python-jobs-scraper

This notebook mirrors the finalized scraper.py pipeline step by step in a clean, minimal form.

1 Setup & Imports

In [1]:
```
# Standard lib
import os
import sys

# Third-party
import pandas as pd
import matplotlib.pyplot as plt

# Your scraper module
from remoteok_scraper.scraper import (
    fetch_data,
    parse_data,
    filter_data,
    # (optional) save_data, plot_date_counts if you want to invoke here
)

# Logging output enabled by default
```

2 Fetch & Parse

In [2]:
```
# 1. Fetch raw JSON job listings
jobs_raw = fetch_data()
print(f"Fetched {len(jobs_raw)} postings")

# 2. Parse into DataFrame
df = parse_data(jobs_raw)
df.head()
```

```
2025-07-11 09:08:57 [INFO] Fetching data from https://remoteok.com/api
Fetched 99 postings
```

Out[2]:

	date	company	position	location	url
0	2025-07-09 11:37:47+00:00	Gymflow	Customer Onboarding & Support Specialist		https://remoteOK.com/remote-jobs/remote-custom...
1	2025-07-09 11:35:52+00:00	Prezly	Senior TS React Developer		https://remoteOK.com/remote-jobs/remote-senior...
2	2025-07-09 00:00:09+00:00	CloudWalk	AI Driven Engineer Ruby	SÃ£o Paulo	https://remoteOK.com/remote-jobs/remote-ai-dri...
3	2025-07-08 08:00:04+00:00	Immutable	Senior Security Engineer	Australia	https://remoteOK.com/remote-jobs/remote-senior...
4	2025-07-07 17:21:00+00:00	Loancrate	Software Engineer		https://remoteOK.com/remote-jobs/remote-softwa...

3 Filter by Keywords

In [3]:
```
# 3. Filter for broad keywords
keywords = ["Python", "Data", "Engineer", "SQL", "JavaScript", "AWS", "Docker"]
filtered = filter_data(df, keywords)
print(f"After filter: {len(filtered)} rows")
filtered.head()
```

```
2025-07-11 09:09:42 [INFO] Filtering data with keywords: ['Python', 'Data', 'Enginee
r', 'SQL', 'JavaScript', 'AWS', 'Docker'], matches found: 45
After filter: 45 rows
```

Out[3]:

	date	company	position	location	url
0	2025-07-09 00:00:09+00:00	CloudWalk	AI Driven Engineer Ruby	SÃ£o Paulo	https://remoteOK.com/remote-jobs/remote-ai-dri...
1	2025-07-08 08:00:04+00:00	Immutable	Senior Security Engineer	Australia	https://remoteOK.com/remote-jobs/remote-senior...
2	2025-07-07 17:21:00+00:00	Loancrate	Software Engineer		https://remoteOK.com/remote-jobs/remote-softwa...
3	2025-07-04 18:00:05+00:00	Best Egg	Lead Software Engineer II Backend	Remote / Flexible	https://remoteOK.com/remote-jobs/remote-lead-s...
4	2025-07-04 00:51:09+00:00	LaunchBrightly	Application Engineer		https://remoteOK.com/remote-jobs/remote-applic...

4 Deduplicate & Master Table

In [4]:
```
# 4. Drop duplicate URLs (keep newest)
filtered = (
    filtered
    .drop_duplicates(subset="url", keep="first")
    .sort_values("date", ascending=False)
    .reset_index(drop=True)
)
filtered.shape
```

Out[4]: (45, 5)

5 Export Results

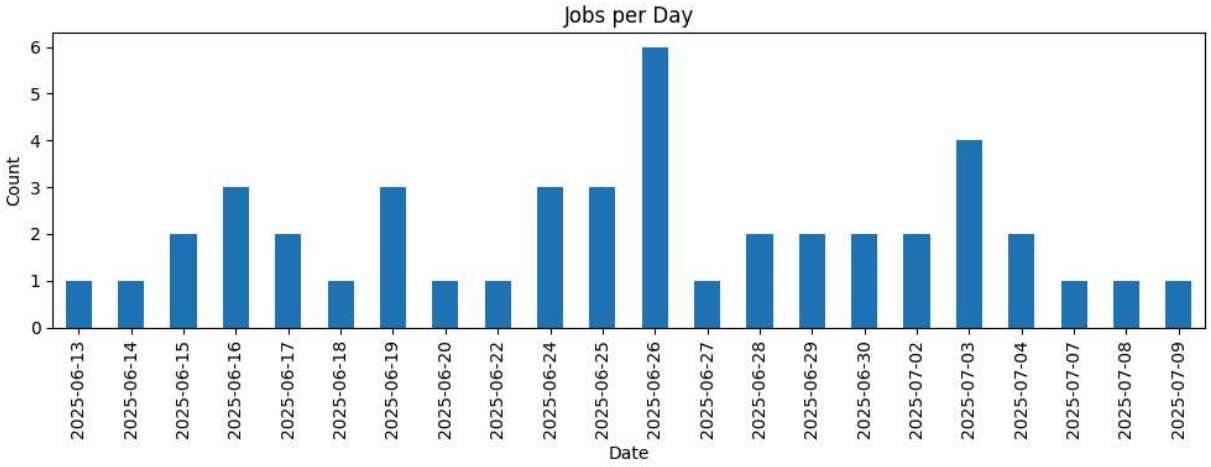
In [5]:
```
# 5a. Save CSV (or call save_data(filtered, ...))
filtered.to_csv("jobs_filtered.csv", index=False)
print("Saved jobs_filtered.csv")

# 5b. Optionally: JSON / Parquet
# filtered.to_json("jobs_filtered.json", orient="records", lines=True)
# filtered.to_parquet("jobs_filtered.parquet", index=False)
```

```
Saved jobs_filtered.csv
```

6 Plot: Time Series of Postings

In [6]:
```
# 6. Jobs per day
counts = filtered["date"].dt.date.value_counts().sort_index()
plt.figure(figsize=(10, 4))
counts.plot(kind="bar")
plt.xlabel("Date")
plt.ylabel("Count")
plt.title("Jobs per Day")
plt.tight_layout()
plt.show()
```



7 Plot: Positions vs. Openings

In [7]:
```
# 7. Job title vs number of openings
title_counts = filtered["position"].value_counts()
plt.figure(figsize=(12, 6))
plt.bar(title_counts.index, title_counts.values)
plt.xticks(rotation=45, ha="right")
plt.xlabel("Job Title")
plt.ylabel("Number of Openings")
plt.title("Open Positions by Job Title")
plt.tight_layout()
plt.show()
```