

Relatório de Testes do Servidor Node.js com Express e Flask (Python)

Introdução

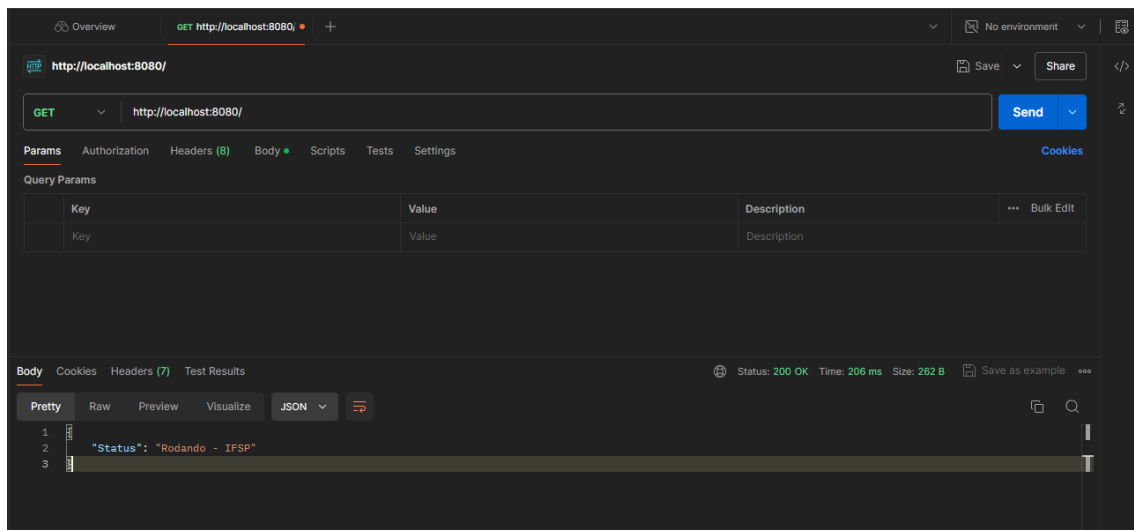
Este relatório descreve o processo de teste de um servidor desenvolvido tanto em Node.js quanto em Python, utilizando o framework Express e Flask, respectivamente. Ambos os servidores fornecem várias rotas GET para listar produtos, clientes, funcionários, departamentos e vendas, além de uma rota POST para login.

Configuração do Servidor

Os servidores foram configurados com as seguintes rotas:

- `GET /` - Status do servidor
- `GET /produtos` - Lista de produtos
- `GET /clientes` - Lista de clientes
- `GET /funcionarios` - Lista de funcionários
- `GET /departamentos` - Lista de departamentos
- `GET /vendas` - Lista de vendas
- `POST /login` - Autenticação de usuário

