O método json\_normalize() é usado para normalizar dados JSON em um formato tabular (como um DataFrame do Pandas) para facilitar a análise e manipulação. Os parâmetros mais importantes do método json\_normalize() são:

* data: o objeto JSON a ser normalizado.
* record\_path: um caminho para acessar o array de registros dentro do objeto JSON.
* meta: uma lista de colunas adicionais a serem incluídas no DataFrame, além das colunas normalizadas.
* errors: como lidar com erros de normalização. Os valores possíveis são "raise" (lançar um erro), "ignore" (ignorar o erro).
* sep: separador de colunas usado para concatenar as chaves do objeto JSON aninhado. O padrão é ".".

Aqui está um exemplo de como usar o método json\_normalize() com alguns dos parâmetros mencionados acima:

import pandas as pd

data = {

"empresa": "alura",

"funcionarios": [

{"nome": "Alice", "endereço": {"cidade": "São Paulo", "estado": "SP"}},

{"nome": "Bob", "endereço": {"cidade": "Rio de Janeiro", "estado": "RJ"}}

]

}

df = pd.json\_normalize(data, record\_path='funcionarios', meta = "empresa", errors='ignore')

**Saída:**

|  | **nome** | **endereço.cidade** | **endereço.estado** | **empresa** |
| --- | --- | --- | --- | --- |
| 0 | Alice | São Paulo | SP | alura |
| 1 | Bob | Rio de Janeiro | RJ | alura |

Este código em Python faz uso das bibliotecas pandas para criar um DataFrame que contém informações sobre pessoas associadas a uma empresa.

O objeto data contém uma chave "empresa" com o valor "alura", bem como uma chave "funcionarios" cujo valor é uma lista contendo dois dicionários. Cada dicionário na lista representa uma pessoa, com informações sobre o nome e o endereço. O endereço é um dicionário aninhado com as chaves "cidade" e "estado".

A função pd.json\_normalize() é usada para normalizar os dados em uma tabela. Nesse caso, data é passado como o objeto JSON a ser normalizado. O parâmetro record\_path é definido como 'funcionarios' para indicar que os dados das pessoas devem ser normalizados. O parâmetro meta é definido como "empresa" para incluir a informação da empresa em cada linha da tabela. Finalmente, o parâmetro errors é definido como 'ignore' para ignorar quaisquer erros que possam ocorrer durante a normalização.

A saída do código é um objeto DataFrame que contém informações sobre as pessoas associadas à empresa, com cada pessoa representada em uma linha e as informações de nome, cidade e estado em colunas separadas.