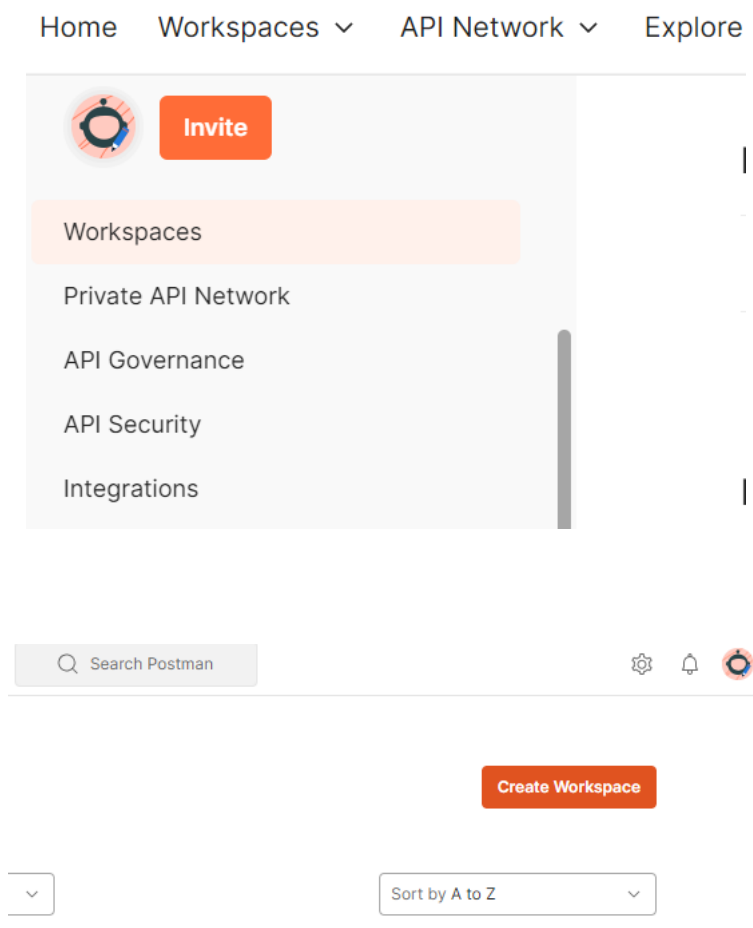


Testando a nossa API pelo Postman

Agora que finalizamos nossa API, chegou a hora de testarmos. Vamos utilizar o postman para realizar as requisições HTTP.

Passo 1: Abra o Postman, clique no menu home, opção workspace e create workspace.



Passo 2: Escreva um nome e uma descrição para sua workspace.

Create workspace

Name

Reprograme_se

Summary

Add a brief summary about this workspace.

Teste da minha primeira API

Visibility

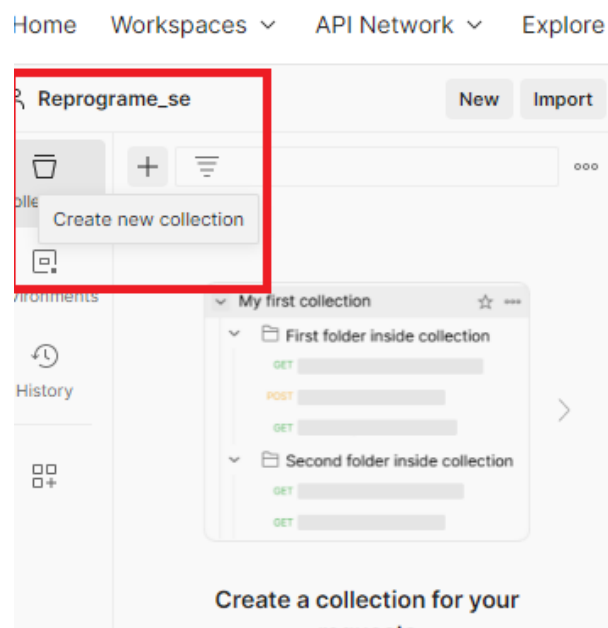
Determines who can access this workspace.

