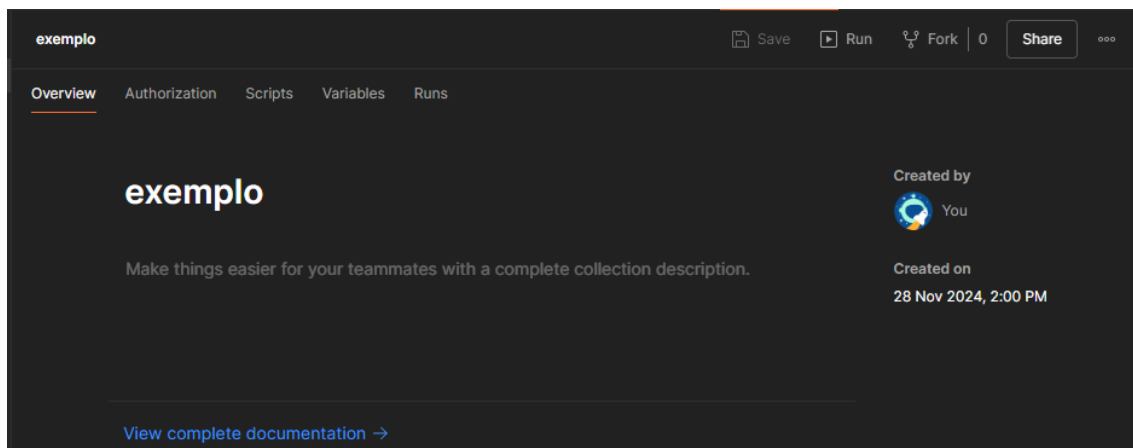
Após fazer a instalação do projeto e ter rodado o flask --app main.py run, seguindo as instruções do README.md (<https://github.com/Alexandre-CES/banco-de-dados/tree/main/desafio>), irá aparecer no terminal:

- Running on <http://127.0.0.1:5000>

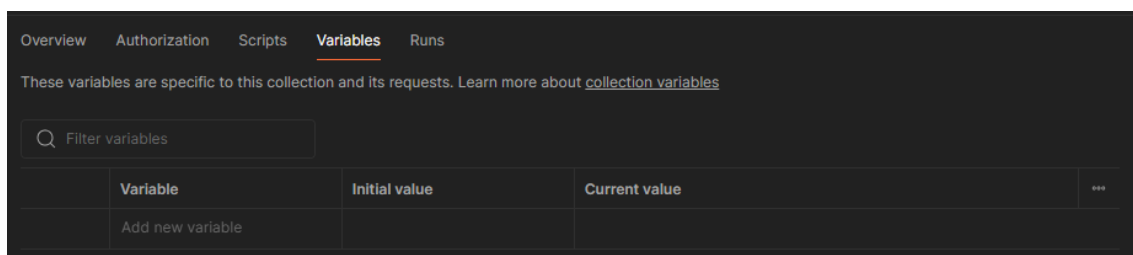
Copie esse link e abra o POSTMAN (ou o que preferir), crie uma **collection**:



Clique nessa nova collection criada, o que vai abrir algo como:



Clique em **Variables**, que abrirá esta outra região:



Clique em **Add new variable** e escreva **baseurl**, depois **cole** dentro do **initial value** o link que copiou no terminal do aplicativo. Vai ficar assim:

	Variable	Initial value	Current value	...
<input checked="" type="checkbox"/>	baseurl	http://127.0.0.1:5000	http://127.0.0.1:5000	
	Add new variable			

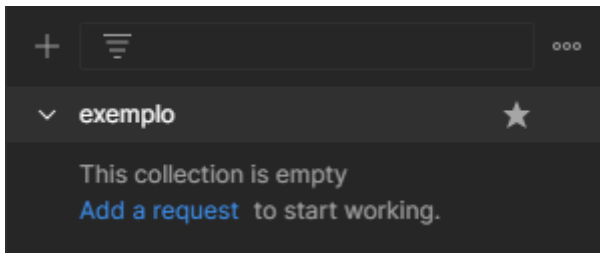
Clique **CTRL + S** para salvar

CRIANDO REQUISIÇÃO:

