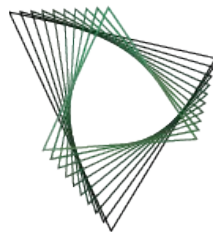


Funded by



Deutsche
Forschungsgemeinschaft

German Research Foundation



BERD
@NFDI

**With the support of BERD@NFDI and the Deutsche
Forschungsgemeinschaft (DFG, German Research Foundation).**

*The views expressed in this video/podcast/other (to be deleted as appropriate) do not reflect those of the **BERD@NFDI** or the **Deutsche Forschungsgemeinschaft** (DFG, German Research Foundation).*

Turning PDFs into Research Data

Unit One: Part One

Course Introduction



Instructor: John 'Jack' Collins (2024)

Who is this Course for?

You have access to data you want to use, but it's not in a neat table, it's spread across a bunch of documents!

Course Structure

- Unit One (Theory): Web Scraping
- Unit Two (Practical): Web Scraping
- Unit Three (Theory): Extracting Data from Text
- Unit Four (Practical): Extracting Data from Text
- Each week, we have an interactive session to talk about the material.

This course is meant to help you learn practical skills

- We encourage you to bring in projects from your own workplace
- It's ok if you don't have your own project, we provide examples from an actual use case

About the Instructor

- MSc Data Science
- PhD Candidate in Sociology (applying Machine Learning to Survey Methodology)
- IT consultant (in a past life)
- Much of the work in this course is based on a project Data Science for Social Good (credit goes to those researchers)

Disclaimer

I am not a lawyer, scraping and use of AI can have legal implications, if you're concerned consult an expert.

What is most likely not ok to do includes:

- Violate Terms of Service with a website
- Infringe on Copy Rights
- Disrupt a website
- Retain private information without permission

Turning PDFs into Research Data

Unit One: Part Two

Introduction to web scraping



Instructor: John 'Jack' Collins (2024)

Who is this Course for?

You have access to data you want to use, but it's not in a neat table, it's spread across a bunch of documents!

What is Web Scraping?



Any **information** you could go to a **website** and manually extract, in principle, should also be possible to **extract automatically** with a 'bot.'

- Automatically download all form templates from a commune website
- Refresh the latest inflation figures from the OECD website
- Compile all news stories about a given topic



Where you see a website with your eyes, the bot extracts and reads **the code which comprises the website.**



Therefore, we will need to understand **how the bot 'sees' the website** in order to formulate the code for how to extract what we want.

All information on a webpage is somehow represented as code

```
<> hello-world.html x
<> hello-world.html > html
1  <html>
2    <body>
3      Hello World!
4    </body>
5  </html>
```



= Hello World!

Turning PDFs into Research Data

Unit One: Part Three

How websites work



Instructor: John 'Jack' Collins (2024)

All information on a webpage is somehow represented as code

```
<> hello-world.html x
<> hello-world.html > html
1  <html>
2    <body>
3      Hello World!
4    </body>
5  </html>
```



= Hello World!

