

Casos de Teste
Sistema de Cadastro de Contribuintes
Versão 1.0

Histórico das alterações

Data	Versão	Descrição	Autor (a)
16/10/2024	1.0	<i>Release</i> Inicial	Lucas Haiaiel A. Zacarias

1. Glossário

Entrada	Definição
CN	Cenário de Teste
CT	Caso de Teste
HU	História de Usuário

HU 001 – Cadastro de Contribuintes

CN_01 Login da Aplicação

API para Cenário: <http://testeqa.pge.ce.gov.br>

CT_01 Autenticação de Usuário Válido: **SUCESSO**

- **Descrição:** Autenticar no sistema com um usuário já cadastrado e válido no sistema.
- **Pré-Condições:** Deve existir um usuário previamente cadastrado e válido no sistema.

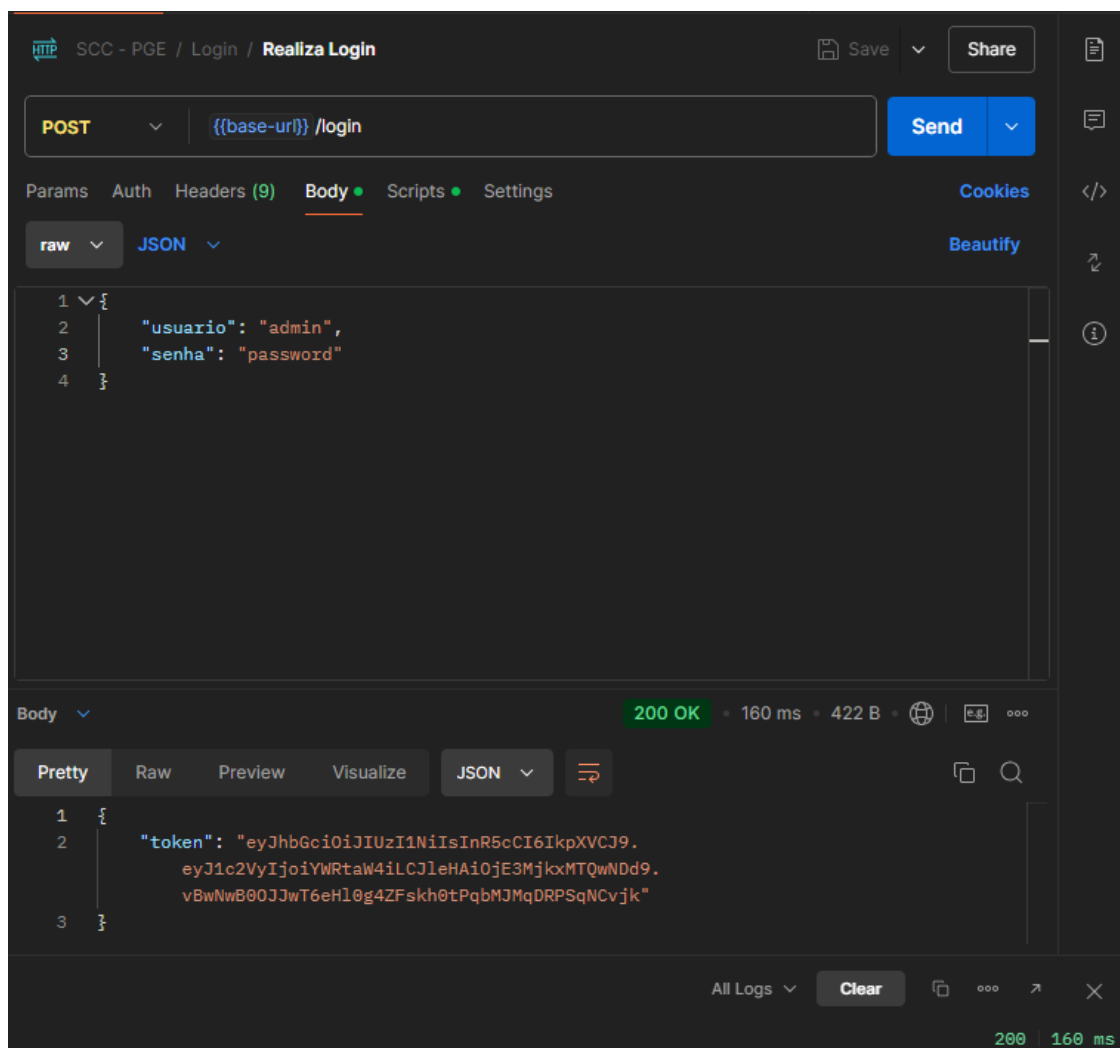
Dado que seja realizado uma requisição do tipo POST no *endpoint* /login com o corpo da requisição (body) abaixo em formato JSON:

```
{ "usuário": "admin", "senha": "password" }
```

Então o sistema deve retornar um status do tipo 200

E deve retornar uma resposta formatada no padrão JSON com o modelo abaixo:

```
{ "token": "seu_token_aqui" }
```



CT_02 Autenticação de Usuário Inválido: **SUCESSO**

- **Descrição:** Tentar autenticar no sistema com um usuário não cadastrado no sistema.

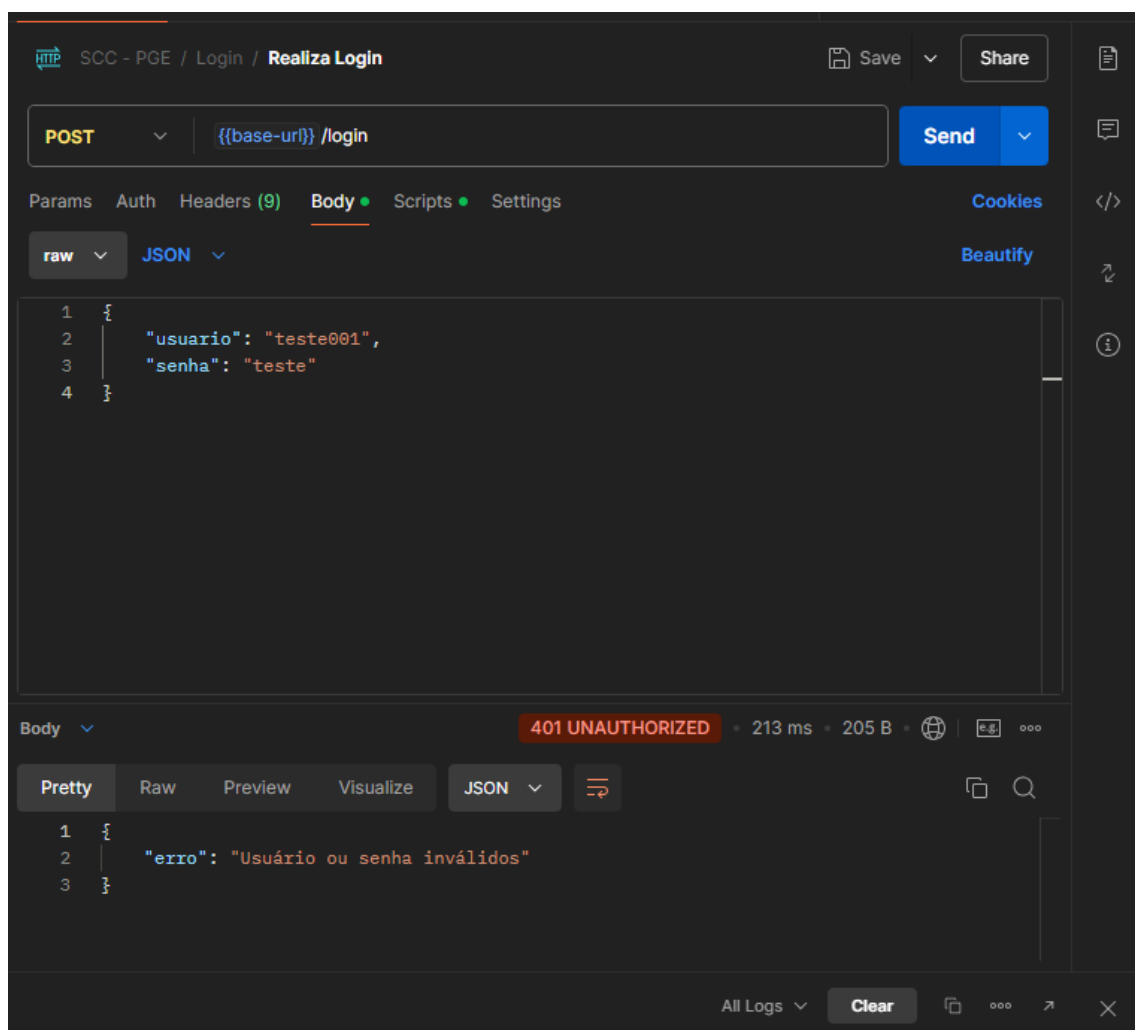
Dado que seja realizado uma requisição do tipo POST no *endpoint* /login com o corpo da requisição (body) abaixo em formato JSON:

```
{ "usuário": "teste001", "senha": "teste" }
```

