

Web Scraping with Java For Fun and Profit

Dynamic web pages

🔗 Concept

- ① Loads data dynamically via AJAX
- 1 Open page in Browser and find API endpoint with Developer Tools
- 2 Reverse engineer API call (params, header, cookies)
- 3 Replicate API call with Unirest and parse the data (XML, JSON, sometimes HTML)
- 4 Extract the desired data
- 5 Export the results

🔗 Unirest

- Make HTTP requests (GET, POST, ...)
- Synchronous and asynchronous requests
- Easy work with query string and route params
- Simple use of proxy
- Automatic JSON parsing

▶ Live example: Scraping results from peoplefinders.com

Static web pages

🔗 Concept

① Includes all data in the HTML

- 1 Find the exact URL
- 2 Examine HTML and find CSS selector
- 3 Download and parse page, extract data with CSS selector
- 4 Export the results

🔗 Jsoup

- Parse HTML from a URL, file, or string
- Find and extract data using CSS selectors
- Manipulate HTML elements
- Sanitize and output HTML
- Simple use of proxy

▶ Live example: Scraping the top 10 Google search results

Export the results

CSV (Comma separated value)

- Use PrintWriter or something similar
- Write a header first, then results line by line
- Can be opened by Numbers, Excel, Open Office for sorting, filtering, etc.

JSON

- Create a POJO to hold your data
- 🔗 Use Jackson Object Mapper to export as string or write to a file

Going undercover

🔗 Techniques

① Mimic a human being using a real browser as close as possible

- User-Agent string of real Browser or Google Bot
- Use a proxy to hide your identity
- Timing: use random waiting times between requests
- Send a Referrer Header
- Be aware of honeypot links