Kando_tech

December 24, 2024

[1]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

[2]:
```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; Win 64 ; x64) Apple␣
↪WeKit /537.36(KHTML , like Gecko) Chrome/80.0.3987.162 Safari/537.36'}
```

[3]:
```
webpage = requests.get("https://kando.tech/investors/all?
↪page=1",headers=headers).text
```

[7]:
```
soup=BeautifulSoup(webpage,'lxml')
```

[ ]:
```
for i in soup.find("div",class_="view-content"):
  print(i.text.strip())
```

[51]:
```
for i in soup.find_all("td",class_="views-field views-field-field-investor"):
  print(i.text.strip())
```

```
TPG Growth LLC
TPG Capital LP
US Venture Partners. (USVP)
3i Group PLC
Benchmark Capital
Canaan Partners
Band of Angels Management LLC
GTCR Capital Partners
Matrix Management Corp (Matrix Partners)
First Round Capital
Battery Ventures LP
Rho Capital Partners (Rho Ventures)
TA Associates Inc
Hercules Technology Growth Capital Inc
Morgenthaler Management Partners (Morgenthaler Ventures),  (Morgenthaler
Partners)
Austin Ventures
California State Teachers' Retirement System (CalSTRS)
Redpoint Ventures
General Catalyst Partners
```

Charles River Ventures (CRV)

[53]:
```
for i in soup.find_all("td",class_="views-field views-field-field-country"):
  print(i.text.strip())
```

```
United States
United States
United States
United Kingdom
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
United States
```

[55]:
```
for i in soup.find_all("td",class_="views-field views-field-field-co-summary"):
  print(i.text.strip())
```

```
The middle market and growth equity investment platform of TPG.
Private equity firm
Silicon Valley venture capital firm.
Major, long established, quoted private equity investor
Early stage venture capital firm focused on social, mobile, local and cloud
companies.
Venture capital firm with a focus on early stage technology and healthcare
companies
Formal group of 165+ angel investors, mostly former and current high tech
executives, who invest into technology startups.
Mezzanine debt fund managed by GTCR.
Venture capital firm.
Venture capital firm that specialises in providing seed-stage funding to
technology companies.
Venture capital firm
Venture capital firm that actively invests in companies related to technology,
new media, cleaning technology and healthcare.
Global growth private equity firm.
Specialised finance company providing debt and equity growth capital to
technology-related companies at all stages of development.
```

```
One of the oldest private equity investment firms in the US investing through
both venture capital and leverage buyout transactions.
Private equity firm focused on VC and growth investments in business services,
supply chain, financial services, new media, Internet, info services.
Provider of retirement, disability and survivor benefits for full-time and part-
time California public school educators.
Venture capital capital firm that invests in startups across the seed, early and
growth phases.
Venture capital firm that makes early-stage and transformational investments.
Independent venture capital firm
```

[57]:
```
for i in soup.find_all("td",class_="views-field views-field-nid␣
 ↪views-align-right"):
  print(i.text.strip())
```

```
268
257
253
247
246
242
241
234
223
221
219
216
213
209
208
203
198
195
192
187
```

[ ]:
```
```

[60]:
```
company0 = soup.find_all("div",class_="view-content")
```

[62]:
```
len(company0)
```

[62]: 1

[68]:
```
Investors = []
Country = []
Profile = []
Deal = []
```

```
if company0:
    for i in company0:
        if i is not None:
            Investors.append(i.find('td', class_='views-field␣
 ↪views-field-field-investor') and i.find('td', class_='views-field␣
 ↪views-field-field-investor').text.strip() or '')
            Country.append(i.find("td",class_="views-field␣
 ↪views-field-field-country") and i.find("td",class_="views-field␣
 ↪views-field-field-country").text.strip() or '')
            Profile.append(i.find("td",class_="views-field␣
 ↪views-field-field-co-summary") and i.find("td",class_="views-field␣
 ↪views-field-field-co-summary").text.strip() or '')
            Deal.append(i.find("td",class_="views-field views-field-nid␣
 ↪views-align-right") and i.find("td",class_="views-field views-field-nid␣
 ↪views-align-right").text.strip() or '')
else:
    print("company0 is empty")

d = {"Investors": Investors, "Country": Country, "Profile": Profile, "Deal":␣
 ↪Deal}

df = pd.DataFrame(d)
print(df)
```

```
        Investors        Country  \
0  TPG Growth LLC  United States

                                                Profile Deal
0  The middle market and growth equity investment…  268
```

[ ]:
```
for i in soup.find("div",class_="view-content"):
  print(i.text.strip())
```

[ ]:
```
```

[72]:
```
for i in company0:
  i.find_all('td', class_='views-field views-field-field-investor')
```