Então o sistema deve retornar um status do tipo 401

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{ "erro": "Usuário ou senha inválidos" }
```



CN_02 Cadastro de Contribuinte

API para Cenário: <http://testeqa.pge.ce.gov.br>

CT_01 Requisição sem Token Informado: **SUCESSO**

- **Descrição:** Tentar realizar a ação de cadastro de um contribuinte sem enviar um token de autenticação.

Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "12141419000", "nome": "Teste 001", "data_nascimento": "1990-01-01",  
"nome_mae": "Maria da Silva"}
```

E não seja informado o token de autenticação;

Então o sistema deve retornar um status do tipo 401

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Token de acesso é necessário"}
```

The screenshot shows a REST client interface with the following details:

- URL:** `{{base-url}} /contribuintes`
- Method:** POST
- Headers:** 10 headers are listed, including `x-access-token` with value `{{x-access-token}}`.
- Status:** 401 UNAUTHORIZED
- Response Body (JSON):**

```
{  
  "erro": "Token de acesso é necessário"  
}
```

CT_02 Requisição com Token Inválido: **SUCESSO**

- **Descrição:** Tentar realizar a ação de cadastro de um contribuinte enviando um token de autenticação inválido

Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

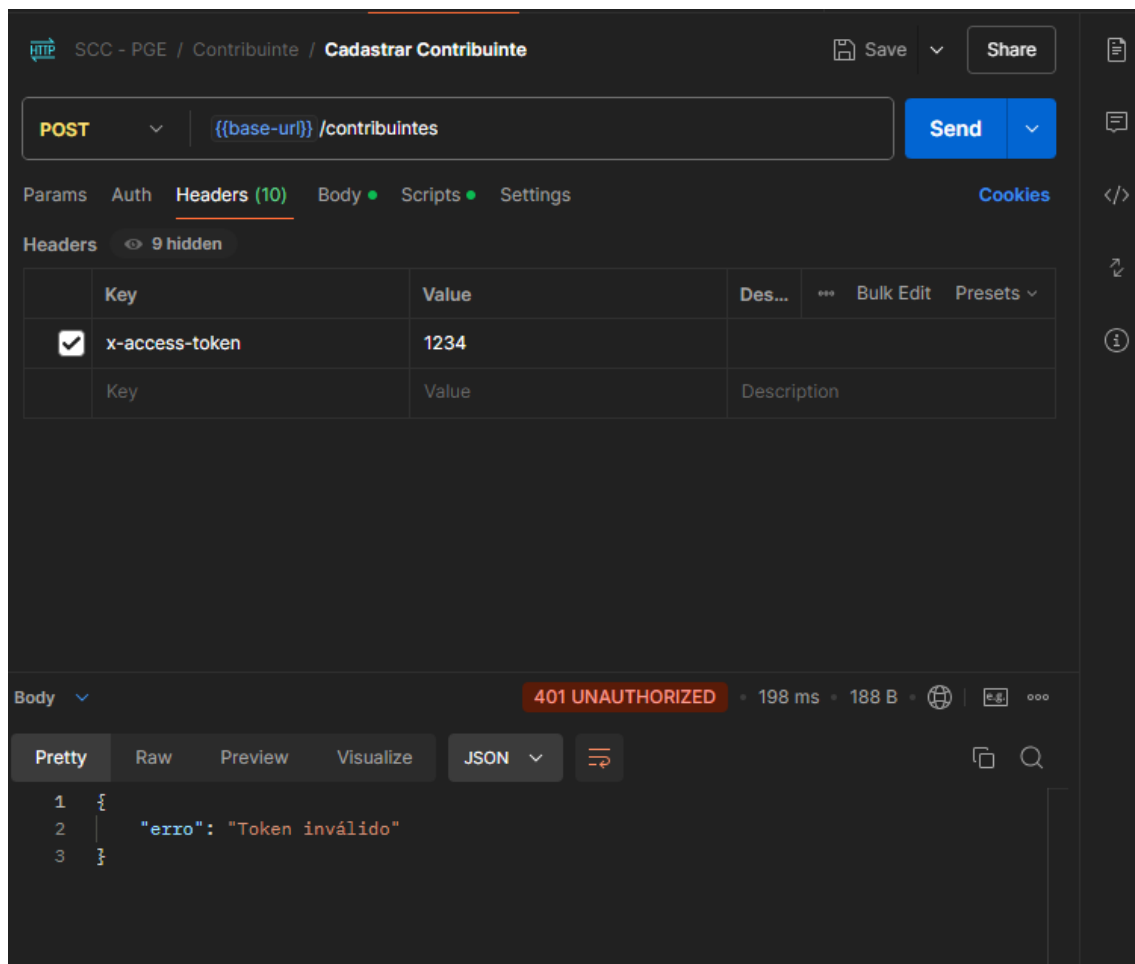
```
{"cpf": "12141419000", "nome": "Teste 001", "data_nascimento": "1990-01-01",  
"nome_mae": "Maria da Silva"}
```

E seja informado o token de autenticação inválido;

Então o sistema deve retornar um status do tipo 401;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Token inválido"}
```



CT_03 Campo Nome Obrigatório: **SUCESSO**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório “nome” não apresentado na requisição.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