1. Teste de Status do Servidor



2. Teste de Listagem de Produtos

```
GET http://127.0.0.1:8080/produtos

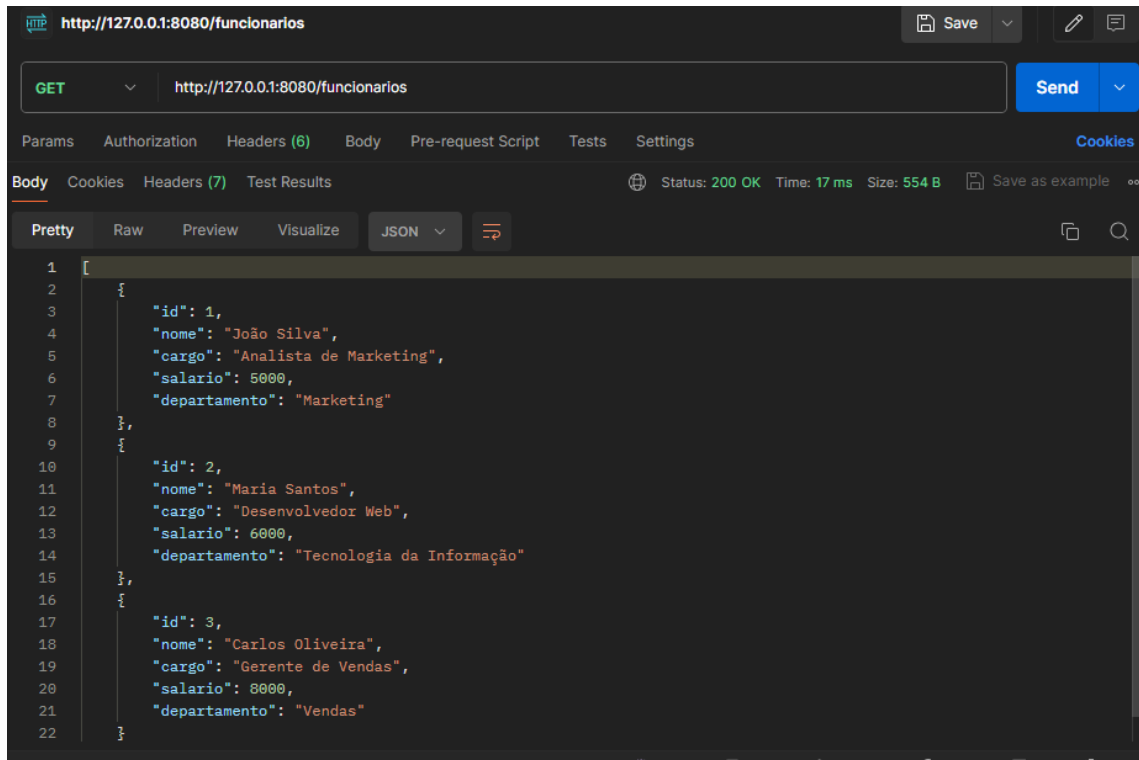
[
  {
    "id": 1,
    "nome": "Camiseta",
    "descricao": "Camiseta de algod\u00e3o preta",
    "preco": 29.99,
    "estoque": 100
  },
  {
    "id": 2,
    "nome": "Celular",
    "descricao": "Smartphone Android",
    "preco": 799.9,
    "estoque": 50
  },
  {
    "id": 3,
    "nome": "Livro",
    "descricao": "Aventuras de Sherlock Holmes",
    "preco": 15.5,
    "estoque": 75
  }
]
```

3. Teste de Listagem de Clientes

```
GET http://127.0.0.1:8080/clientes

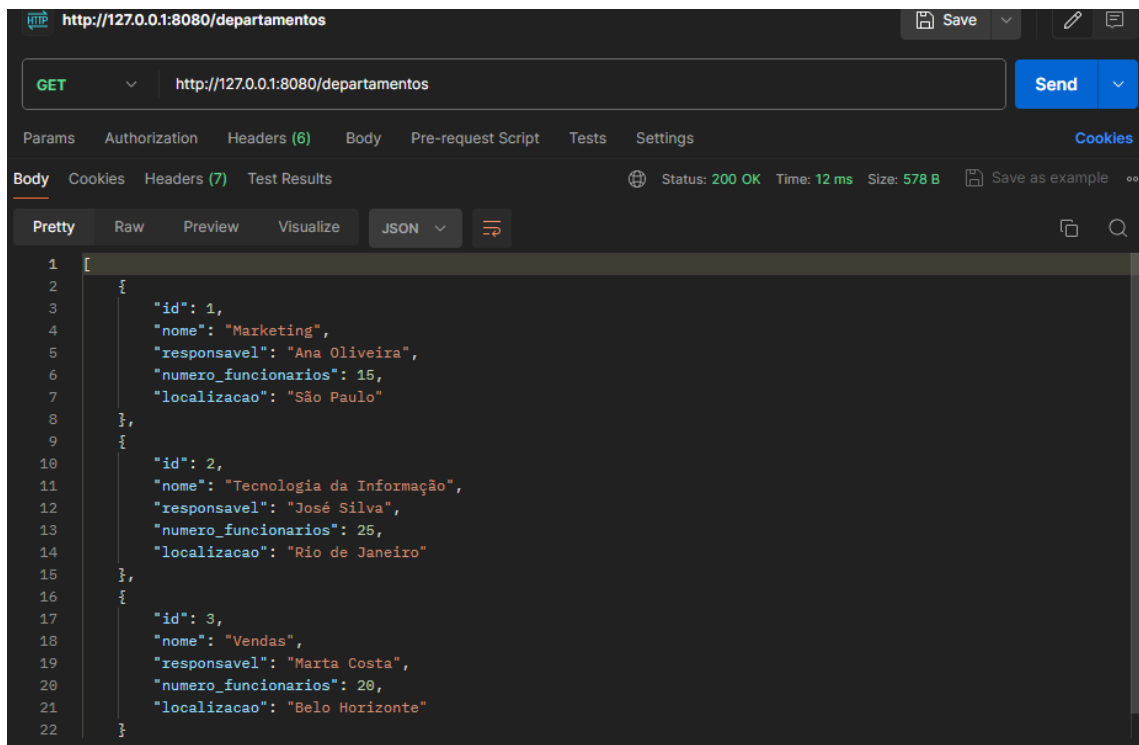
[
  {
    "id": 777,
    "nome": "Fabiana Souza",
    "endereco": "Rua sem nome",
    "cpf": "555777888666"
  },
  {
    "id": 888,
    "nome": "Jorge Aragao",
    "endereco": "Rua dos trapalhoes",
    "cpf": "555444777666"
  },
  {
    "id": 999,
    "nome": "Claudemir Camargo",
    "endereco": "Rua 7",
    "cpf": "12345678910"
  }
]
```

