



Dica do dia

Identificação e Tratamento de Erro de forma simples



Entrada

- ✓ Código abaixo está consumindo uma API de conversão de Moeda onde na função “cotacaomoeda” recebe o valor a ser transformado e o par de moedas a ser feita a conversão. A saída é a conversão do par de moedas com o tratamento.

```
# imports
import requests
import backoff
import json
Run Cell | Run Above | Debug Cell | Go to [2]
# %%
url = 'https://economia.awesomeapi.com.br/last/USD-BRL'
ret = requests.get(url)
```

```
# Usando decorador
@backoff.on_exception(backoff.expo,(),max_tries=1)
def error_check(func):
    def inner_func(*args, **kwargs):
        try:
            func(*args, **kwargs)
        except Exception as e:
            print(f"Falhou na Função : {func.__name__} , no Argumento: {e} ")
    return inner_func
```

Run Cell | Run Above | Debug Cell

```
# %%
@error_check
def cotacaomoeda(valor, moeda):
    url = f'https://economia.awesomeapi.com.br/last/{moeda}'
    ret = requests.get(url)
    dolar = json.loads(ret.text)[moeda.replace('-', '')]
    print(f" {valor} {moeda[:3]}, hoje custam {float(dolar['bid']) * valor} {moeda[-3:]}. ")
```

Run Cell | Run Above | Debug Cell

```
# %%
```

```
cotacaomoeda(20, 'USD-BRL')
cotacaomoeda(10, 'USD-BRL')
cotacaomoeda(15, 'TPP-TPP')
cotacaomoeda(20, 'MOEDATESTES')
cotacaomoeda(10, 'BRL-USD')
```

Saída de qualquer função com o tratamento

✓ @error_check ...

```
20 USD, hoje custam 104.834 BRL.
10 USD, hoje custam 52.417 BRL.
Falhou na Função : cotacaomoeda , no Argumento: 'TPPTPP'
Falhou na Função : cotacaomoeda , no Argumento: 'MOEDATESTES'
10 BRL, hoje custam 1.907 USD.
```

- ✓ Como podemos ver foi chamado a função 5 vezes onde foi passado o argumento que não existe na API de consulta os parâmetros com erros “TPPTPP” e “MOEDATESTES” no caso a função “error_check” + a lib backoff foram projetadas para executar a identificação do erro retornando o nome da função com falha e quais parâmetros estão incorretos ou não identificados.
- ✓ Não e preciso ficar tratando função por função e só usar o decorador em cima de qualquer função implementada, no caso foi implementada com o nome error_check então usar @error_check.

Ganhos



- Reuso de código
- Velocidade de implementação
- Facilidade de identificar o erro
- Escalabilidade
- Componentização
-

Entrada

- ✓ Código abaixo está consumindo uma API de conversão de Moeda onde na função "cotacaomoeda" recebe o valor a ser transformado e o par de moedas a ser feita a conversão. A saída é a conversão do par de moedas SEM o tratamento.

```
# imports
import requests
import backoff
import json
Run Cell | Run Above | Debug Cell | Go to [2]
# %%
url = 'https://economia.awesomeapi.com.br/last/USD-BRL'
ret = requests.get(url)
```

```
def cotacaomoeda(valor, moeda):
    url = f'https://economia.awesomeapi.com.br/last/{moeda}'
    ret = requests.get(url)
    dolar = json.loads(ret.text)[moeda.replace('-', '')]
    print(f" {valor} {moeda[:3]}, hoje custam {float(dolar['bid']) * valor} {moeda[-3:]}. ")
Run Cell | Run Above | Debug Cell | Go to [90]
# %%
cotacaomoeda(20, 'USD-BRL')
cotacaomoeda(10, 'USD-BRL')
cotacaomoeda(15, 'TPP-TPP')
cotacaomoeda(20, 'MOEDATESTETE')
cotacaomoeda(10, 'BRL-USD')
```

Saída Sem tratamento apenas da função

```
cotacaomoeda(20, 'USD-BRL') ...
20 USD, hoje custam 104.834 BRL.
10 USD, hoje custam 52.417 BRL.

-----
KeyError                                Traceback (most recent call last)
c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in <cell line: 4>()
     72 cotacaomoeda(20, 'USD-BRL')
     73 cotacaomoeda(10, 'USD-BRL')
----> 74 cotacaomoeda(15, 'TPP-TPP')
     75 cotacaomoeda(20, 'MOEDATESTETE')
     76 cotacaomoeda(10, 'BRL-USD')

c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in cotacaomoeda(valor, moeda)
     67 url = f'https://economia.awesomeapi.com.br/last/{moeda}'
     68 ret = requests.get(url)
----> 69 dolar = json.loads(ret.text)[moeda.replace('-', '')]
     70 print(f" {valor} {moeda[:3]}, hoje custam {float(dolar['bid']) * valor} {moeda[-3:]}. ")

KeyError: 'TPPTPP'

20 USD, hoje custam 104.834 BRL.
10 USD, hoje custam 52.417 BRL.

-----
Output exceeds the size limit. Open the full output data in a text editor
KeyError                                Traceback (most recent call last)
c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in error_check.<locals>.inner_fun
c(*args, **kwargs)
----> 60 try:
     61     func(*args, **kwargs)
     62 except:

c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in cotacaomoeda(valor, moeda)
     67 url = f'https://economia.awesomeapi.com.br/last/{moeda}'
     68 ret = requests.get(url)
----> 69 dolar = json.loads(ret.text)[moeda.replace('-', '')]
     70 print(f" {valor} {moeda[:3]}, hoje custam {float(dolar['bid']) * valor} {moeda[-3:]}. ")

KeyError: 'TPPTPP'

During handling of the above exception, another exception occurred:

TypeError                                Traceback (most recent call last)
c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in <cell line: 11>()
     72 cotacaomoeda(20, 'USD-BRL')
     73 cotacaomoeda(10, 'USD-BRL')
----> 74 cotacaomoeda(15, 'TPP-TPP')
     75 cotacaomoeda(20, 'MOEDATESTETE')
     76 cotacaomoeda(10, 'BRL-USD')

c:\Users\welli\OneDrive\Desktop\ESTUDOS\Fundamentos de Ingestão de Dados\001\main.py in error_check.<locals>.inner_fun
c(*args, **kwargs)
...
     61     func(*args, **kwargs)
     62 except:
----> 63     print(f"Falhou na função : {func.__name__} , {func(Exception)} ")

TypeError: cotacaomoeda() missing 1 required positional argument: 'moeda'
```

Perdas



- Reuso de código
- Velocidade de implementação
- Facilidade de identificar o erro
- Escalabilidade
- Componentização
-