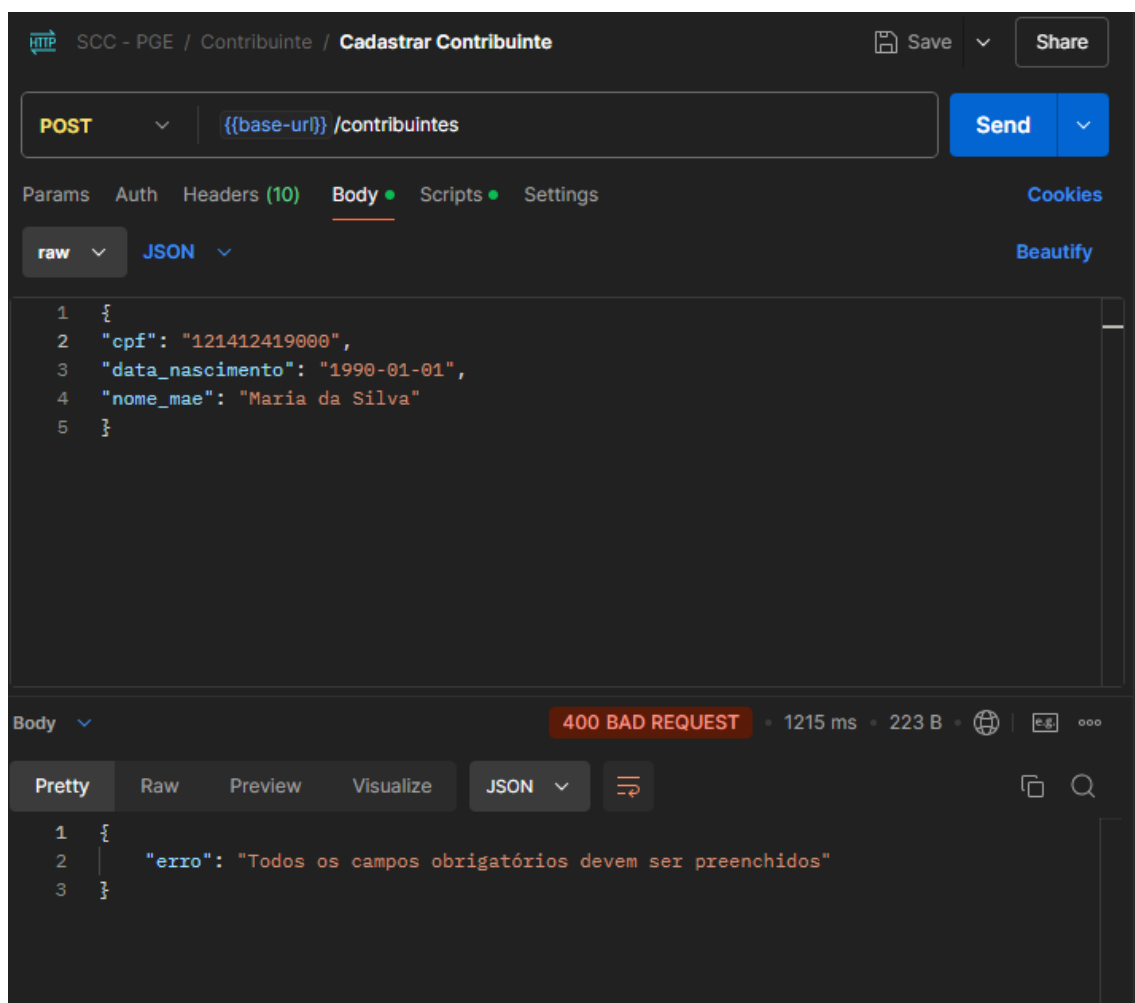
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "12141419000", "data_nascimento": "1990-01-01", "nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Todos os campos obrigatórios devem ser preenchidos"}
```



CT_04 Campo Data de Nascimento Obrigatório: **SUCESSO**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório “data_nascimento” não apresentado na requisição.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

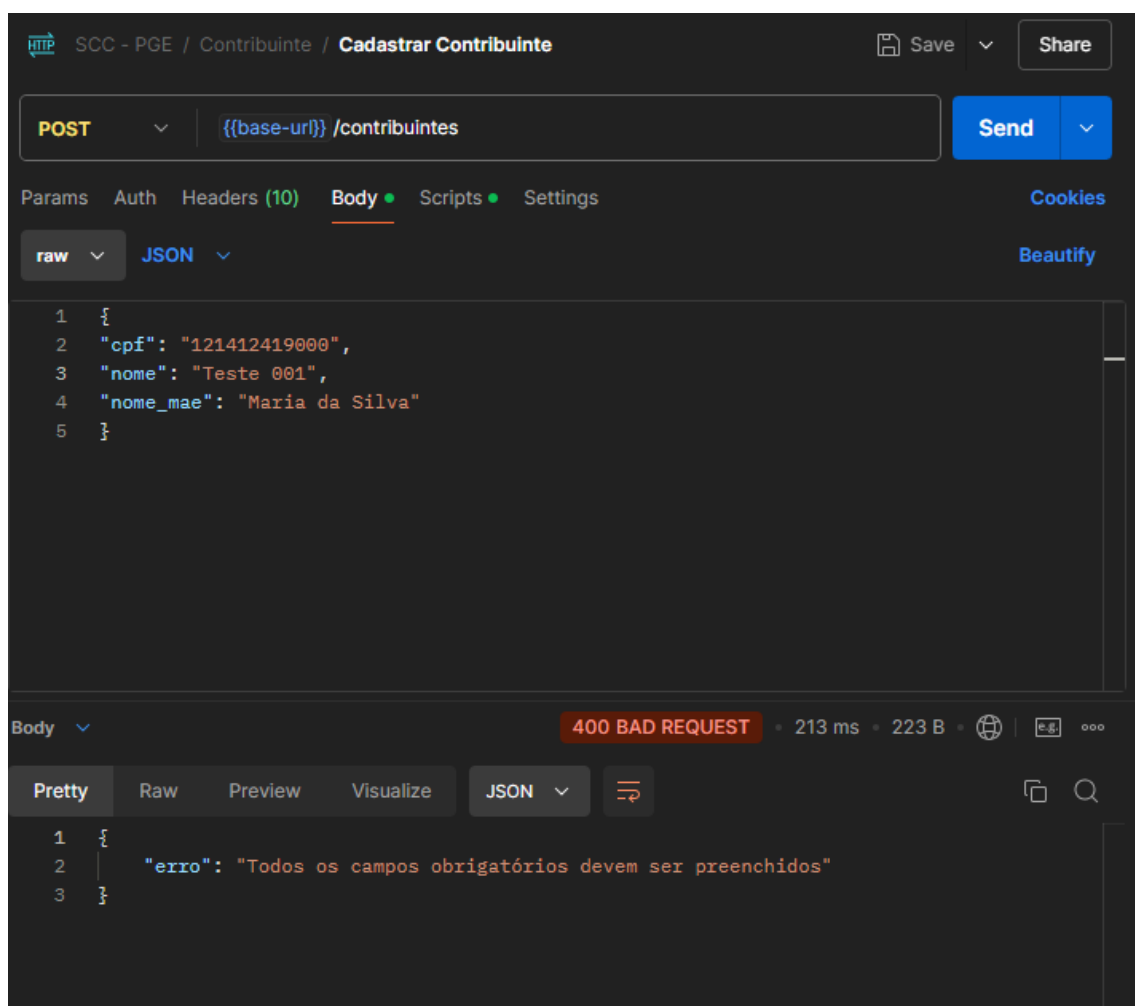
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "121412419000", "nome": "Teste 001", "nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Todos os campos obrigatórios devem ser preenchidos"}
```



CT_05 Campo Nome da Mãe Obrigatório: **SUCESSO**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório “nome_mae” não apresentado na requisição.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