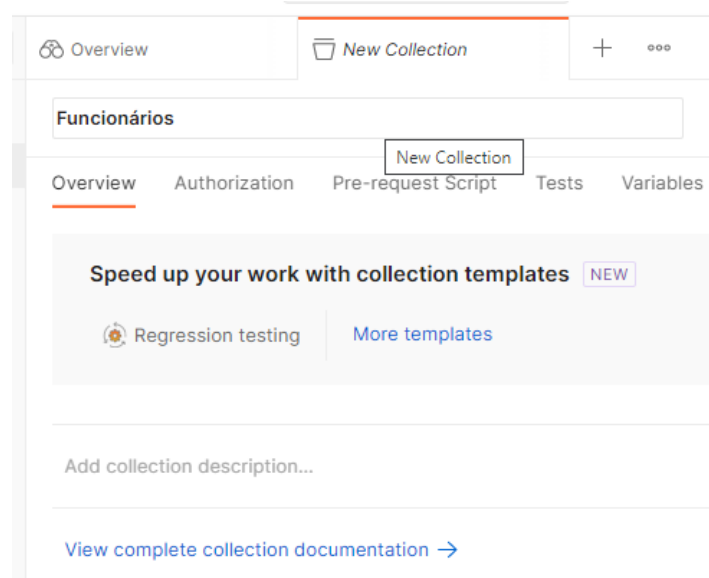
- ☐ Personal
Only you can access
- ☐ Private
Only invited team members can access
- ☒ Team
All team members can access
- ☐ Partner NEW
Only invited partners and team members can access
- ☐ Public
Everyone can view

Create Workspace

Cancel

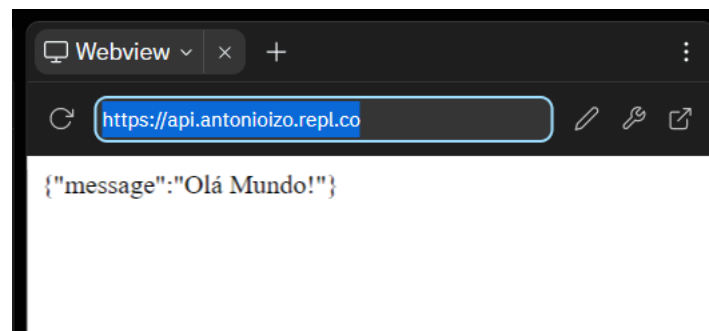
Passo 3: Vamos agora criar uma coleção para nossa workspace chamada Funcionários.





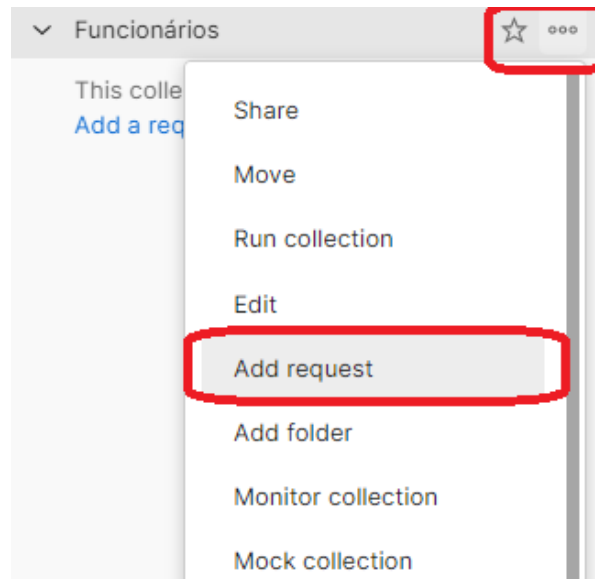
Requisição Get - Listar

Passo 1: Chegou a hora de adicionarmos requisições (GET, POST, PUT e DELETE), vamos antes no replit e executar nosso aplicativo para copiarmos o endereço da nossa URL da aplicação.

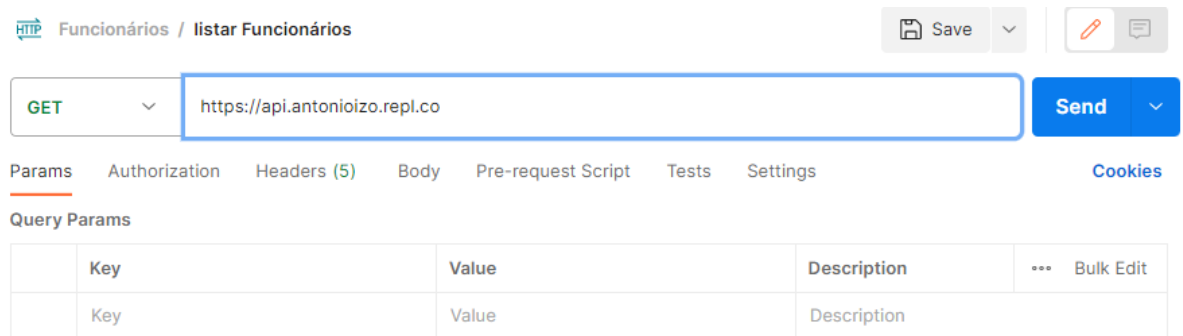


Atenção: Enquanto fazemos as requisições, nossa aplicação precisa estar sendo executada.

