Scraping

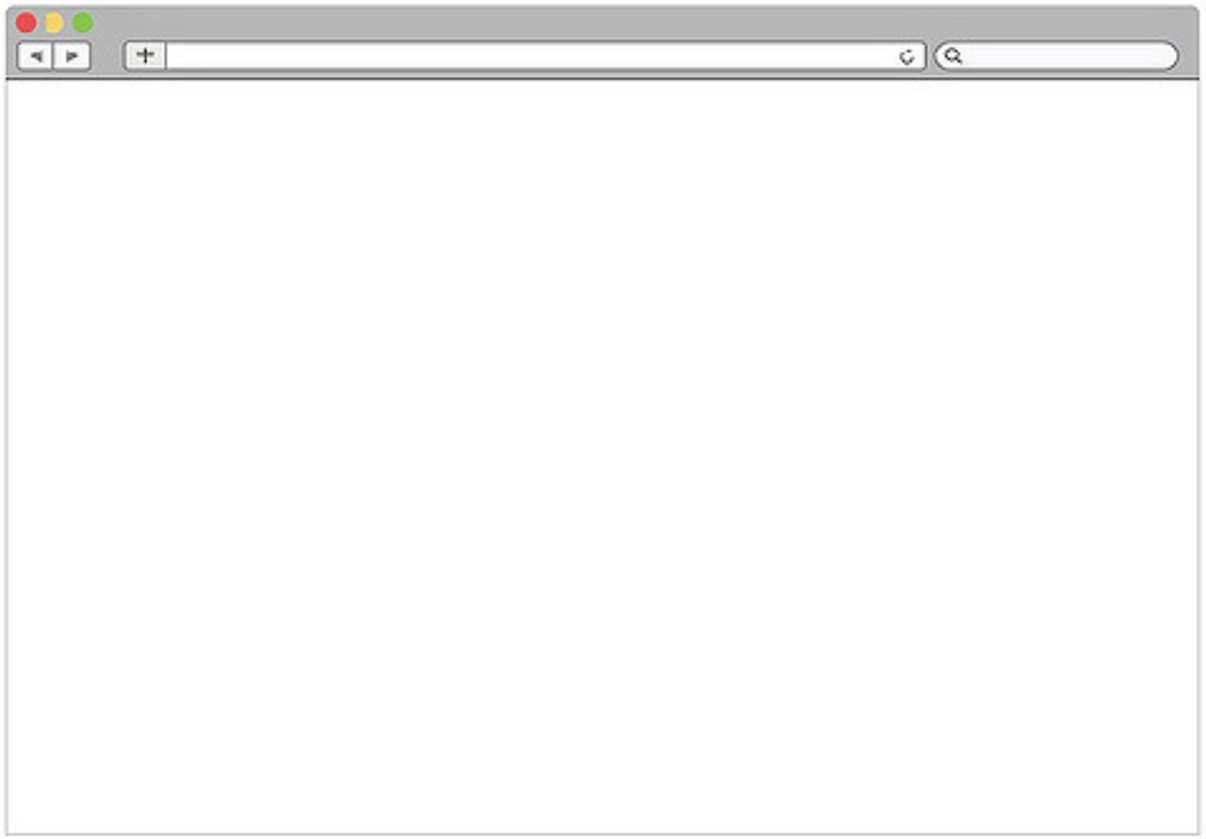
collecting data from websites

Goals

- Understand how information is accessed over the internet
- Discover tools to automatically collect this information
- Running example: airbnb.com