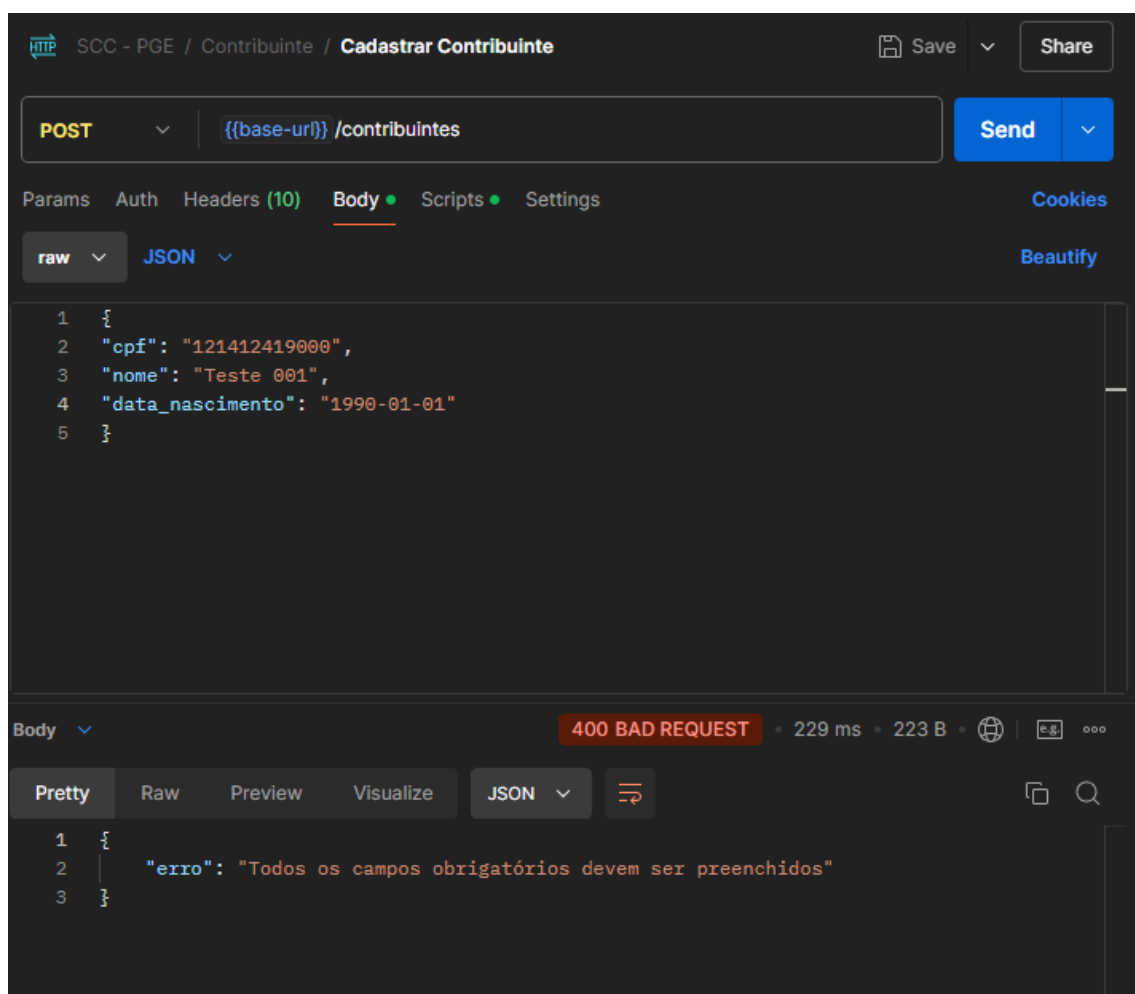
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "121412419000", "nome": "Teste 001", "data_nascimento": "1990-01-01"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Todos os campos obrigatórios devem ser preenchidos"}
```



CT_06 Validar quantidade de caracteres na padronização do CPF: **FALHA**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório “CPF” fora dos padrões requeridos:
 - Ter obrigatoriamente onze dígitos.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

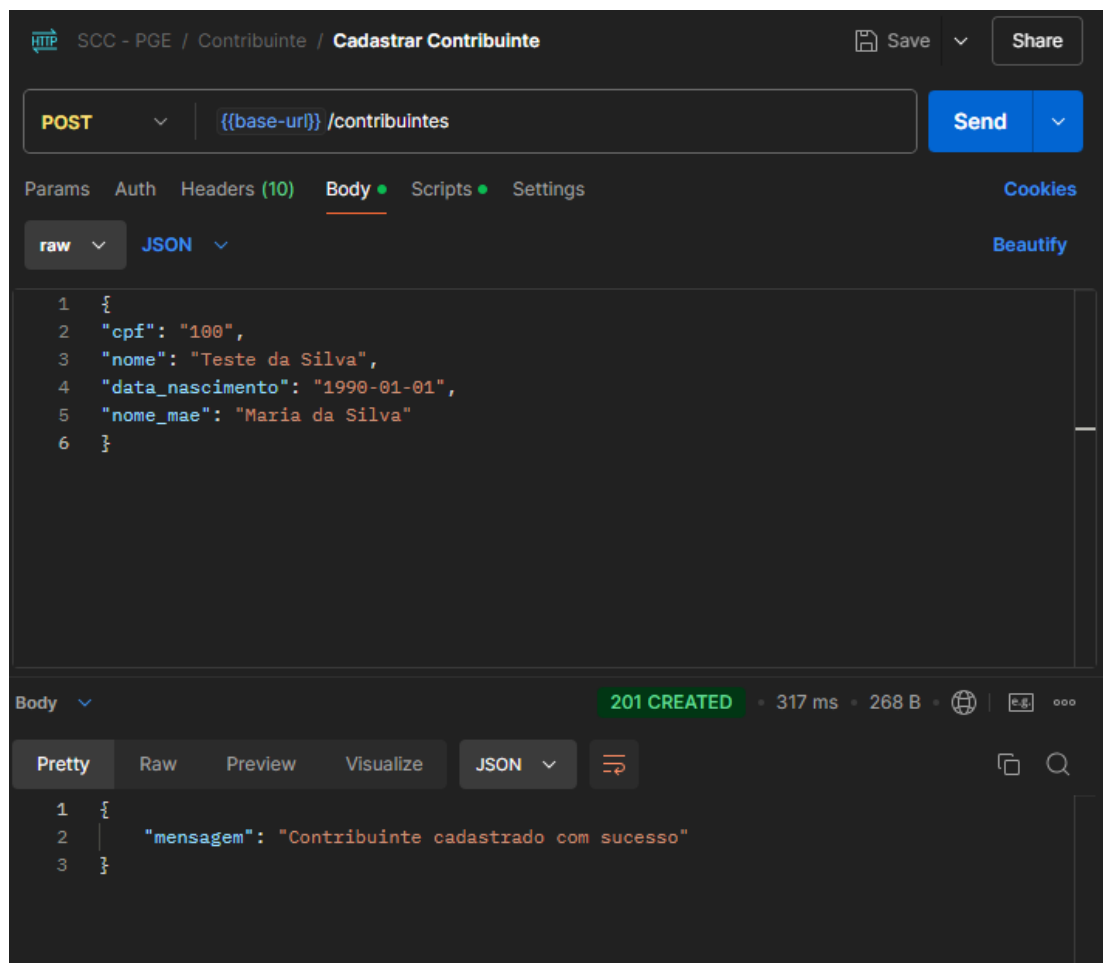
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "100", "nome": "Teste da Silva", "data_nascimento": "1990-01-01",  
"nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "CPF Inválido "}
```



CT_07 Validar padronização do CPF: **FALHA**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório “CPF” fora dos padrões requeridos:
 - Não realizar cálculo conforme Regulamentação da Receita Federal;
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