[74]:
```
Investors = []
Country = []
Profile = []
Deal = []

for i in company0:
    # Find the element and then extract the text
```

```
    investor_element = i.find("td", class_="views-field␣
 ↪views-field-field-investor")
    Investors.append(investor_element.text.strip() if investor_element else␣
 ↪None)

    country_element = i.find("td", class_="views-field␣
 ↪views-field-field-country")
    Country.append(country_element.text.strip() if country_element else None)

    profile_element = i.find("td", class_="views-field␣
 ↪views-field-field-co-summary")
    Profile.append(profile_element.text.strip() if profile_element else None)

    deal_element = i.find("td", class_="views-field views-field-nid␣
 ↪views-align-right")
    Deal.append(deal_element.text.strip() if deal_element else None)

d = {"Investors": Investors, "Country": Country, "Profile": Profile, "Deal":␣
 ↪Deal}

df = pd.DataFrame(d)
d
```

[74]:
```
{'Investors': ['TPG Growth LLC'],
 'Country': ['United States'],
 'Profile': ['The middle market and growth equity investment platform of TPG.'],
 'Deal': ['268']}
```

[ ]:
```
```

[77]:
```
company = soup.find_all("div",class_="dialog-off-canvas-main-canvas")
```

[79]:
```
len(company)
```

[79]:
```
1
```

[81]:
```
company1 = soup.find_all("div",class_="layout-container")
```

[83]:
```
len(company1)
```

[83]:
```
2
```

[85]:
```
company2 = soup.find_all("div",class_="region region-content")
```

[87]:
```
len(company2)
```

[87]:
```
1
```

[89]:
```
company5 = soup.find_all("div",class_="investor-class")
```

[ ]:
```
```

[94]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; Win 64 ; x64)␣
 ↪AppleWebKit /537.36(KHTML, like Gecko) Chrome/80.0.3987.162 Safari/537.36'}

# Function to scrape a single page
def scrape_page(page_number):
    url = f"https://kando.tech/investors/all?page={page_number}"
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.text, 'lxml')

    investors = []
    countries = []
    profiles = []
    deals = []

    # Extract data from this page
    for i in soup.find_all("td", class_="views-field␣
 ↪views-field-field-investor"):
        investors.append(i.text.strip())

    for i in soup.find_all("td", class_="views-field␣
 ↪views-field-field-country"):
        countries.append(i.text.strip())

    for i in soup.find_all("td", class_="views-field␣
 ↪views-field-field-co-summary"):
        profiles.append(i.text.strip())

    for i in soup.find_all("td", class_="views-field views-field-nid␣
 ↪views-align-right"):
        deals.append(i.text.strip())

    return investors, countries, profiles, deals

# Scrape multiple pages
all_investors = []
all_countries = []
all_profiles = []
all_deals = []
```

```
for page_number in range(0, 6):  # Adjust this number based on how many pages you want to scrape
    try:
        investors, countries, profiles, deals = scrape_page(page_number)
        all_investors.extend(investors)
        all_countries.extend(countries)
        all_profiles.extend(profiles)
        all_deals.extend(deals)

        print(f"Scraped page {page_number}")
    except Exception as e:
        print(f"Error scraping page {page_number}: {e}")

# Create DataFrame
data = {
    "Investors": all_investors,
    "Country": all_countries,
    "Profile": all_profiles,
    "Deal": all_deals
}

df = pd.DataFrame(data)

# Save the DataFrame to a CSV file
df.to_csv('kando_investors_data3.csv', index=False)
print("Data saved to kando_investors_data.csv")
```

```
Scraped page 0
Scraped page 1
Scraped page 2
Scraped page 3
Scraped page 4
Scraped page 5
Data saved to kando_investors_data.csv
```

[96]:
```
df.head()
```

[96]:

	Investors	Country	Profile	Deal
0	New Enterprise Associates Inc (NEA)	United States	Global venture capital firm focused on helping…	706
1	Sequoia Capital Operations LLC (Sequoia)	United States	Venture capital firm.	636
2	Norwest Venture Partners (NVP), (Norwest Vent…	United States		
3	Western Technology Investment (WTI)	United States		
4	Kleiner Perkins Caufield & Byers LLC (KPCB)	United States		

```
2                              Venture capital firm.  618
3  Venture debt firm investing debt and equity ca…  606
4                              Venture capital firm.  530
```

[102]:
```
df.shape      #In one page there are total 20 data so for the six page it is␣
↪120(20x6)
```

[102]:
```
(120, 4)
```

[106]:
```
df.count()
```

[106]:
```
Investors    120
Country      120
Profile      120
Deal         120
dtype: int64
```

[ ]:
```

```