

# Zomato

teste 123

```
In [ ]: from bs4 import BeautifulSoup
import requests
import pandas as pd
```

texto

```
In [ ]: headers = {
    # "Access-Control-Allow-Origin": "*",
    # "Access-Control-Allow-Methods": "GET",
    # "Access-Control-Allow-Headers": "Content-Type",
    # "accept": "*/*",
    "User-Agent": "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko)
}

# url = "https://www.zomato.com/beja/beja-restaurants"
# response = requests.get(url, headers=headers)
# response.status_code
# soup = BeautifulSoup(response.content, 'lxml')
soup = BeautifulSoup(open(r"C:\Users\vitor\Desktop\pi2021\projecto\webscrape\scrapes\zomato\restaurantes\zomato-1
```

```
In [ ]: restaurant = []
for name in soup.findAll('h4',{'class':'sc-1hp8d8a-0'}):
    restaurant.append(name.text.strip())
print(len(restaurant))
```

```
In [ ]: type = []
for name in soup.findAll('p',{'class':'jaK0Qh'}):
    print(name)
    type.append(name.text.strip())
for name in soup.findAll('p',{'class':'kegdaG'}):
    print(name)
    type.append(name.text.strip())
print(len(type))
```

```
In [ ]: price = []
for p in soup.findAll('p',{'class':'ftdqla'}):
    price.append(p.text.replace('€ para dois','').strip())
for p in soup.findAll('p',{'class':'k00Nhy'}):
    price.append(p.text.replace('€ para dois','').strip())
print(len(price))
```

```
In [ ]: #bad code

length = len(price)
if length > len(type):
    length = len(type)
if length > len(restaurant):
    length = len(restaurant)
```

```
In [ ]: d1 = {'Restaurante': restaurant[:length], 'Tipo': type[:length], 'Preço': price[:length]}
df = pd.DataFrame.from_dict(d1)
print(df)
df.to_csv('listtable.csv')
```

```
In [ ]: reviews_links = []
for link in soup.findAll('a', {'class': 'ieKty'}):

    a = link['href']
    reviews_links.append(a.replace('/info', '/reviews'))
for link in soup.findAll('a', {'class': 'jjSACU'}):

    a = link['href']
    reviews_links.append(a.replace('/info', '/reviews'))
# print(reviews_links)
```

```
In [ ]: count = 0
allreviews = []
```

```

for link in reviews_links:
    try:
        response2 = requests.get(link, headers=headers)
        soup2 = BeautifulSoup(response2.content, 'lxml')
        for r in soup2.findAll('p'): # sempre a mudar a class, vai sofrer ETL
            try:
                rev = r.text
                # print(rev)
                allreviews.append(rev + '\n')
            except:
                pass
    except:
        pass
count += 1
if allreviews != []:
    seen = set()
    allreviews = [item for item in allreviews if not(
        tuple(item) in seen or seen.add(tuple(item)))]
    dfr = pd.DataFrame.from_dict({'Avaliações': allreviews})
    print(dfr)
    dfr.to_csv('restaurante' + str(count) + '.csv')
    allreviews = []

```