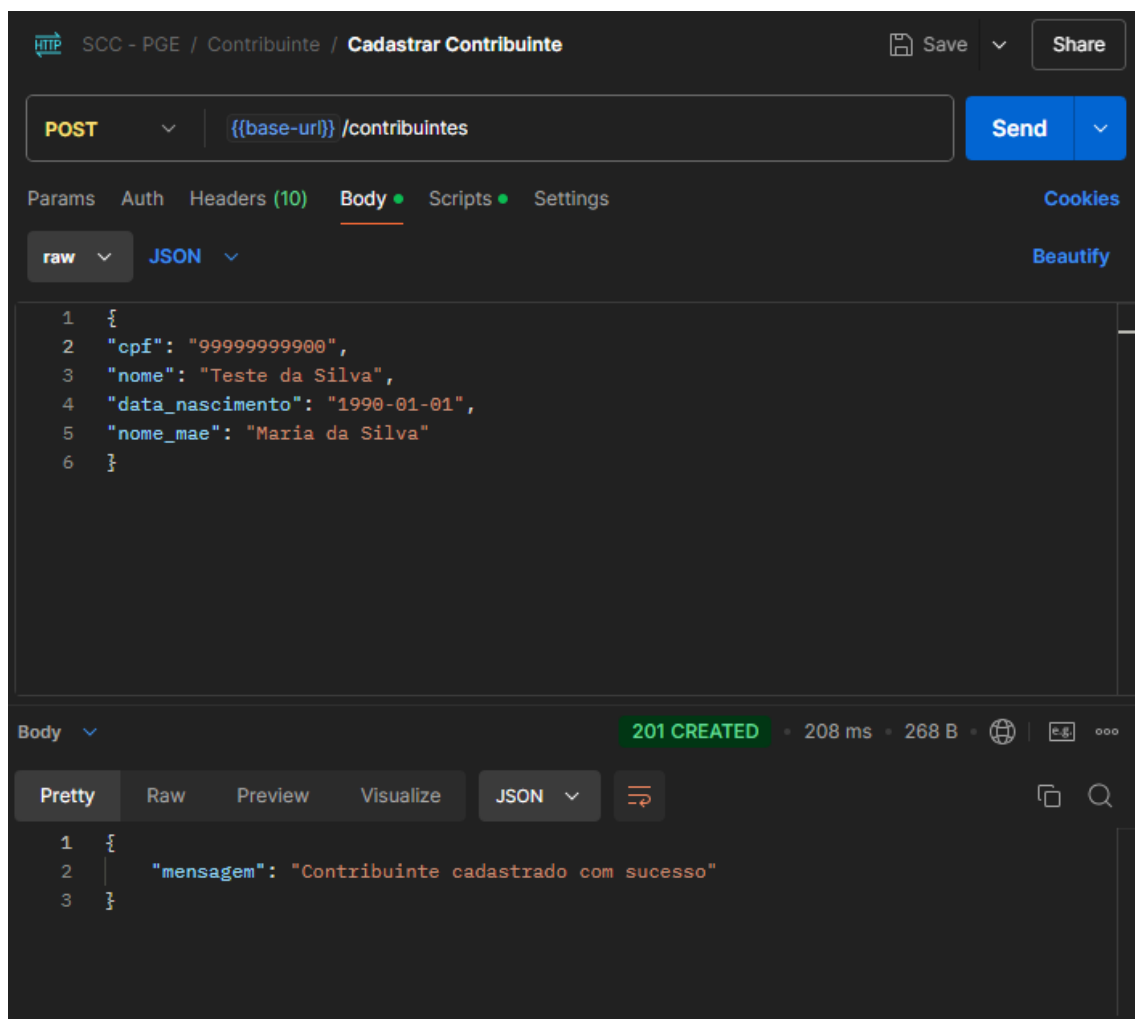
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "99999999900", "nome": "Teste da Silva", "data_nascimento": "1990-01-01",  
"nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "CPF Inválido "}
```



CT_08 Cadastrar Contribuinte: **SUCESSO**

- **Descrição:** Realizar o cadastro de um contribuinte no sistema.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

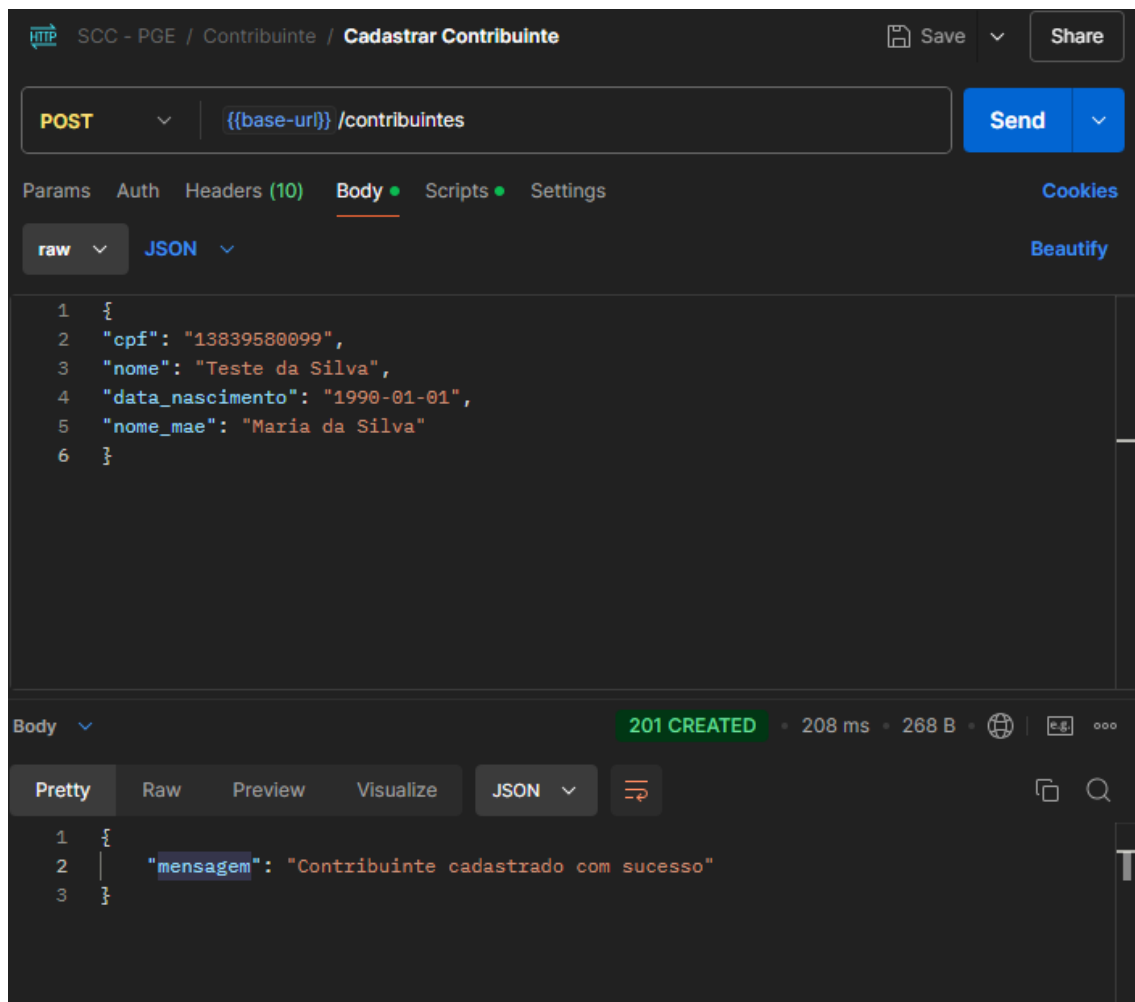
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "13839580099", "nome": "Teste da Silva", "data_nascimento": "1990-01-01",  
"nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 201;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"mensagem": "Contribuinte cadastrado com sucesso"}
```



CT_09 Contribuinte já cadastrado: **SUCESSO**

- **Descrição:** Realizar um novo cadastro com um CPF já cadastrado no sistema.
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