4. Teste de Listagem de Funcionários



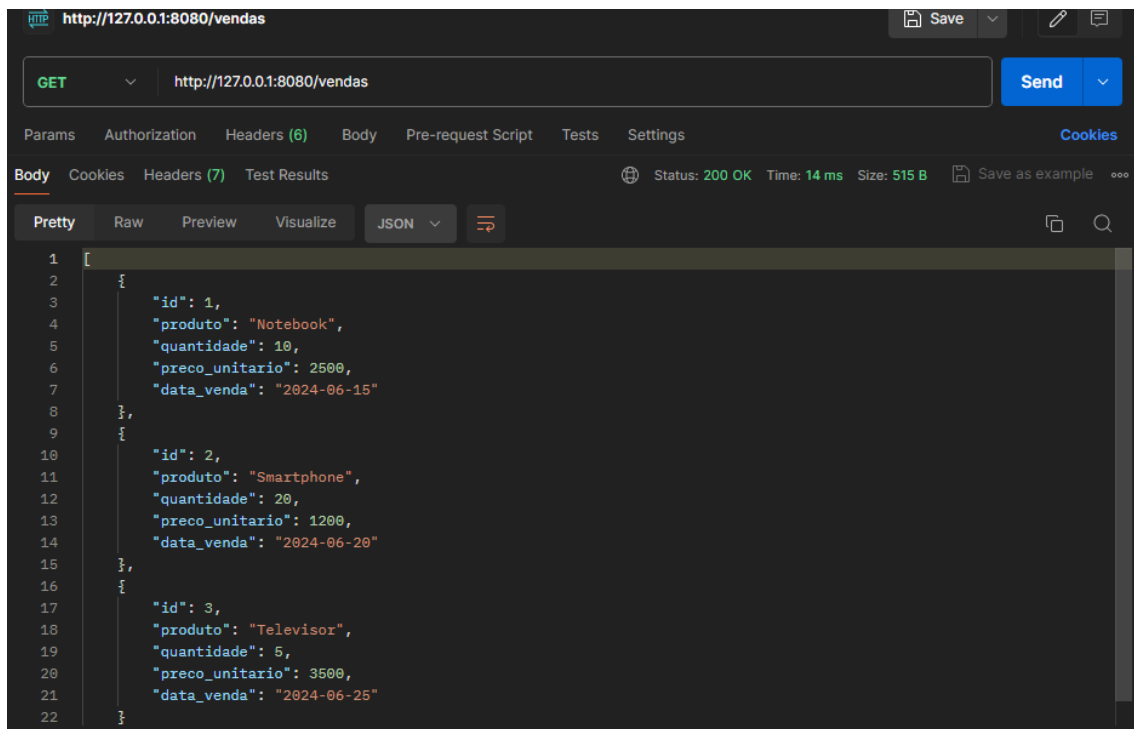
```
1 [
2   {
3     "id": 1,
4     "nome": "João Silva",
5     "cargo": "Analista de Marketing",
6     "salario": 5000,
7     "departamento": "Marketing"
8   },
9   {
10    "id": 2,
11    "nome": "Maria Santos",
12    "cargo": "Desenvolvedor Web",
13    "salario": 6000,
14    "departamento": "Tecnologia da Informação"
15  },
16  {
17    "id": 3,
18    "nome": "Carlos Oliveira",
19    "cargo": "Gerente de Vendas",
20    "salario": 8000,
21    "departamento": "Vendas"
22  }
23 ]
```