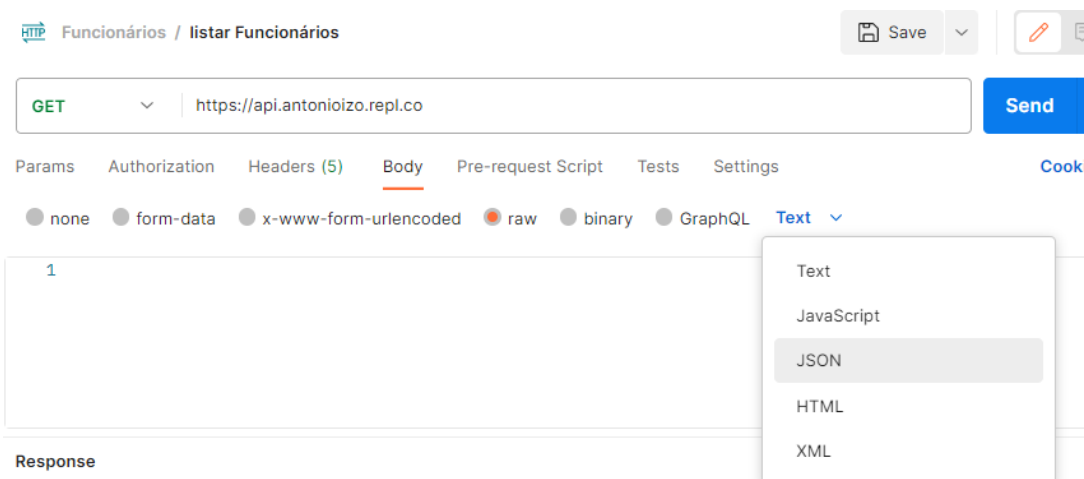
Passo 2: Retornando no Postman, clique na opção de add request da coleção Funcionários.



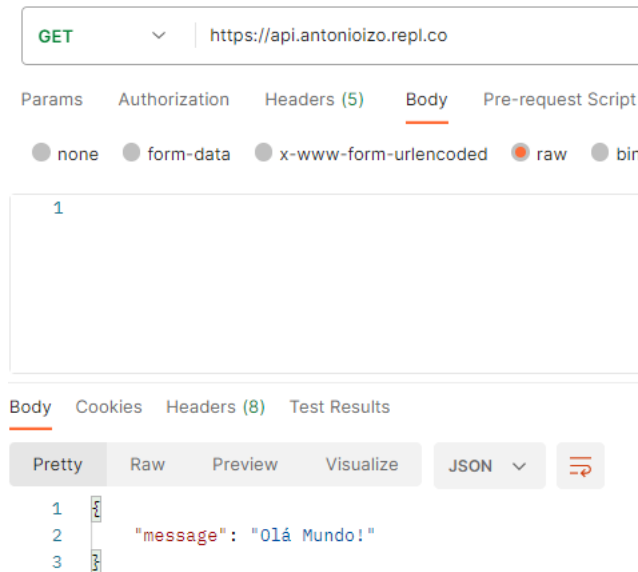
Passo 3: Coloque o nome da requisição como listar funcionários, ela será uma requisição GET e adicione o endereço da sua aplicação.