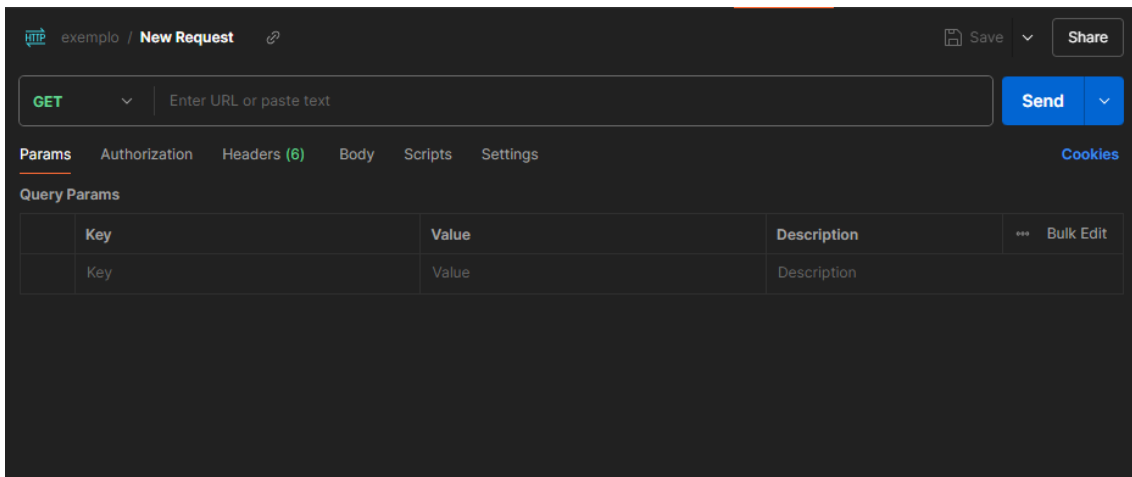
Os passos que ensinarei agora servem para todas as rotas do aplicativo, mais para frente explicarei melhor o que cada rota faz e o que será preciso enviar no corpo da requisição.

ADICIONE UMA REQUISIÇÃO

Dentro da collection que criou, clique em **Add a request** (*se já houver outra(s) rota(s) essa opção não aparecerá assim, então, você pode só clicar com o botão direito da collection e ir na opção Add request*):

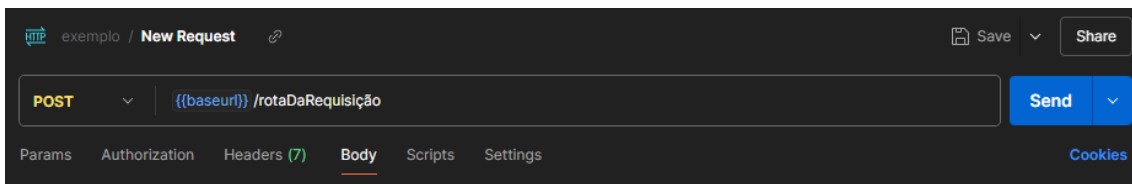


O que abrirá o seguinte:



COLOQUE O MÉTODO E URL BASE

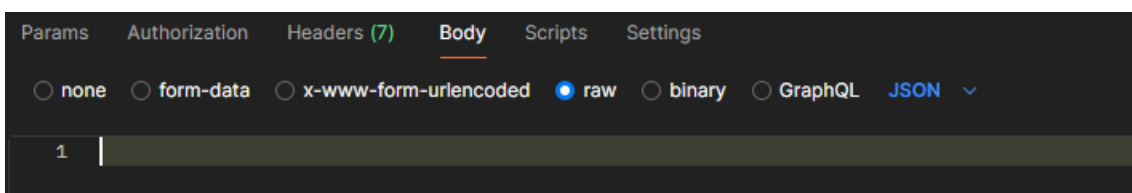
Clique no método de requisição, que provavelmente estará como GET, e coloque POST. Depois escreva {{baseurl}}/ na URL, ficando dessa forma:



- Onde está escrito **rotaDaRequisição**, é justamente o que você irá alterar de uma rota para outra, juntamente com o objeto JSON, que será explicado a seguir.

