INFO-664-01
Programming For Cultural Heritage

Web Scraping
with
Python

Agenda

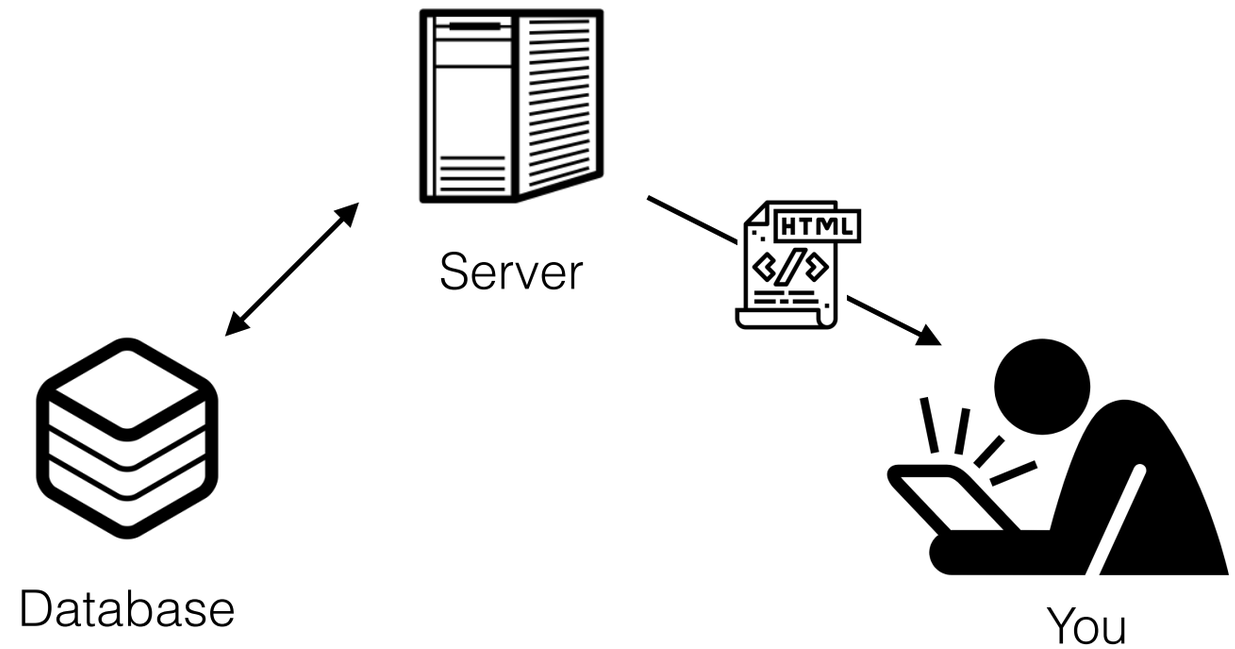
- Why / When to Web Scrape
- Traditional Web Scraping

Web Scraping

- To extract structured data from HTML when no other data option is available.

