

# WebService VIA CEP

## Como Executar o Código

**Certifique-se de ter o Robot Framework instalado.** Caso não tenha, pode instalar usando o seguinte comando:

```
bash
```

```
pip install robotframework
```

**Instalar as Bibliotecas Requeridas:** O código utiliza as bibliotecas RequestsLibrary e JSONLibrary para fazer requisições HTTP e manipular dados JSON. Instale-as com:

```
pip install robotframework-requests
```

```
pip install robotframework-jsonlibrary
```

**Estrutura do Código:** O código está estruturado em duas seções principais:

- **Settings:** Aqui são carregadas as bibliotecas e definidas variáveis como a URL base, CEPs válidos e inválidos, e outros parâmetros de teste.
- **Keywords:** Aqui são definidas funções reutilizáveis para validar o status da resposta, conteúdo do JSON, entre outros.

**Executar o Teste:** Para rodar o teste, execute o comando abaixo no terminal:

```
robot viaCepTeste.robot
```

2.

## Resultados Esperados

1. **Cenário 01 - Verificar se todos os campos retornados pelo CEP válido estão corretos**

**Testando o CEP Válido (53230200):**

- Status Code: 200
- O JSON retornado deve conter as seguintes informações:
  - "cep": 53230-200
  - "logradouro": Rua Estudante Alfredo Cantalice
  - "bairro": Vila Popular
  - "localidade": Olinda
  - "uf": PE
  - "estado": Pernambuco
  - "regiao": Nordeste

- "ibge": 2609600
- "ddd": 81
- "siafi": 2491
- "complemento", "unidade", "gia": Valores vazios.

**2. Cenário 02 - Verificar se todos os campos retornados pelo CEP que possui traço estão corretos**

**Testando um CEP Válido com Traço (53230-200):**

- Status Code: 200
- O JSON retornado deve conter as seguintes informações:
  - "cep": 53230-200
  - "logradouro": Rua Estudante Alfredo Cantalice
  - "bairro": Vila Popular
  - "localidade": Olinda
  - "uf": PE
  - "estado": Pernambuco
  - "regiao": Nordeste
  - "ibge": 2609600
  - "ddd": 81
  - "siafi": 2491
  - "complemento", "unidade", "gia": Valores vazios.

**3. Cenário 03 - Verificar CEP inválido com 8 dígitos**

**Testando um CEP Inválido (11111111):**

- Status Code: 200
- O JSON retornado deve ter o campo "erro" com o valor true

**4. Cenário 04 - Verificar CEP invalido com menos de 8 dígitos**

**Testando CEP Menor que 8 Dígitos:**

- Status Code: 400
- O teste espera o erro retornado para um CEP com menos de 8 dígitos.

**5. Cenário 05 - Verificar CEP invalido com mais de 8 dígitos**

**Testando CEP Maior que 8 Dígitos:**

- Status Code: 400
- O teste espera o erro retornado para um CEP com mais de 8 dígitos.
-

## 6. Cenário 06 - Verificar se o CEP retornado está diferente do esperado

### Testando um CEP Diferente do Esperado (53230115):

- Status Code: 200
- O valor do "cep" no JSON retornado não deve ser igual ao valor esperado (53230200).

## 7. Cenário 07 - Verificar CEP não informado

### Testando quando o CEP não é informado:

- Status Code: 404
- Quando o campo de CEP não for informado, espera-se que o status seja 404.

## Funções Reutilizáveis

- **Criar Sessão:** A função Criar Sessão é criar uma sessão HTTP para fazer as requisições.
- **Fazer Requisição GET:** A função Fazer Requisição GET envia uma requisição GET para um endpoint específico.
- **Verificar Status Code:** A função Verificar Status Code válida se o status code da resposta corresponde ao esperado.

## Considerações

- Se algum teste falhar, o Robot Framework exibirá uma mensagem de erro detalhada indicando o que não corresponde ao esperado.
- Os testes cobrem uma variedade de cenários de CEP, incluindo válidos, inválidos e formatação com traços.

Esse conjunto de testes tem como objetivo validar a resposta da API de CEP, tratando de diferentes formatos de entrada e os respectivos resultados esperados.