PREPARE O CORPO DA REQUISIÇÃO

Por último, clique em Body, selecione a opção raw e coloque JSON:



É aqui que você colocará o corpo da requisição, será sempre um objeto JSON.

EXPLICANDO ROTAS:

Cada rota precisa receber um JSON com dados e estruturas específicas para que o programa funcione corretamente. Para cada rota, irei separar um modelo de JSON para que você possa copiar e apenas trocar para as informações que deseja.

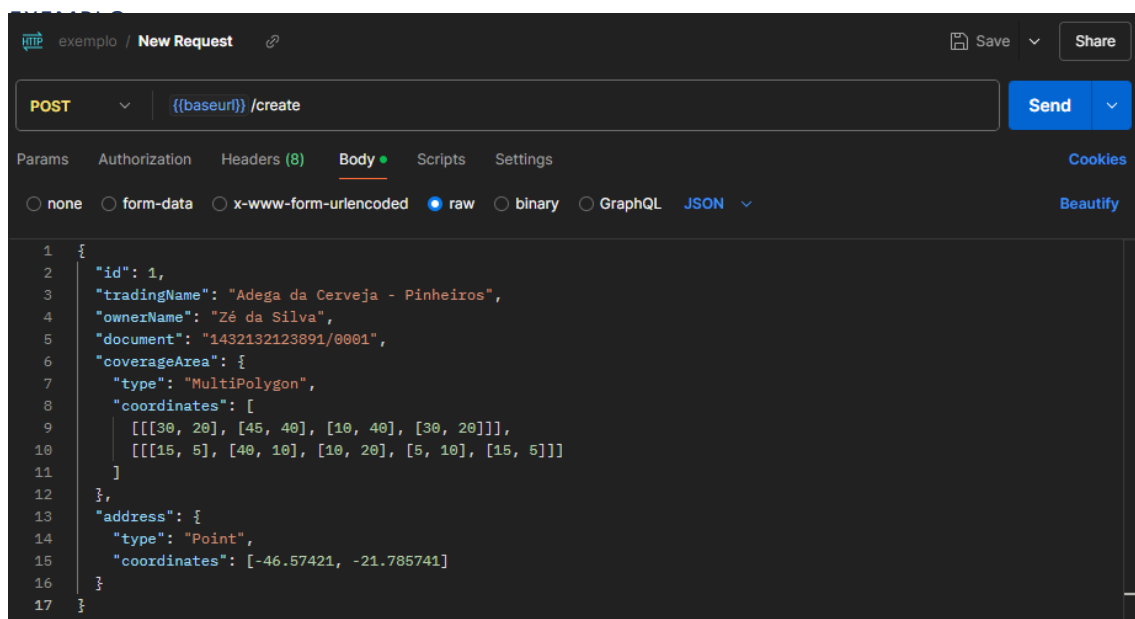
CREATE

A rota create cria um novo partner, o adicionando na base de dados.

MODELO DE JSON

```
{
  "id": 1,
  "tradingName": "Adega da Cerveja - Pinheiros",
  "ownerName": "Zé da Silva",
  "document": "1432132123891/0001",
  "coverageArea": {
    "type": "MultiPolygon",
    "coordinates": [
      [[[30, 20], [45, 40], [10, 40], [30, 20]]],
      [[[15, 5], [40, 10], [10, 20], [5, 10], [15, 5]]]
    ]
  },
  "address": {
    "type": "Point",
    "coordinates": [-46.57421, -21.785741]
  }
}
```

- Id e document são campos únicos, então não será possível adicionar dois partners de id 1, por exemplo.
- O campo "type" de coverageArea e address devem ser sempre MultiPolygon e Point, respectivamente



