

MÉTODO	ROTA	DESCRIÇÃO	OBS	RETORNO
GET	http://localhost:3000/company	Este método retorna todas empresas existentes na base de dados local.	ESTE MÉTODO NÃO REQUER QUALQUER PARÂMETRO OU PASSAR UM BODY	Ex. : <pre>{ "message": [{ "id": 14, "cep": "70842970", "state": "DF", "city": "Brasília", "street": "EQN 204/404 Lote Único", "neighborhood": "Asa Norte", "service": "viacep", "createdAt": "2021-08-19T11:13:22.000Z", "updatedAt": "2021-08-19T11:13:22.000Z" }] }</pre>
POST	http://localhost:3000/company	Este método permite cadastrar uma nova empresa através de CEP enviado na requisição pelo usuário	ESTE MÉTODO REQUER UM BODY no seguinte formato: <pre>{ "cep": "243562-647", "state": "Brazil", "city": "Angola", "street": "Rio Seco", "neighborhood": "Brazil", "service": "CAE", "location": { "type": "Ponto",</pre>	Ex. <pre>{ "message": "Company Crated Successfully", "companies": { "id": 15, "cep": "Kaleeeues", "state": "Brazil", "city": "Angola", "street": "Rio Seco", "neighborhood": "Brazil", "service": "CAE",</pre>

			<pre> "coordinates":{ "latitude":"-523667782", "longitude":"63487" } } </pre>	<pre> "updatedAt": "2021-08-19T11:28:30.942Z", "createdAt": "2021-08-19T11:28:30.942Z" } } </pre>
PUT	http://localhost:3000/empresas/{params}	este método permite atualizar os dados de uma empresa existente no banco de dados, apenas permite usuários autenticados	é requerido um parâmetro que definirá que empresa a ser atualizado, e os dados a serem enviados no body para devida atualização	<pre> { "message": "Data updated successfully!", "empresa": [1] } </pre>
DELETE	http://localhost:3000/empresas/{params}	Este método, passado um parâmetro, permite eliminar um registro no banco de dados e apenas permite usuário autenticado	Neste método apenas é passado o parâmetro ou o ID da empresa e será deletado.	<pre> { "message": "Register deleted successfully", "resultado": 1 } </pre>
POST	http://localhost:3000/user/signUp	Esta rota, permite criar um usuário para que possa ter acesso aos dados cadastrados	<pre> { "name":"Marcos Mandamba", "email":"marcosmandamba@gmail.com", "password": "marcos2278@nmx" } </pre>	

REQUISITOS PARA RODAR O BACKEND:

- 1- Ter o nodejs instalado**
- 2- Servidor MySql para base de dados**

Procedimentos para execução:

-Executar o comando npm install no projecto Backend para instalação das dependências e npm start para iniciar o servidor

-Executar o comando npx sequelize db:migrate, para criação das migrations no banco de dados

REQUISITOS PARA RODAR O FRONTEND:

Tecnologia usada para o consumo das APIs: Angular frame work.

Npm: Para instalação das dependências

Procedimentos de execução: Ter o node instalado

Execução dos seguintes comandos:

npm install : para instalação das dependências usadas no projecto.

ng serve – para executar o projecto FrontEnd

INFORMAÇÃO ADICIONAL

- 1- Qualquer usuário poderá fazer o cadastramento de empresa apartir de um cep válido;
- 2- Para listar as empresas, o usuário deverá estar cadastrado e autenticado, caso contrário a API não dará acesso aos mesmos dados, por isso está disponível a rota ou API:

<http://localhost:3000/user/signUp> - Para o cadastramento de um usuário

corpo: exemplo.

```
{  
  "name": "Marcos Mandamba",  
  "email": "marcosmandamba@gmail.com",  
  "password": "12345"
```

} A password é incryptada, por isso deverá conservá-la melhor.

SERVIDOR UTILIZADO

1. MYSQL

2. Deverá ser criado uma base de dados **Empresa** e no projecto do backend, deverá executar o comando '**npx sequelize db:migrate**' para criar as migrations definidas no código