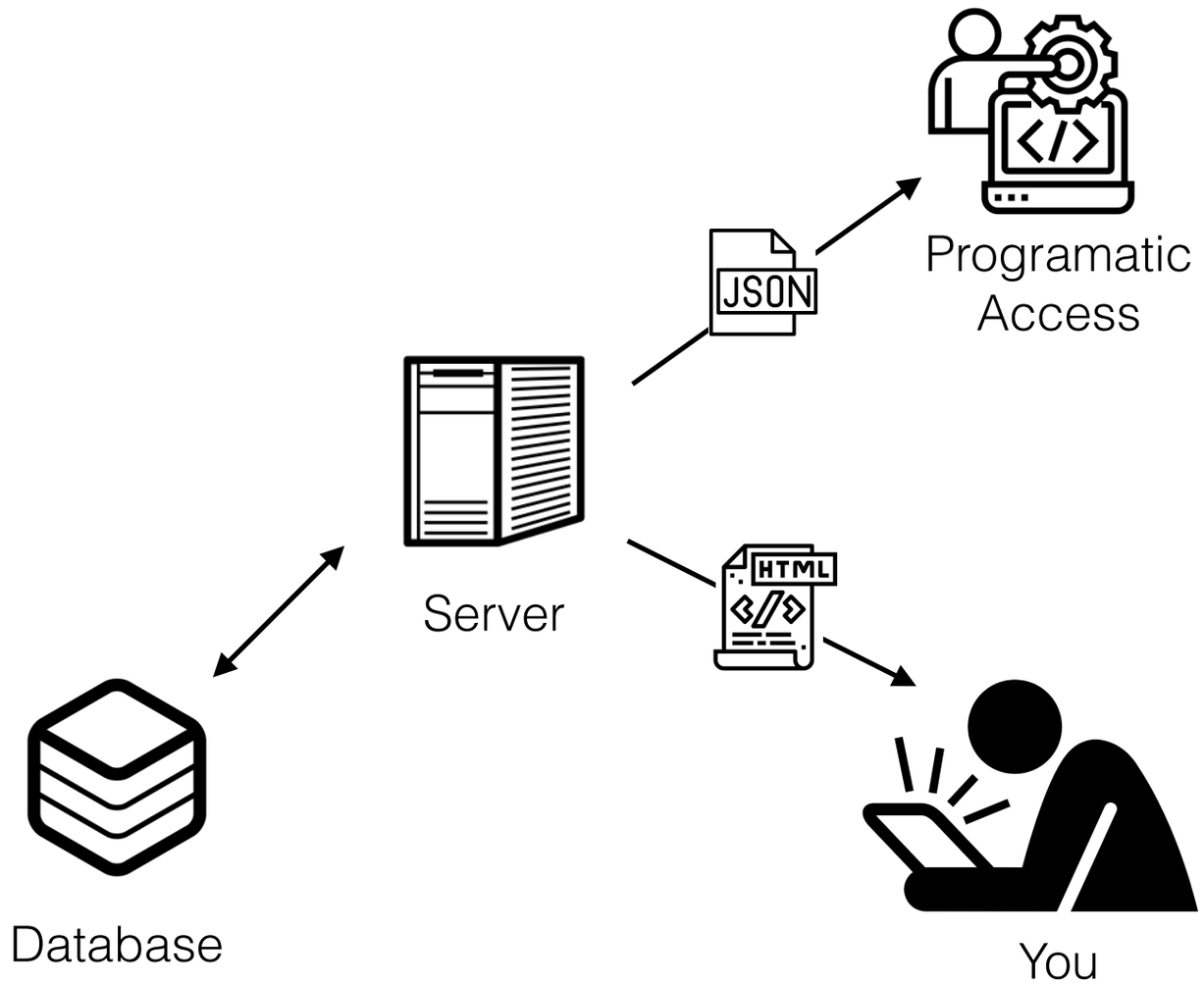
- Know when to scrape and when it is not needed.

