



Economia, Finanza e Analisi dei Dati

Laurea Magistrale

Data Analysis for Economics

Surfing the Web with Python

Marco Nieddu

Fall 2025

Who are we building on?

- ▶ This lecture builds on the lecture slides from Jared Hutchins
<https://github.com/jphutch/ACE-592-SAE/>.
- ▶ And we are going to do some web surfing on
 - ▶ the MIMIT webpage containing the data we used in the previous classes
<https://www.mimit.gov.it/index.php/it/open-data/elenco-dataset/carburanti-archivio-prezzi>
 - ▶ and the clone version we created in order to minimize pressure on MIMIT servers:
<https://garaugarau.github.io/DAEc/>

Surfing the Web with Python

- ▶ An absolutely vital skill for data science is being able to interface with websites from Python.
- ▶ **Why?**
 - ▶ Makes data collection much easier.
 - ▶ No tedious surfing.
 - ▶ Can automate tasks that are usually time-consuming.

Things to know

- ▶ What “http” is.
- ▶ How to send requests via “http” using the requests package.
- ▶ How to download and process JSON data.
- ▶ How to download and process HTML data.

What is “http”?

- ▶ Hypertext Transfer Protocol (HTTP) is the way clients (e.g., your computer) talk to servers (e.g., websites).
- ▶ The “user agent” (usually a web browser) requests a resource, and the server responds with an HTML file or another resource.
- ▶ While the web browser is a high-level “user agent,” we can use Python as our user agent instead.

There is way more to this than we need to know...

What's the upshot?

- ▶ Using Python as your user agent allows you to send requests to websites and get back responses without opening a browser window.
- ▶ Since some information exists statically on a webpage, it is sometimes easier to just pull the info when you need it rather than saving it.
- ▶ It also can be used to automate many tasks for you.

Some terminology

- ▶ **GET**: a common way to “request data.”
- ▶ **POST**: a common way to “post data” to a server.
- ▶ **HEAD**: makes a “get” request but doesn’t return anything (good to test whether your request works or not).
- ▶ **PUT**: like post, except will not produce multiple resources if done multiple times.
- ▶ **DELETE**: delete the resource.
- ▶ **Status Code**: a number your request gives you to indicate success or failure.

Many more, but these are the most relevant to us.

The Golden Rule of STATUS Codes

- ▶ If it starts with a 2? **GOOD**
 - ▶ 200 OK
 - ▶ 201 Created
 - ▶ 202 Accepted
- ▶ If it starts with a 4 or 5? **BAD**
 - ▶ 403 Forbidden
 - ▶ 404 Not Found
 - ▶ 405 Method Not Allowed
 - ▶ 500 Internal Server Error
 - ▶ 501 Not Implemented
- ▶ Something else? **look it up**
 - ▶ 3xx are redirection requests (needs more info).