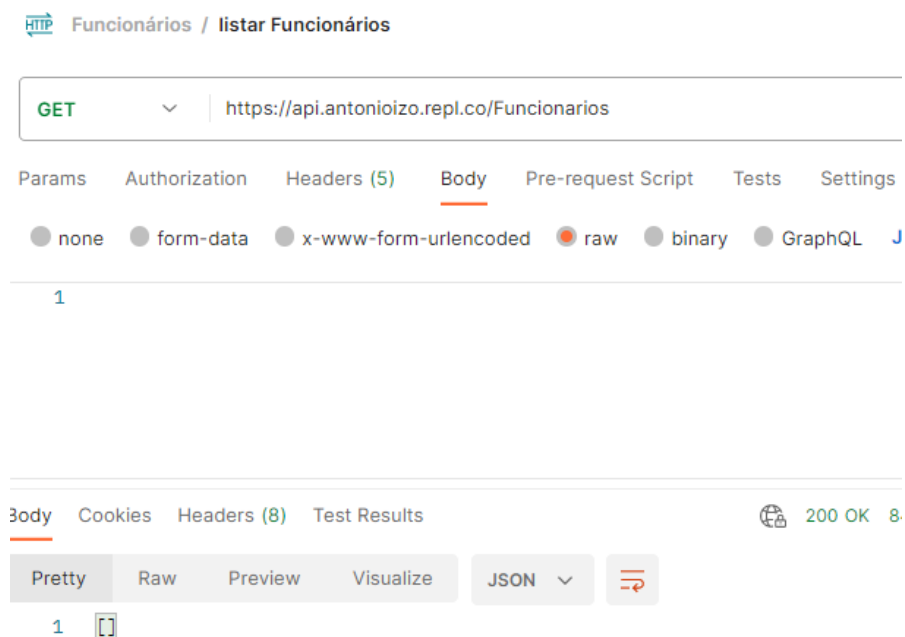
Passo 4: Agora vamos marcar a opção body, opção raw e selecionar nosso retorno que será um JSON.



Passo 5: Clique no botão send para verificar se foi reconhecido suas rotas.



Nosso olá mundo foi retornado sem problemas, vamos agora colocar a rota correta para o listar funcionários é uma rota /Funcionarios, vamos então acrescentar na requisição get e executar novamente.



Retornou um JSON [] sem valores. Está correto, ainda não incluímos objetos na nossa base.

Requisição Post

Para criar a requisição Post, faça o mesmo procedimento da requisição anterior mudando o tipo de requisição para **post**, o nome para **cadastar funcionários** e o endereço para nossa rota **\Cadastrar**.

HTTP Funcionários / Cadastrar Funcionários

POST ▼ | https://api.antonioizo.repl.co/Cadastrar

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	Key	Value	D
	Key	Value	D

Precisamos agora cadastrar os valores que serão inseridos na tabela funcionarios do banco de dados.

HTTP Funcionários / Cadastrar Funcionários

POST ▼ | https://api.antonioizo.repl.co/Cadastrar

Params Authorization Headers (6) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value	De:
	Key	Value	De:

Como incluímos na nossa API o comando que só aceitaríamos dados codificados em formato JSON, vamos utilizar a inclusão via **x-www-form-urlencoded** ou via **json**.

```
const express = require('express');
const app = express();
app.use(express.urlencoded({ extended: true }));
app.use(express.json());

//BIBLIOTECAS/MODULOS UTILIZADOS
```

Vou mostrar as duas opções abaixo.

Utilizando x-www-form-urlencoded

Selecione body e depois a opção x-www-form-urlencoded. Coloque os campos da tabela e os valores que serão inseridos.

HTTP Funcionários / Cadastrar Funcionários

POST <https://api.antonioizo.repl.co/Cadastrar>

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

☐ none
 ☐ form-data
 ☒ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

	Key	Value
<input checked="" type="checkbox"/>	nome	Antonio Izo
<input checked="" type="checkbox"/>	endereco	Rua central, Cachoeiro
<input checked="" type="checkbox"/>	telefone	28 000000000
<input checked="" type="checkbox"/>	email	antonio@reprograme-se.com
<input checked="" type="checkbox"/>	nascimento	1980-01-01
	Key	Value

Clique no botão send.

Body Cookies Headers (8) Test Results 200 OK 347 ms

Pretty Raw Preview Visualize JSON 

```

1 {
2   "message": "Cadastro realizado com sucesso!"
3 }
```

Utilizando JSON

Selecione a opção raw e escolha JSON no menu.

HTTP Funcionários / Cadastrar Funcionários

POST https://api.antonioizo.repl.co/Cadastrar

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL Text

1

Body Cookies Headers (8) Test Results 200 OK

Pretty Raw Preview Visualize JSON

Text
JavaScript
JSON
HTML
XML

Coloque o código JSON abaixo.

```
{
  "nome": "Carlos Drummond",
  "endereco": "Esquina da poesia, 10",
  "telefone": "21 000000000",
  "email": "drummond@reprograme-se.com",
  "nascimento": "1902-10-31"
}
```