Website Serving only HTML



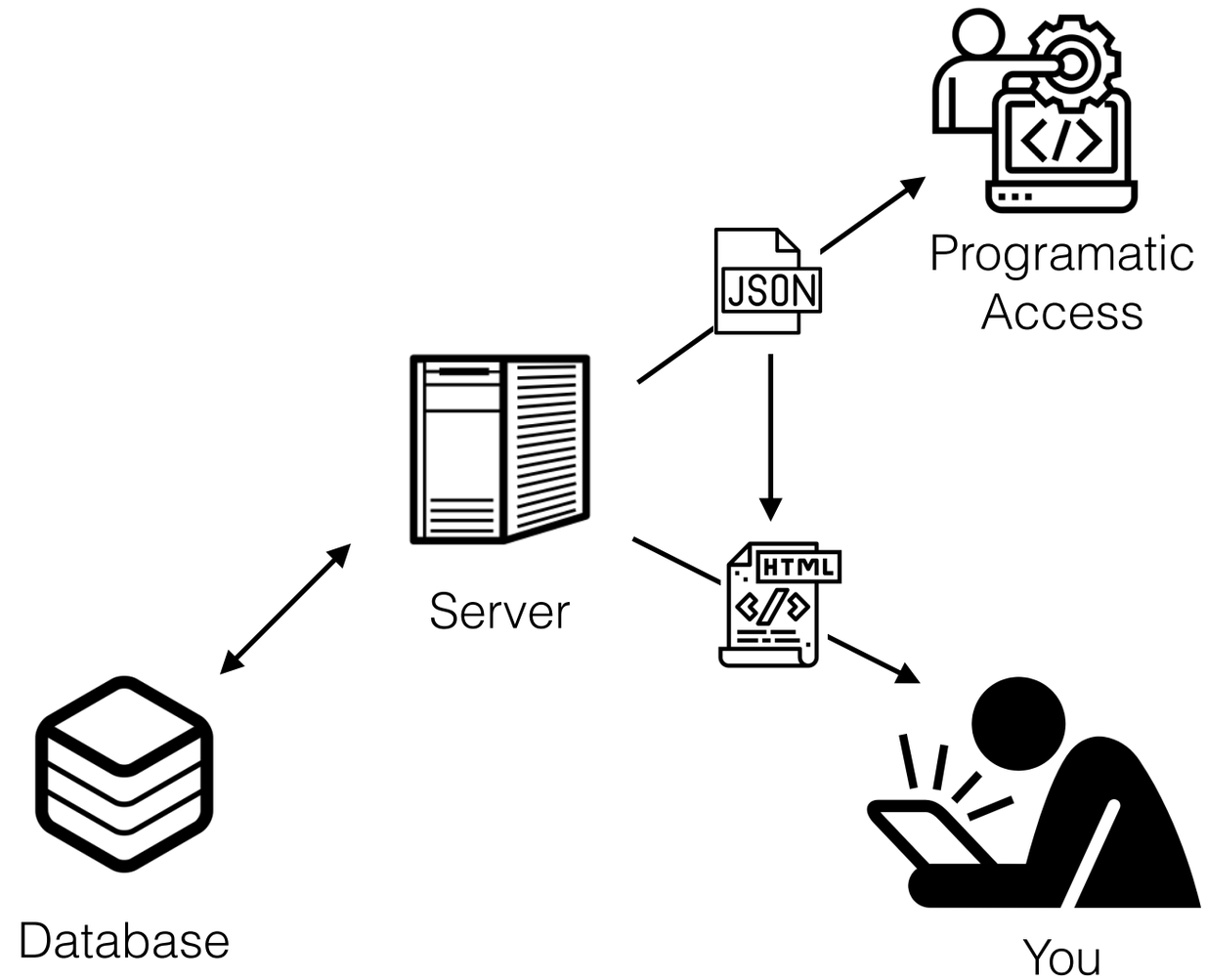
The only access to the database is via the HTML generated to send to your web browser