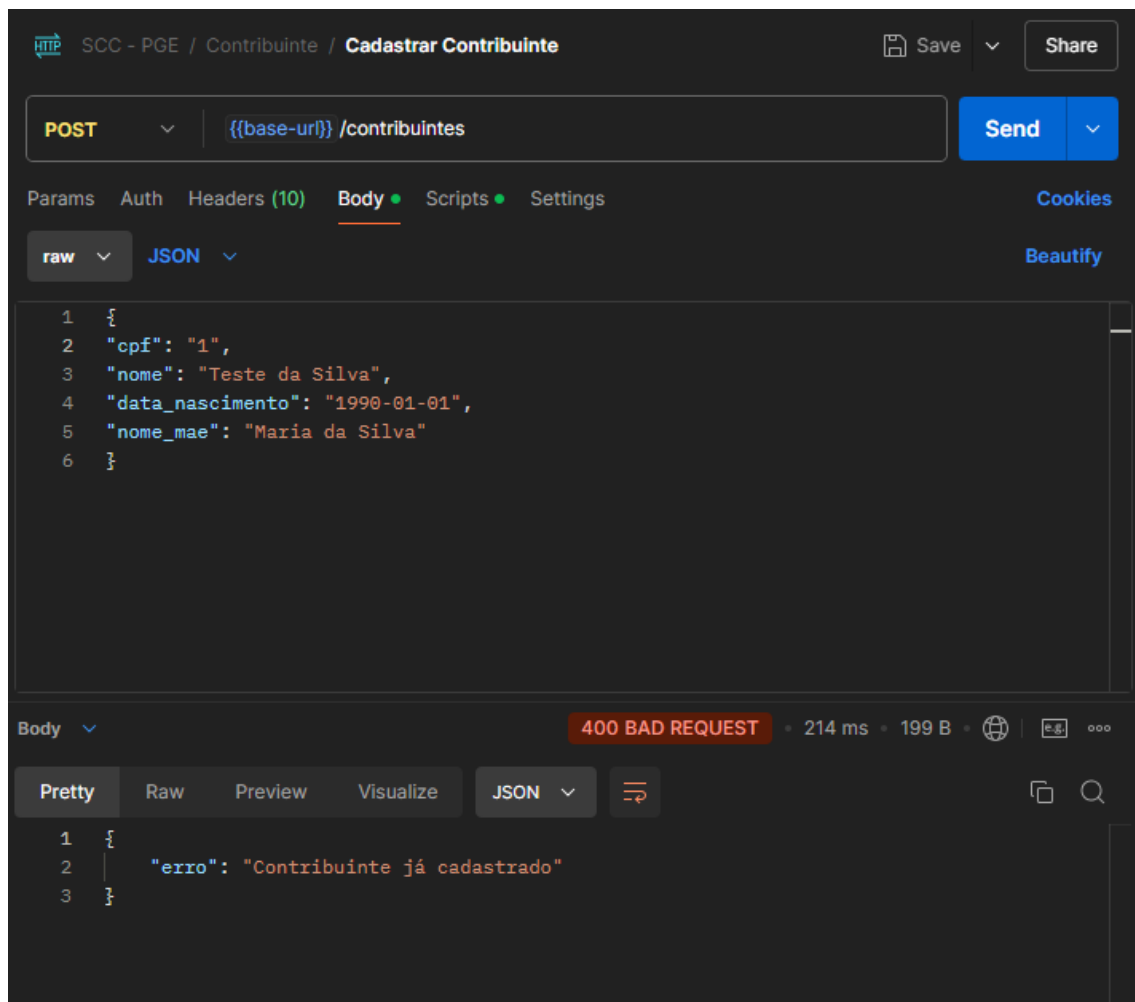
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "1", "nome": "Teste da Silva", "data_nascimento": "1990-01-01", "nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"mensagem": "Contribuinte já cadastrado"}
```



CT_10 Validar padronização campo Data de Nascimento: **FALHA**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório "data_nascimento" fora de padronizações:
 - Ser um campo do tipo Data;
 - Não permitir um valor que não seja uma data;
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

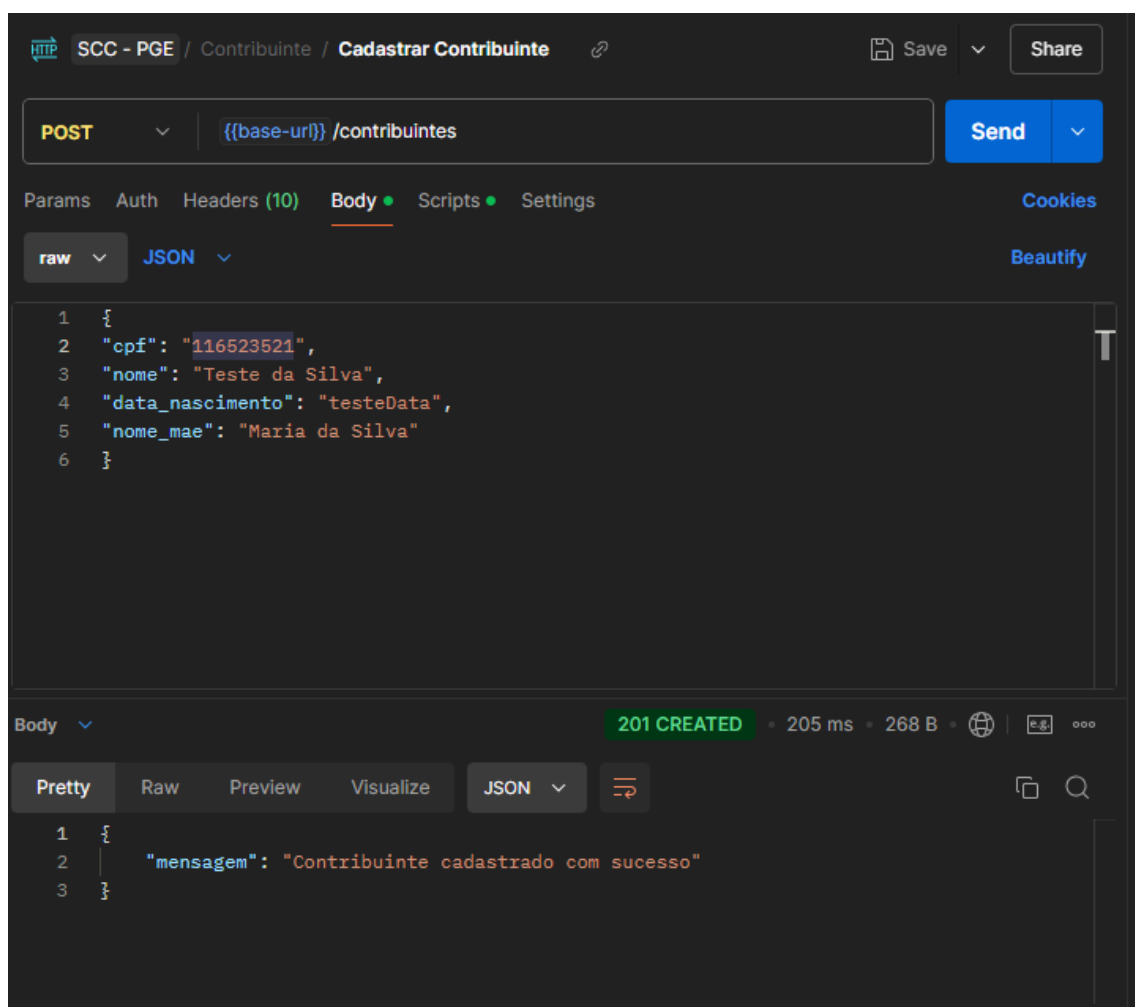
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{"cpf": "116523521", "nome": "Teste da Silva", "data_nascimento": "testeData",  
"nome_mae": "Maria da Silva"}
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Data de Nascimento Inválida"}
```



CT_11 Validar data de nascimento maior que data atual: **FALHA**

- **Descrição:** Realizar o cadastro de um contribuinte com o campo obrigatório "data_nascimento" maior que a Data Atual
- **Pré-Condições:** O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições.