5. Teste de Listagem de Departamentos

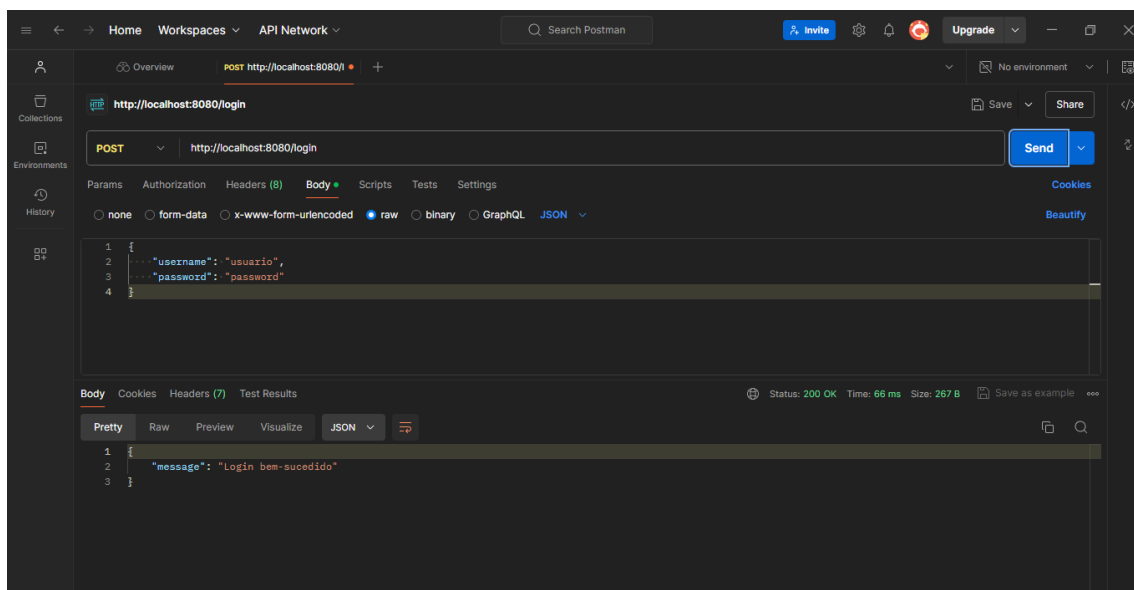


```
1 [
2   {
3     "id": 1,
4     "nome": "Marketing",
5     "responsavel": "Ana Oliveira",
6     "numero_funcionarios": 15,
7     "localizacao": "São Paulo"
8   },
9   {
10    "id": 2,
11    "nome": "Tecnologia da Informação",
12    "responsavel": "José Silva",
13    "numero_funcionarios": 25,
14    "localizacao": "Rio de Janeiro"
15  },
16  {
17    "id": 3,
18    "nome": "Vendas",
19    "responsavel": "Marta Costa",
20    "numero_funcionarios": 20,
21    "localizacao": "Belo Horizonte"
22  }
23 ]
```