HTML = HyperText Markup Language

How webpages work: Code Languages

- HTML: Defines **how a webpage is structured** (i.e.: where textboxes, images, etc are placed and what is displayed in them).
- JavaScript: A code language **similar to python or R**, it executes complicated logic which HTML cannot. JavaScript often **outputs** html to be dynamically inserted into the main html file.


```
<!DOCTYPE html>
<html>
<body>

<h1>My First JavaScript</h1>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```



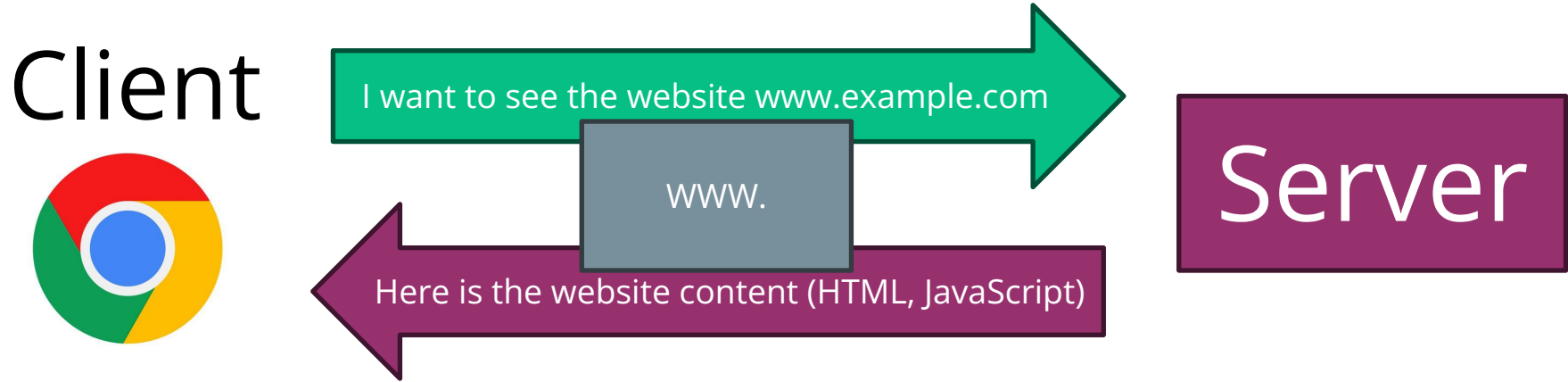
JavaScript inside the HTML

My First JavaScript

Click me to display Date and Time.

Thu May 23 2024 11:34:44 GMT+0200 (Central European Summer Time)

How webpages work: Requests and Queries



- Your browser (Chrome, Firefox, Edge, etc) is a program on your computer that executes the website code to render the website you see.
- **TAKEAWAY: Your R or Python code might need to include a 'Driver' for a browser because R and Python does not compile JavaScript on its own.**
- Often, the server doesn't send all the data at once. Instead, the website allows you to send **queries** which request bits of data from the server. These queries are just like **SQL** which you are familiar with.

Turning PDFs into Research Data

Unit One: Part Four

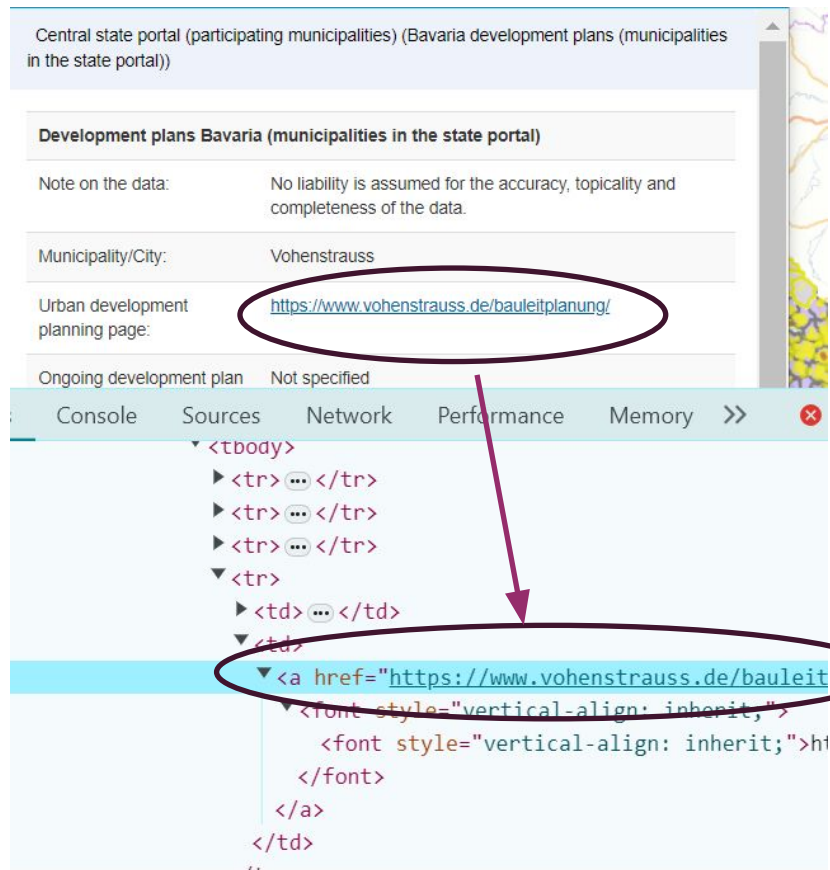
Inspecting a Website



Instructor: John 'Jack' Collins (2024)

Using “Inspect Element”

- In the example in our exercises, we will extract PDF documents from links on a webpage.
- But first, I need to understand how my bot can find the right link.
- All browsers have an ‘**inspect element**’ option which lets us see the HTML content which corresponds to anything you see on the page.
- All pieces of HTML contents have an address called an “**X-Path**.”
- In the exercise, we’ll see how to use this to make instructions for our web scraping bot.



HTML Xpath to a link