Website Serving HTML and API



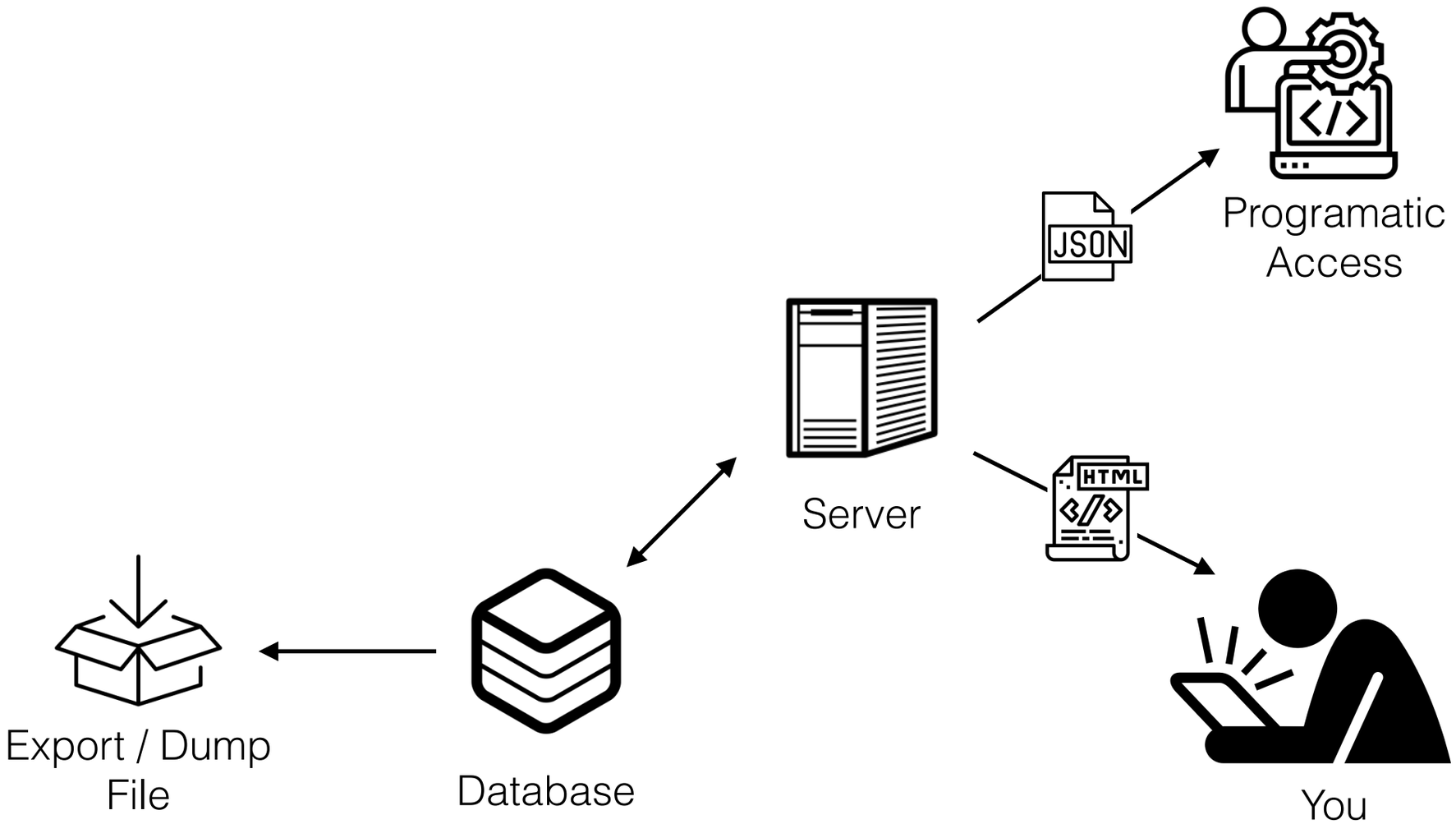
There might be an API that you can use to access the database

Website Serving HTML and (non-public) API



There might be an API that you can use to access the database

Website Serving HTML and API or EXPORT file



Or even better they have provided a data dump file of the contents of the dataset for you.

When to scrape

- You only need to scrape a website if the information you want is only available in the HTML pages
- If there is an API use the API
- If they provide a data dump file use that
- Scraping should be a last resort type of thing

When to scrape

- Be wary of legal concerns with scraping
- In our examples we are doing it for educational purposes

- Be polite when interacting with other's servers programmatically. Don't request too quickly, try to cache data when you can, etc.

High Level Workflow

- The goal should be:
  - Request the HTML pages from the website
  - Parse the HTML into a python data structure
  - Export that data for later use as JSON or CSV for example

Setup

- We are going to be using two libraries to scrape
  - Requests - for retrieving the HMTL
  - Beautiful Soup - parse the HTML and extract data
- We need to install these two libraries, make sure to:
  - `pip3 install requests`
  - `pip3 install beautifulsoup4`

Web Scraping Workflow

- Use request to get the text of the page
- Parse the text with beautiful soup
- Ask BS to look for specific features such as element names, class names, etc
- BS returns matching features, work with the data.

Web Scraping

- Make sure the HTML is really there in the file being sent from the server and is not dynamically being created with javascript. Use the “view source” feature to make sure the HTML is really there.

- Inspect the page using the browser tools (right click “inspect”) or do right click and view source to find patterns and features.

More Reading

- http://www.pythonforbeginners.com/python-on-the-web/beautifulsoup-4-python/

- https://www.dataquest.io/blog/web-scraping-tutorial-python/