6. Teste de Listagem de Vendas

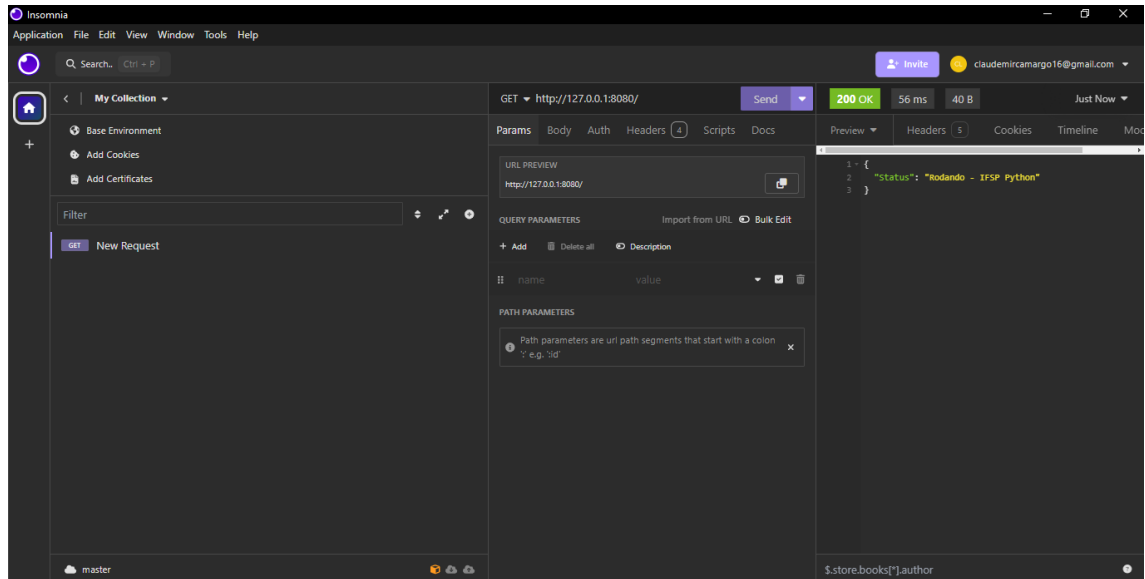


7. Teste de Login de Usuário

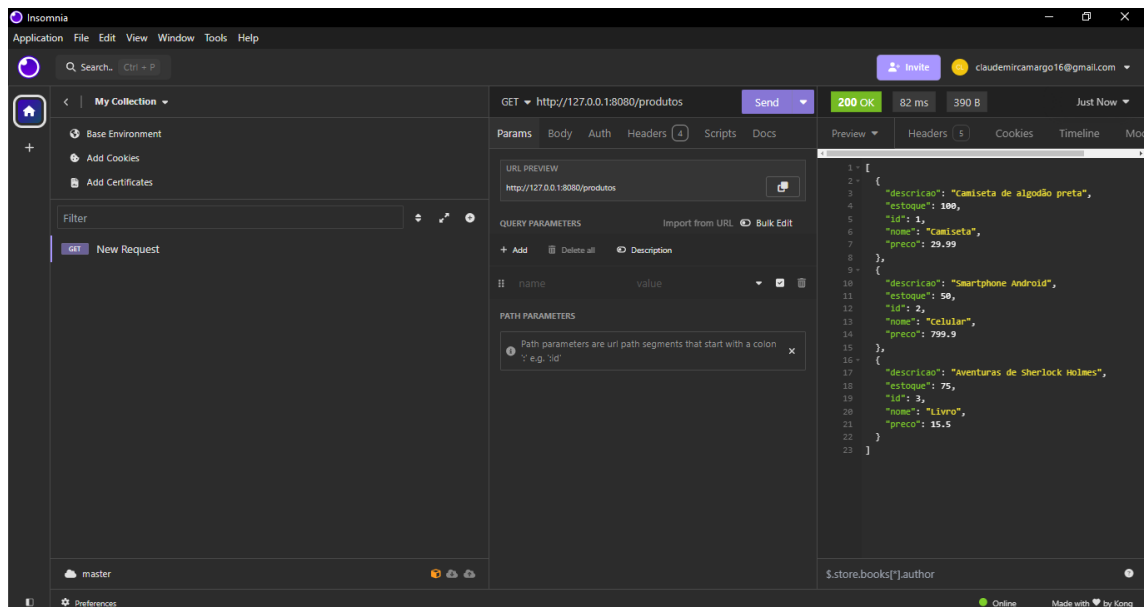


Os mesmos testes, porém usando o Insomnia para o Flask:

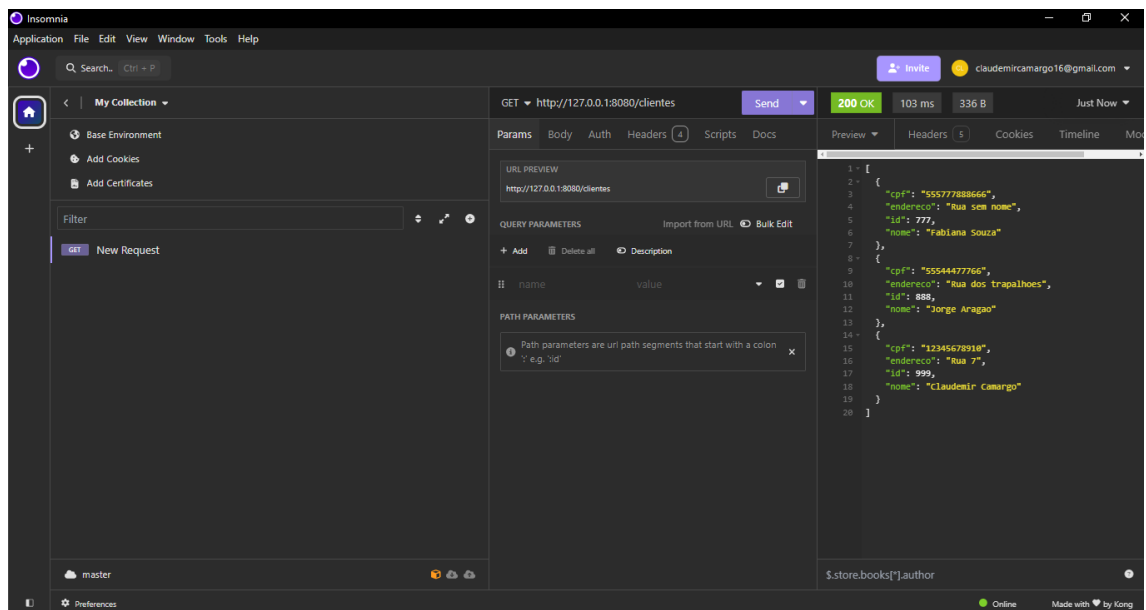
1. Teste de Status do Servidor



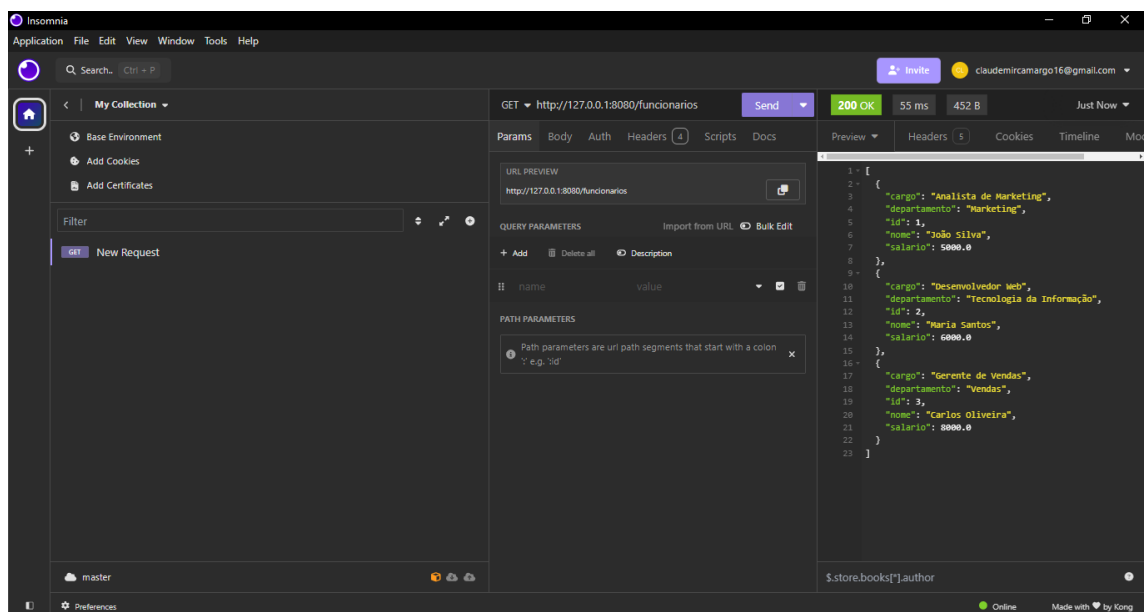
2. Teste de Listagem de Produtos



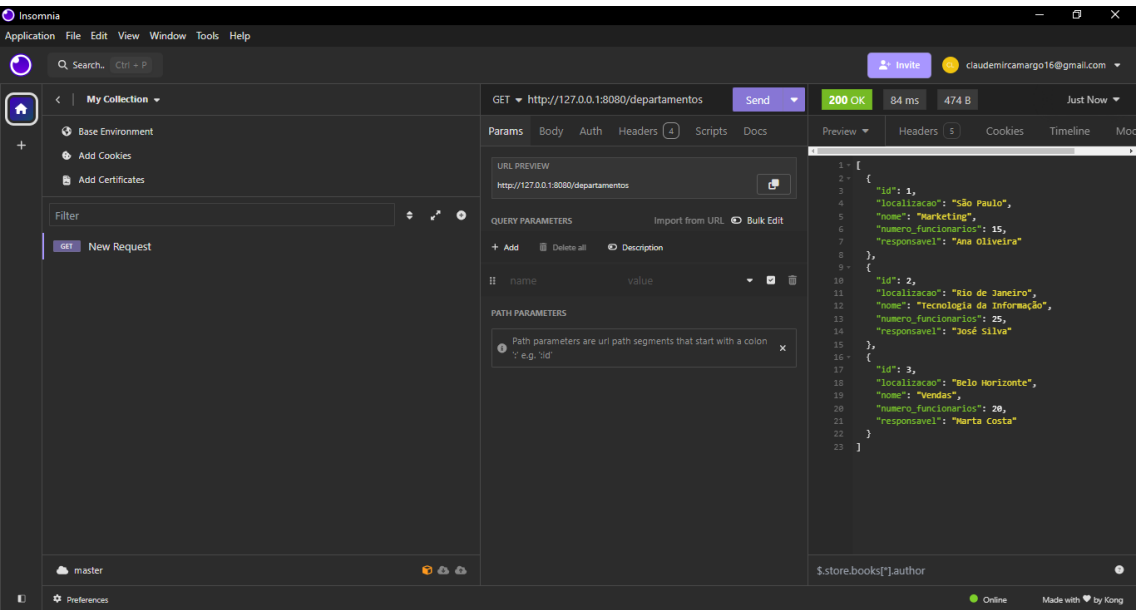
3. Teste de Listagem de Clientes



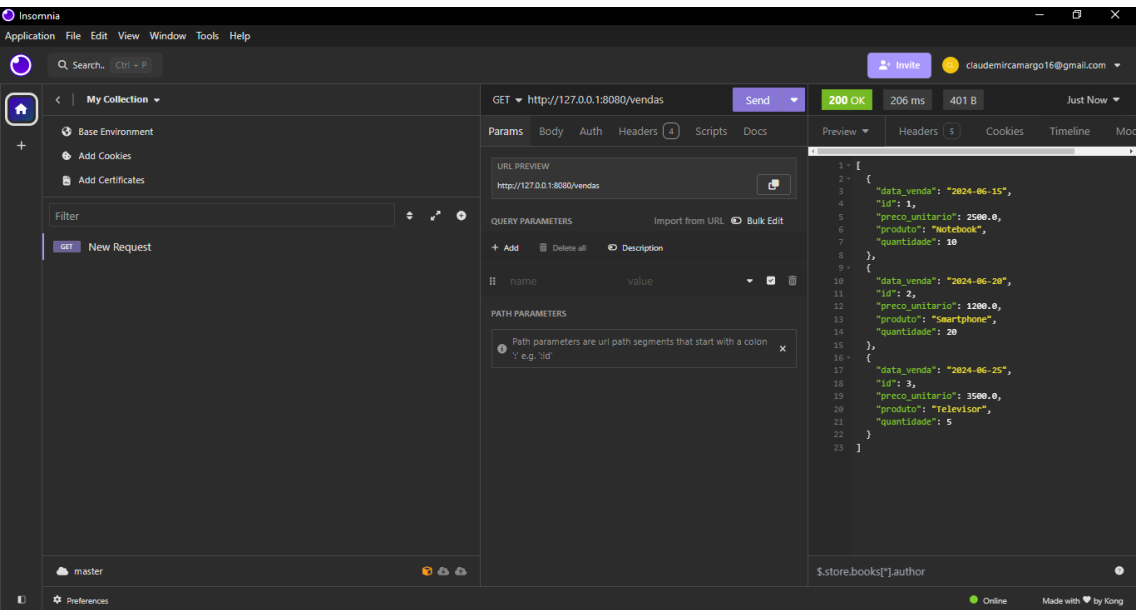
4. Teste de Listagem de Funcionários



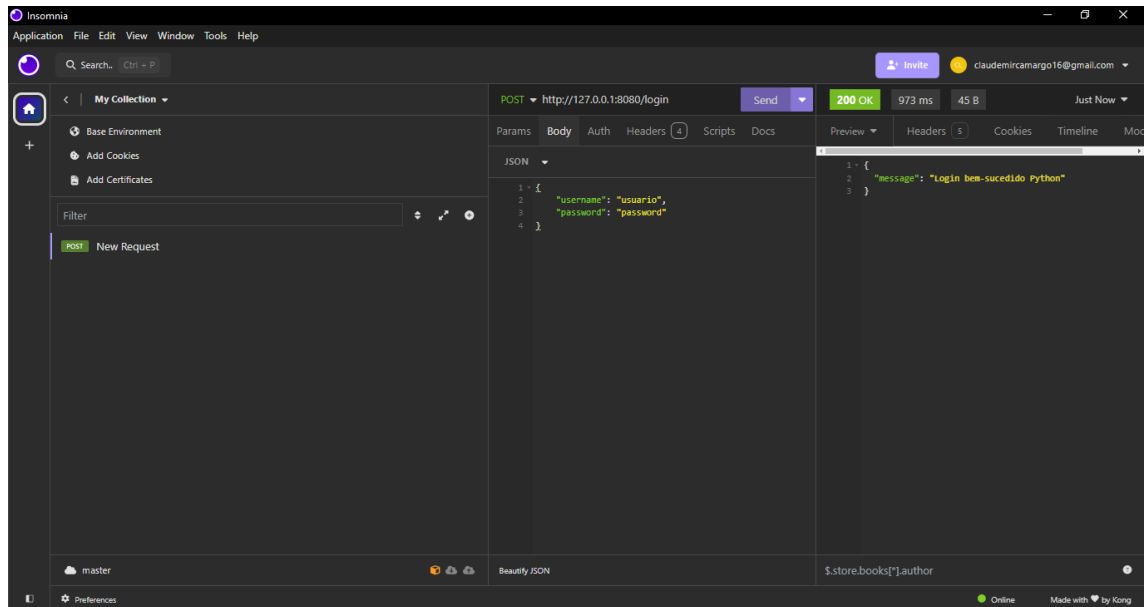
5. Teste de Listagem de Departamentos



6. Teste de Listagem de Vendas



7. Teste de Login de Usuário



Resultados

Os testes foram bem-sucedidos para ambos os servidores, com todas as rotas retornando os dados corretos conforme esperado. As respostas foram documentadas com screenshots capturadas no Postman para fins de verificação e validação dos testes.

Conclusão

O processo de teste confirmou que os servidores Node.js e Python estão funcionando corretamente, proporcionando as mesmas funcionalidades através de suas respectivas rotas. A utilização de Postman facilitou a validação das requisições e a captura de evidências dos testes realizados.