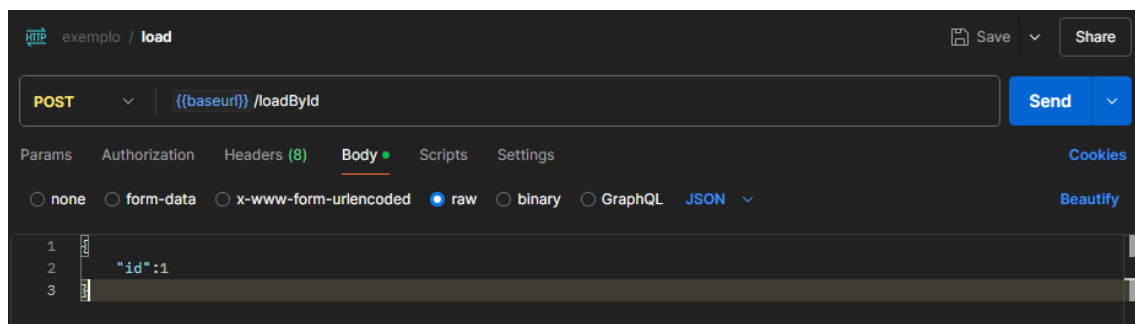
LOADBYID

Recebe um id no corpo da requisição, e retorna o partner correspondente

MODELO DE JSON

```
{
  "id":1
}
```

EXEMPLO



SEARCH

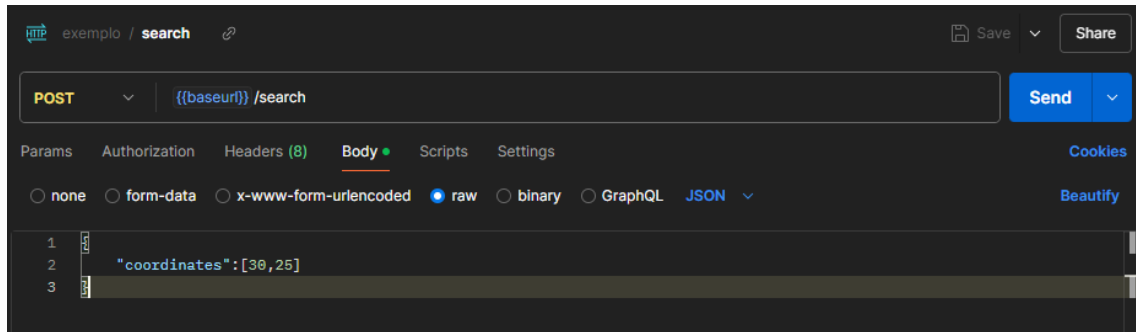
Retorna o Partner mais próximo das coordenadas enviadas, se estas estão dentro de sua área de atuação:

MODELO JSON

```
{
  "coordinates": [30, 25]
}
```

- Longitude e latitude

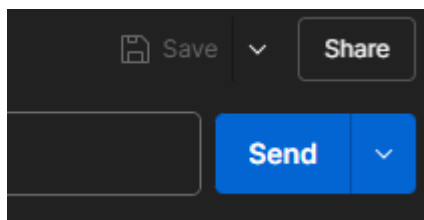
EXEMPLO



ENVIANDO REQUISIÇÃO

Agora é simples, basta clicar em **Send** e, se tudo estiver correto, o programa retornará um código de status 200 juntamente com um corpo de resposta. Se der errado, voltará com um código 400 no caso de você ter preenchido algo errado ou 500 se o problema está no projeto:

SEND



EXEMPLO DE RESPOSTA

BodyCookiesHeaders (5)Test Results

200 OK32 ms454 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "address": {
3     "coordinates": [
4       -21.785741,
5       -46.57421
6     ],
7     "type": "Point"
8   },
9   "coverageArea": {
10    "coordinates": [
11      [
12        [
13          [
```

PostbotRunnerCapture requestsCookiesVaultTrash