HTTP Funcionários / Cadastrar Funcionários

POST https://api.antonioizo.repl.co/Cadastrar

Params Authorization Headers (7) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {
2   .... "nome": "Carlos Drummond",
3   .... "endereco": "Esquina da poesia, 10",
4   .... "telefone": "21 000000000",
5   .... "email": "drummond@reprograme-se.com",
6   .... "nascimento": "1902-10-31"
7 }
```

Clique no botão de send.

POST ▼ | https://api.antonioizo.repl.co/Cadastrar

Params Authorization Headers (7) **Body** • Pre-request Script Tests S

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```

1 {
2   "nome": "Carlos Drummond",
3   "endereco": "Esquina da poesia, 10",
4   "telefone": "21-000000000",
5   "email": "drummond@reprograme-se.com",
6   "nascimento": "1902-10-31"
7 }

```

body Cookies Headers (8) Test Results 🌐 200 OK

Pretty Raw Preview Visualize JSON ▼ ↔

```

1 {
2   "message": "Cadastro realizado com sucesso!"
3 }

```

Clique no botão de save e vamos retornar na requisição get para verificar se nossos objetos foram inseridos no banco.

POST ▼ | https://api.antonioizo.repl.co/Cadastrar Save request Send ▼

Params Authorization Headers (7) **Body** • Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

```

1 {
2   "nome": "Carlos Drummond",
3   "endereco": "Esquina da poesia, 10",
4   "telefone": "21-000000000",

```

Pretty Raw Preview Visualize JSON ▼ ↔

```

2 {
3   "matricula": 1,
4   "nome": "Antonio Izo",
5   "endereco": "Rua central, Cachoeiro",
6   "telefone": "28 000000000",
7   "email": "antonio@reprograme-se.com",
8   "nascimento": "1980-01-01 00:00:00.000 +00:00",
9   "createdAt": "2023-05-27 23:42:02.724 +00:00",
10  "updatedAt": "2023-05-27 23:42:02.724 +00:00"
11 },
12 {
13   "matricula": 2,
14   "nome": "Carlos Drummond",
15   "endereco": "Esquina da poesia, 10",
16   "telefone": "21 000000000",
17   "email": "drummond@reprograme-se.com",
18   "nascimento": "1902-10-31 00:00:00.000 +00:00",

```

Nossos arquivos foram inseridos no banco.

Requisição Put – Atualizar objetos

Vamos criar uma requisição put agora.

Devemos passar como parâmetro no endereço o a matrícula do funcionário que será alterado.

Você pode passar somente os campos que deseja alterar sem a necessidade de enviar todos os campos.

Vamos alterar o nome do funcionário com a matrícula 2 e seu email.

Podemos utilizar o JSON ou o x-www, vou realizar pelo json.



```
{  
  "nome": "Machado de Assis",  
  "email": "machado@reprograme-se.com"  
}
```

Antes de clicar no botão send, verifique se você passou o parâmetro no endereço da URL, **/Funcionarios/2**.

Overview | Funcionários | GET listar Fun | POST Cadastrar | PUT Alterar Fl

HTTP Funcionários / Alterar Funcionários

PUT | https://api.antonioizo.repl.co/Funcionarios/2

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests

none | form-data | x-www-form-urlencoded | raw | binary | Graph

```

1 {
2   ... "nome": "Machado de Assis",
3   ... "email": "machado@reprograme-se.com"
4 }

```

Body | Cookies | Headers (8) | Test Results

Pretty | Raw | Preview | Visualize | JSON |

```

1 {
2   "message": "Cadastro atualizado com sucesso!"
3 }

```

Vamos salvar nossa requisição put com o nome de alterar funcionários.

Requisição delete

A requisição delete é igual a put, porém precisamos passar somente a url com o id do funcionário que será excluído.

Irei excluir o funcionário id=2.

Funcionários / Excluir Funcionários

Save |

DELETE | https://api.antonioizo.repl.co/Funcionarios/2 | Send

Params | Authorization | Headers (5) | Body | Pre-request Script | Tests | Settings

Ctrl + Enter

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

HTTP Funcionários / Excluir Funcionários

DELETE ▼ | https://api.antonioizo.repl.co/Funcionarios/2

Params Authorization Headers (5) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (8) Test Results 200 OK

Pretty Raw Preview Visualize JSON ▼ ↻

```

1  {
2    "message": "Funcionário excluído com sucesso!"
3  }

```

Se tudo deu certo, temos uma estrutura de coleções no Postman igual a mostrada abaixo:

Reprograme_se New Import

Collections + ☰ ...

Environments

History

Funcionários

- GET listar Funcionários
- POST Cadastrar Funcionários
- PUT Alterar Funcionários
- DEL Excluir Funcionários

Finalizamos os testes de acesso dos serviços da nossa API. Criamos uma API do zero com o seu CRUD e consumimos os seus serviços via uma aplicação de testes.

Um abraço e até a próxima semana.