job_crawling

November 29, 2024

1 Import Section

[1]:
```
from bs4 import BeautifulSoup
import pandas as pd
import duckdb
import requests
from dataclasses import dataclass
from datetime import datetime
```

2 Job Extraction

[2]:
```
search_term = r"data%20engineer"
```

[3]:
```
@dataclass
class Job:
    published: datetime
    title: str
    location: str
    workload: str
    employment_type: str
    company: str

def get_jobs(soup):
    job_elements = soup.find("div", class_="d_grid gap_s16 ov_hidden p_s8␣
 ↪pb_s16").find_all(
        "div", class_="d_flex bg-c_white bdr_r16 flex-d_column h_100% p_s16␣
 ↪pos_relative"
    )
    for job_element in job_elements:
        found_job = list(job_element.stripped_strings)

        del found_job[1]
        found_job = [item for item in found_job if "Quick apply" not in item␣
 ↪and "Recommended" not in item]
        if len(found_job) == 6:
            yield Job(
                published=datetime.strptime(found_job[0].replace("Published: ",␣
↪""), "%d %B %Y").date(),
                title=found_job[1],
                location=found_job[2],
                workload=found_job[3],
                employment_type=found_job[4],
                company=found_job[5],
            )
```

[4]:
```
url = f"https://www.jobs.ch/en/vacancies/?term={search_term}"
page = requests.get(url)
soup = BeautifulSoup(page.content, "html.parser")
pages = soup.find("div", class_="d_flex ai_center gap_s4").find_all("a")
last_page = int(list(pages[-1].stripped_strings)[-1])

jobs = []

for page_number in range(1, last_page):
    uri = f"{url}&page={page_number}"
    page = requests.get(uri)
    soup = BeautifulSoup(page.content, "html.parser")
    jobs.extend(job for job in get_jobs(soup))

print(len(jobs))
print(jobs[0])
```

```
1304
Job(published=datetime.date(2024, 11, 27), title='MES Engineer',
location='Solothurn', workload='100%', employment_type='Unlimited employment',
company='Emerson Automation Solutions')
```

3 Data Cleaning

[5]:
```
jobs_df = pd.DataFrame([j.__dict__ for j in jobs])
jobs_df.head()
```

[5]:
```
    published                             title   location   workload  \
0  2024-11-27                      MES Engineer  Solothurn       100%
1  2024-11-13   Process Development Engineer 100%     Luzern       100%
2  2024-11-02                  DevOps Engineer C2I  Dübendorf       100%
3  2024-11-12  Production Process Control Engineer  Schlieren  80 - 100%
4  2024-11-20          Senior FPGA Engineer (f/m/d)  Heerbrugg       100%

        employment_type                      company
0  Unlimited employment  Emerson Automation Solutions
1  Unlimited employment                  Schurter AG
2  Unlimited employment            RUAG MRO Holding AG
```

```
3  Unlimited employment                       EXALOS AG
4  Unlimited employment             Leica Geosystems AG
```

[6]:
```
jobs_df['published'] = pd.to_datetime(jobs_df['published'], errors='coerce')
jobs_df.fillna(value={'salary': 'Not specified', 'employment_type': 'Not␣
 ↪specified'}, inplace=True)
```

4 Insert Data into DB

[7]:
```
con = duckdb.connect(database='jobs.duckdb', read_only=False)
con.execute("DROP TABLE IF EXISTS jobs")
con.register('jobs_df_view', jobs_df)
con.execute("CREATE TABLE jobs AS SELECT * FROM jobs_df_view")
```

[7]:
```
<duckdb.duckdb.DuckDBPyConnection at 0x115fd6930>
```

5 Inspect data

[8]:
```
amount, = con.execute("select count(*) from jobs").fetchone()
amount
```

[8]:
```
1304
```

[9]:
```
example = con.execute("select * from jobs limit 5").fetchdf()
example
```

[9]:
```
   published                               title   location   workload  \
0 2024-11-27                        MES Engineer  Solothurn       100%
1 2024-11-13    Process Development Engineer 100%      Luzern       100%
2 2024-11-02                   DevOps Engineer C2I  Dübendorf       100%
3 2024-11-12  Production Process Control Engineer  Schlieren  80 - 100%
4 2024-11-20            Senior FPGA Engineer (f/m/d)  Heerbrugg       100%

        employment_type                        company
0  Unlimited employment  Emerson Automation Solutions
1  Unlimited employment                    Schurter AG
2  Unlimited employment            RUAG MRO Holding AG
3  Unlimited employment                      EXALOS AG
4  Unlimited employment            Leica Geosystems AG
```

6 Cleanup

[10]:
```
con.close()
```