Thought for 2 seconds```python project="nasajon-api-python" file="app.py"

...

```

```python project="nasajon-api-python" file="static/swagger.json"

...

```

I found some issues in the code block.

- expected `:` but instead found `{`

I will fix them.

```python project="nasajon-api-python" file="static/swagger.json"

...

```

```python project="nasajon-api-python" file="client\_simulation.py"

...

```

```python project="nasajon-api-python" file="requirements.txt"

...

```

```python project="nasajon-api-python" file="README.md"

...

```

## Executando a API

```shellscript

python app.py

```

A API estará disponível em [http://localhost:5000](http://localhost:5000)

## Documentação Swagger

Acesse a documentação Swagger em [http://localhost:5000/api/docs](http://localhost:5000/api/docs)

## Simulação de Cliente

Para testar a API, execute o script de simulação:

```shellscript

python client\_simulation.py

```

## Autenticação

Todas as requisições devem incluir o header `X-API-Key` com a chave de API.

Por padrão, a chave é `api\_key\_simulada`, mas pode ser alterada através da variável de ambiente `NASAJON\_API\_KEY`.

## Exemplos de Requisições

### Financeiro - Criar Lançamento

```shellscript

curl -X POST http://localhost:5000/api/financeiro/lancamentos \

-H "Content-Type: application/json" \

-H "X-API-Key: api\_key\_simulada" \

-d '{

"identificador": "LF001",

"data": "2024-02-21",

"valor": 1000.00,

"tipo": "RECEITA",

"conta": "1001",

"descricao": "Venda à vista"

}'

```

### Contábil - Criar Lançamento

```shellscript

curl -X POST http://localhost:5000/api/contabil/lancamentos \

-H "Content-Type: application/json" \

-H "X-API-Key: api\_key\_simulada" \

-d '{

"identificador": "LC001",

"data": "2024-02-21",

"valor": 1000.00,

"debito": "1001",

"credito": "2001",

"historico": "Venda à vista"

}'

```

### Fiscal - Emitir Nota Fiscal

```shellscript

curl -X POST http://localhost:5000/api/fiscal/notas-fiscais \

-H "Content-Type: application/json" \

-H "X-API-Key: api\_key\_simulada" \

-d '{

"numero": "NF001",

"data": "2024-02-21",

"valor": 1000.00,

"cliente": {

"cnpj": "12345678000100",

"nome": "Cliente Exemplo"

},

"itens": [

{

"codigo": "001",

"descricao": "Produto A",

"quantidade": 1,

"valorUnitario": 1000.00

}

]

}'

```

```plaintext

## Instruções de Uso

Esta simulação de API para integração com o Nasajon permite testar todos os módulos do sistema sem precisar de uma conexão real com o Nasajon. Siga os passos abaixo para utilizar:

1. \*\*Instale as dependências\*\*:

```bash

pip install -r requirements.txt

```

2. \*\*Inicie a API\*\*:

```shellscript

python app.py

```

3. \*\*Execute a simulação de cliente\*\*:

```shellscript

python client\_simulation.py

```

4. \*\*Acesse a documentação Swagger\*\*:

Abra seu navegador e acesse `http://localhost:5000/api/docs`

A simulação inclui:

1. \*\*Módulo Financeiro\*\*:

1. Criação de lançamentos financeiros

2. Consulta de lançamentos

2. \*\*Módulo Contábil\*\*:

1. Lançamentos contábeis

2. Partidas dobradas

3. \*\*Módulo Fiscal\*\*:

1. Emissão de notas fiscais

2. Gestão de impostos

4. \*\*Módulo RH\*\*:

1. Cadastro de funcionários

2. Folha de pagamento

5. \*\*Módulo Compras\*\*:

1. Pedidos de compra

2. Gestão de fornecedores

6. \*\*Módulo Vendas\*\*:

1. Pedidos de venda

2. Gestão de clientes

7. \*\*Módulo Estoque\*\*:

1. Cadastro de produtos

2. Movimentação de estoque

8. \*\*Módulo Patrimônio\*\*:

1. Cadastro de bens

2. Depreciação

Todos os dados são armazenados em memória durante a execução da aplicação, simulando um banco de dados temporário.