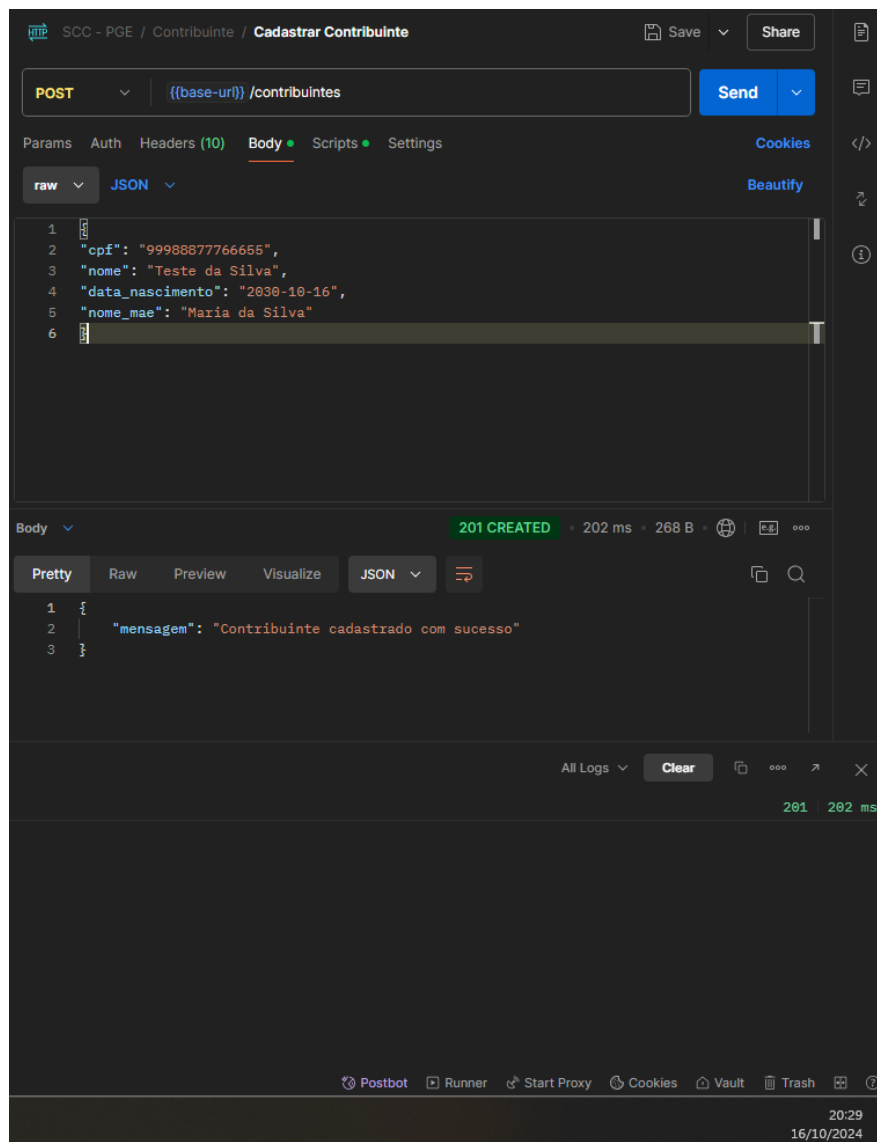
Dado que seja realizado uma requisição do tipo POST no *endpoint* /contribuintes com o corpo da requisição (body) abaixo em formato JSON:

```
{ "cpf": "999888777666-55", "nome": "Teste da Silva",  
  "data_nascimento_maior_que_data_atual": "testeData", "nome_mae": "Maria da Silva" }
```

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{ "erro": "Data de Nascimento informada não pode ser maior que a data atual" }
```



CN_03 Consulta Cadastro de Contribuinte

API para Cenário: <http://testeqa.pge.ce.gov.br>

CT_01 Requisição sem Token Informado: **SUCESSO**

- **Descrição:** Realizar a consulta de inscrições protestadas de um contribuinte sem enviar um token de autenticação.

Dado que seja realizado uma requisição do tipo POST no *endpoint* /inscricoes/{{cpf_a_ser_buscado}}

E não seja informado o token de autenticação;

Então o sistema deve retornar um status do tipo 401

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Token de acesso é necessário"}
```

The screenshot shows a REST client interface with the following details:

- URL:** `{{base-url}} /inscricoes/ {{cpf}}`
- Method:** GET
- Headers:** One header is visible: `x-access-token` with value `1`.
- Status:** 401 UNAUTHORIZED
- Response Body (JSON):**

```
{  "erro": "Token de acesso é necessário"}
```

CT_02 Requisição com Token Inválido: **SUCESSO**

- **Descrição:** Realizar a consulta de inscrições protestadas de um contribuinte enviando um token de autenticação inválido

Dado que seja realizado uma requisição do tipo POST no *endpoint* `/inscricoes/{{cpf_a_ser_buscado}}`

E seja informado o token de autenticação inválido;

Então o sistema deve retornar um status do tipo 401;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Token inválido"}
```

The screenshot shows a REST client interface with the following details:

- URL:** `{{base-url}} /inscricoes/ {{cpf}}`
- Method:** GET
- Headers:**

Key	Value	Description
<input checked="" type="checkbox"/> x-access-token	1	
- Status:** 401 UNAUTHORIZED
- Response Body (JSON):**

```
1 {
2   "erro": "Token inválido"
3 }
```


CT_03 Consultar Inscrições: **SUCESSO**

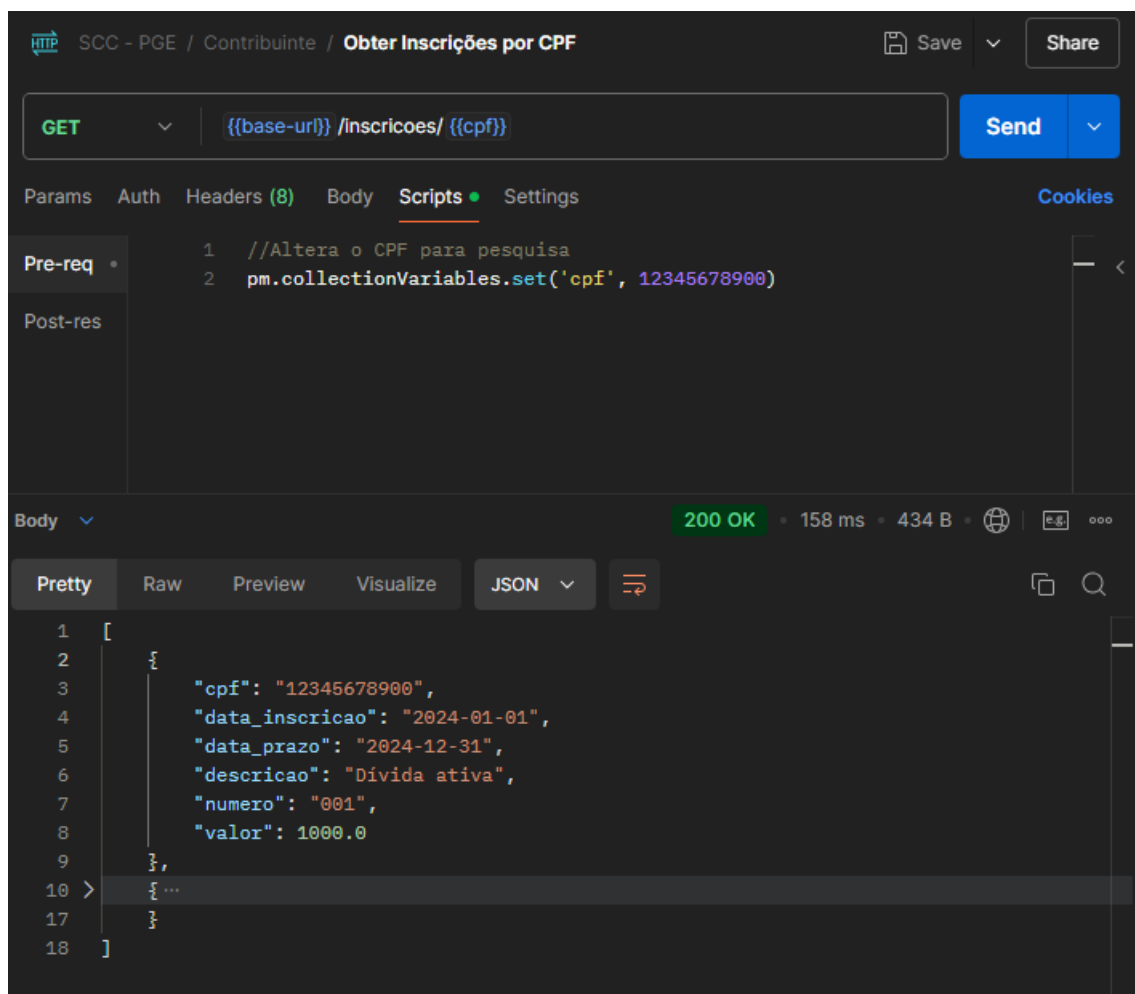
- **Descrição:** Realizar a consulta de inscrições protestadas de um contribuinte
- **Pré-Condições:**
 - O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições;
 - Deve existir um contribuinte com inscrições já cadastradas.

Dado que seja realizado uma requisição do tipo POST no *endpoint* `/inscricoes/{{cpf_a_ser_buscado}}`

Então o sistema deve retornar um status do tipo 200;

E deve retornar uma resposta formatada no padrão JSON com o modelo abaixo com um ou mais registros:

```
{"cpf": "12345678900", "data_inscricao": "2024-01-01", "data_prazo": "2024-12-31",  
"descricao": "Dívida ativa", "numero": "001", "valor": 1000.0}
```



CT_04 Consultar contribuinte sem Inscrições: **SUCESSO**

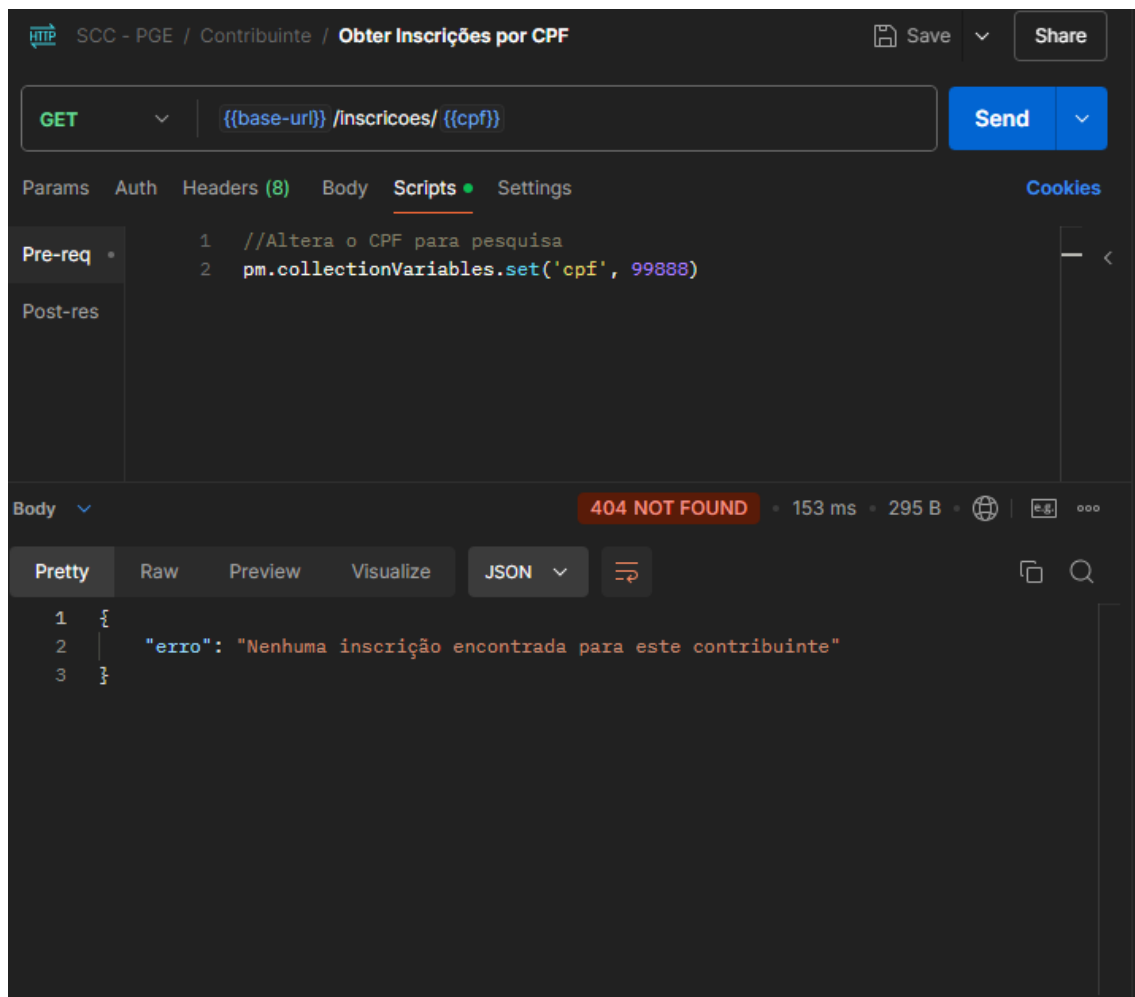
- **Descrição:** Realizar a consulta de inscrições protestadas de um contribuinte que não possui inscrições
- **Pré-Condições:**
 - O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições;
 - Deve existir um contribuinte sem inscrições cadastradas.

Dado que seja realizado uma requisição do tipo POST no *endpoint* `/inscricoes/{{cpf_a_ser_buscado}}`

Então o sistema deve retornar um status do tipo 404;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "Nenhuma inscrição encontrada para este contribuinte"}
```



CT_05 Validar CPF consultado: **FALHA**

- **Descrição:** Realizar a validação do CPF informado para a consulta de inscrições protestadas de um contribuinte
- **Pré-Condições:**
 - O usuário já deve estar logado no sistema e incluir o token para autenticação das requisições;

Dado que seja realizado uma requisição do tipo POST no *endpoint* `/inscricoes/{{cpf_a_ser_buscado}}`

E seja informado um valor no campo CPF que não seja um CPF;

Então o sistema deve retornar um status do tipo 400;

E deve retornar uma resposta formatada no padrão JSON com a mensagem abaixo:

```
{"erro": "CPF Inválido "}
```

The screenshot shows a REST client interface with the following details:

- URL:** `GET {{base-url}} /inscricoes/ {{cpf}}`
- Method:** GET
- Scripts:**

```
1 //Altera o CPF para pesquisa
2 pm.collectionVariables.set('cpf', "teste")
```
- Status:** 404 NOT FOUND (215 ms, 295 B)
- Response Body (JSON):**

```
1 {
2   "erro": "Nenhuma inscrição encontrada para este contribuinte"
3 }
```