```
/html
/body
/div[9]
/div[3]
/div[1]
/div[1]
/div
/div[2]
/div
/table
/tbody
/tr[4]

```

href="https://www.markt-koenigstein.de/immobilien/index/kategorie/cat/6/bauland"

Zentrales Landesportal (teilnehmende Gemeinden) (Bauleitpläne Bayern (Gemeinden im Landesportal))	
Bauleitpläne Bayern (Gemeinden im Landesportal)	
Hinweis zu den Daten:	Für die Richtigkeit, Aktualität und Vollständigkeit der Daten wird keine Gewähr übernommen.
Gemeinde/Stadt:	Königstein
Bauleitplanungsseite:	https://www.markt-koenigstein.de/immobilien/index/kategorie/cat/6/bauland
Laufende Bauleitplanverfahren:	Keine Angabe
Abgeschlossene Bauleitplanverfahren:	Keine Angabe

How do I harvest each URL like this?: Harvest the link at each 'href' at each 'td[2]' address.

Interacting with a Website

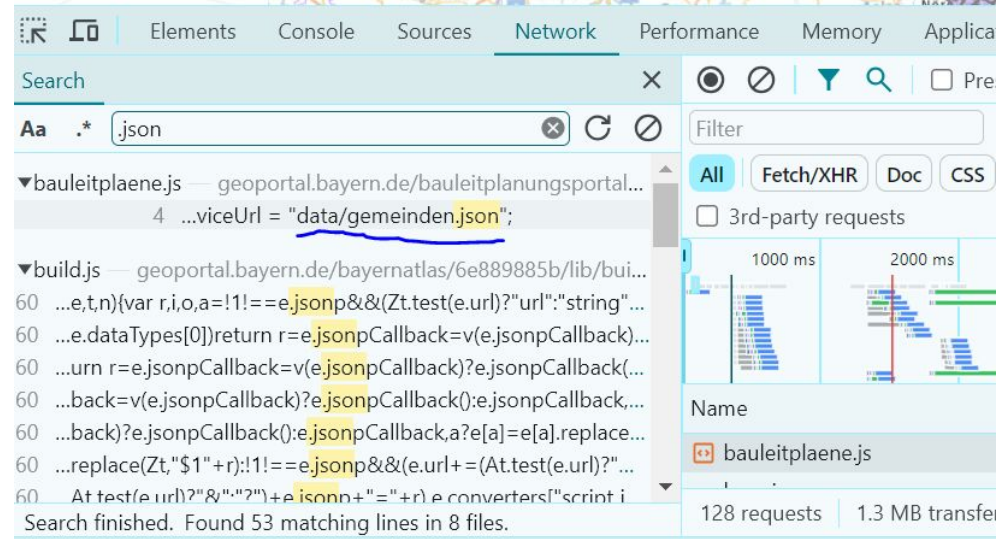
- The content you want to scrape **might not be available as soon as you load a web page** though, it might only **become available when you interact with the webpage**.
- In principle, anything you can do manually, you will also be able to instruct your bot to do.
- Click a button to make new content appear (so you can scrape it)
- Type text into a search bar and hit the 'search' button to bring up new results
- Scroll across a page to load new content

How webpages work: Files, Endpoints, and Queries

- Often, you will be seeking data which is located on the **Server** but it is difficult to make it appear on the webpage.
- For this, you want to understand how to **Query the Server** to make the Server send you the desired data.
- The data might be in a file on the Server, or in a database.

How webpages work: Network Inspector

- As you use a website, we can see what queries your browser sends to the server with the **Network Inspector Tool** and thereby determine where the data is and possibly, how to access it directly with our bot.
- Common things to search for when looking for endpoints in the network tab include:
 - XML, JSON, ?, PHP, CSV, YAML
- The practical benefit is that you don't need to think about clicking buttons or finding URLs, you can access the data directly.



Example Query

REQUEST:

GET <https://api.example.com/users.php?id=1>

I want the details
of the 'user' with
ID '1'

The URL for the "endpoint"

The name of a 'php' file on the server,
which contains a function that
takes an 'id' parameter.

RESPONSE:

```
{  
  "id": 1,  
  "name": "John Doe",  
  "email": "john.doe@example.com",  
  "created_at": "2023-05-01T12:00:00Z",  
  "updated_at": "2023-05-22T14:30:00Z"  
}
```

Question: What if I wanted to get the data of all users?

Takeaway: By learning what queries your website sends to the server, you might learn how to send your own queries to the endpoint and extract data more conveniently.

Turning PDFs into Research Data

Unit One: Part Five

Things to Know



Instructor: John 'Jack' Collins (2024)

Queries: Things to Know

- Typically, when you send a query, you also send information about who you are.
 - IP Address
 - User-Agent data
 - Cookies (data which your computer holds onto so the server can identify you – essential for user sessions!)
- If your bot sends a query without a User Agent or some Cookie Data, the query might get rejected because this information is expected by the Server. We'll go over how to do that in the exercises.

Web Scraping: Things to Know

- Be careful not to send too many requests to a server too fast. (This could damage the web site!). Best practice is to load a webpage as few times as possible.
- You can use 'wait' functions which makes your program wait between requests.
- You can download files from links, but you can also download metadata about the file from the website.
- A good practice is to name (or store) downloaded files in a way you can link back to where you got it from on the website.
- Iframes: websites within websites
- JavaScript objects: content you can see, but not scrape (but you can possibly query).
- Be aware you may download many GBs of data!

Turning PDFs into Research Data

End of Unit One



Instructor: John 'Jack' Collins (2024)