Loading a website

Web browser



Server connected to the internet

Loading a website

Web browser

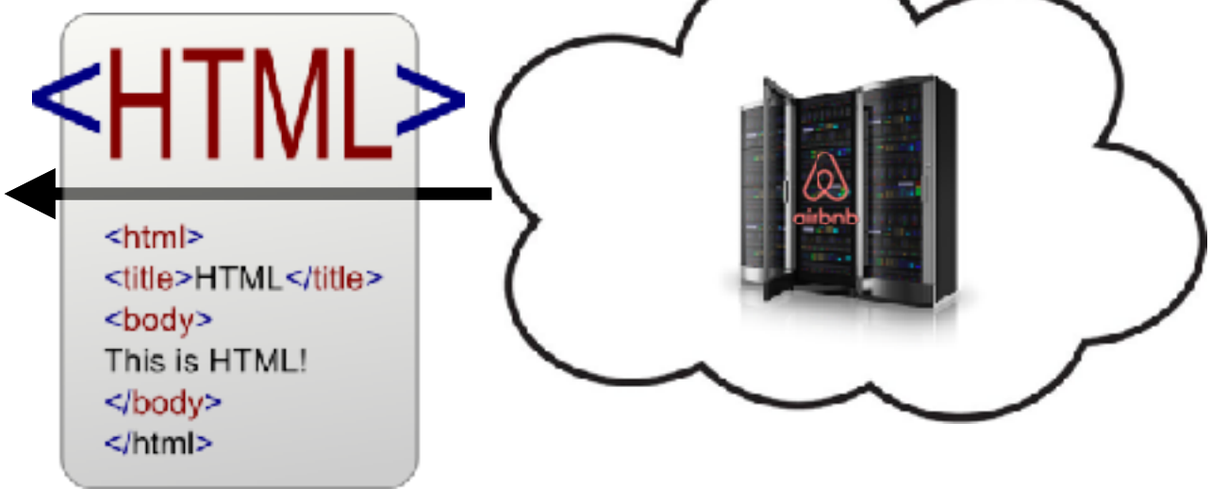
Server connected to the internet

The browser sends an HTTP request:
https://www.airbnb.com/rooms/6412281

Loading a website

Web browser

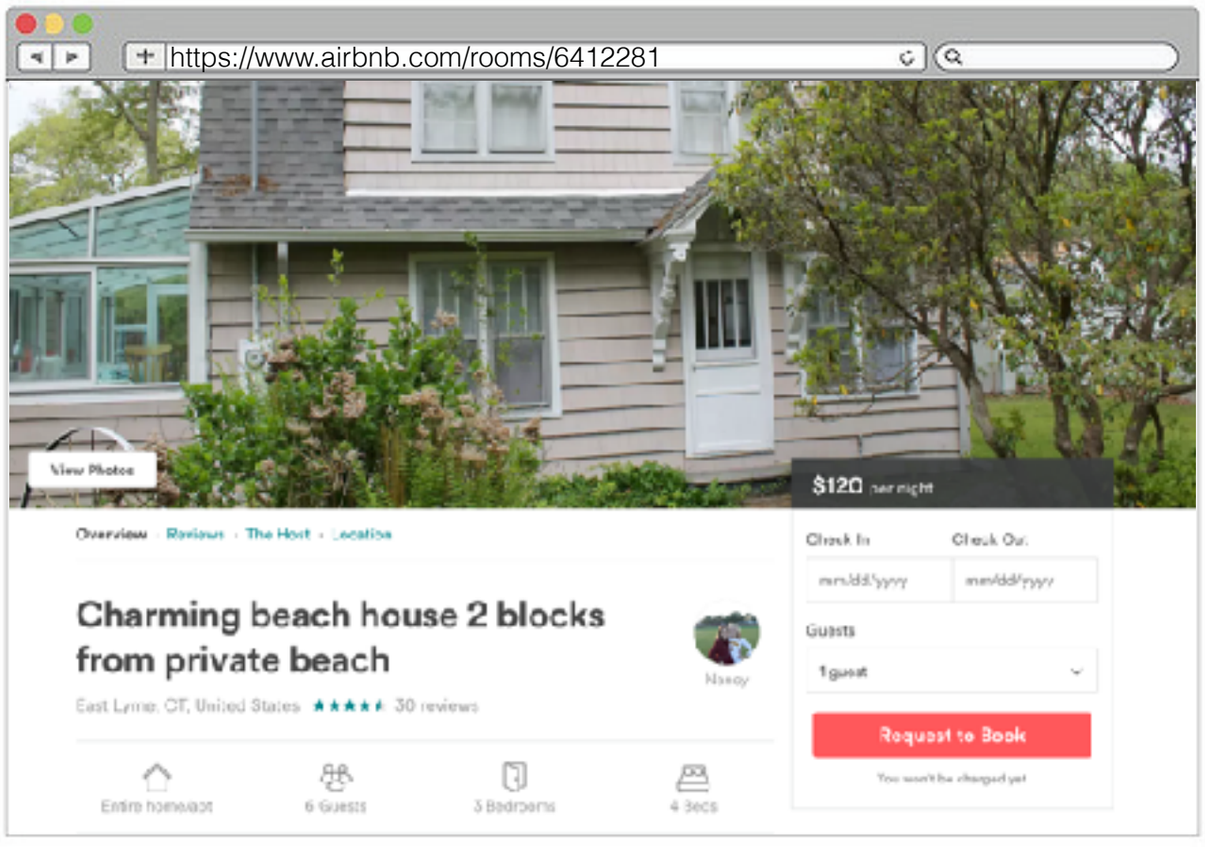
Server connected to the internet



The server returns some HTML

Loading a website

Web browser



Server connected to the internet

HTML is a markup language to describe what a page should look like. The browser interprets it to display the page.

HTML

What does it look like? How can we see it?

Tools

- Browser’s HTML view

Demo

Viewing and interacting with HTML in the browser

Tools

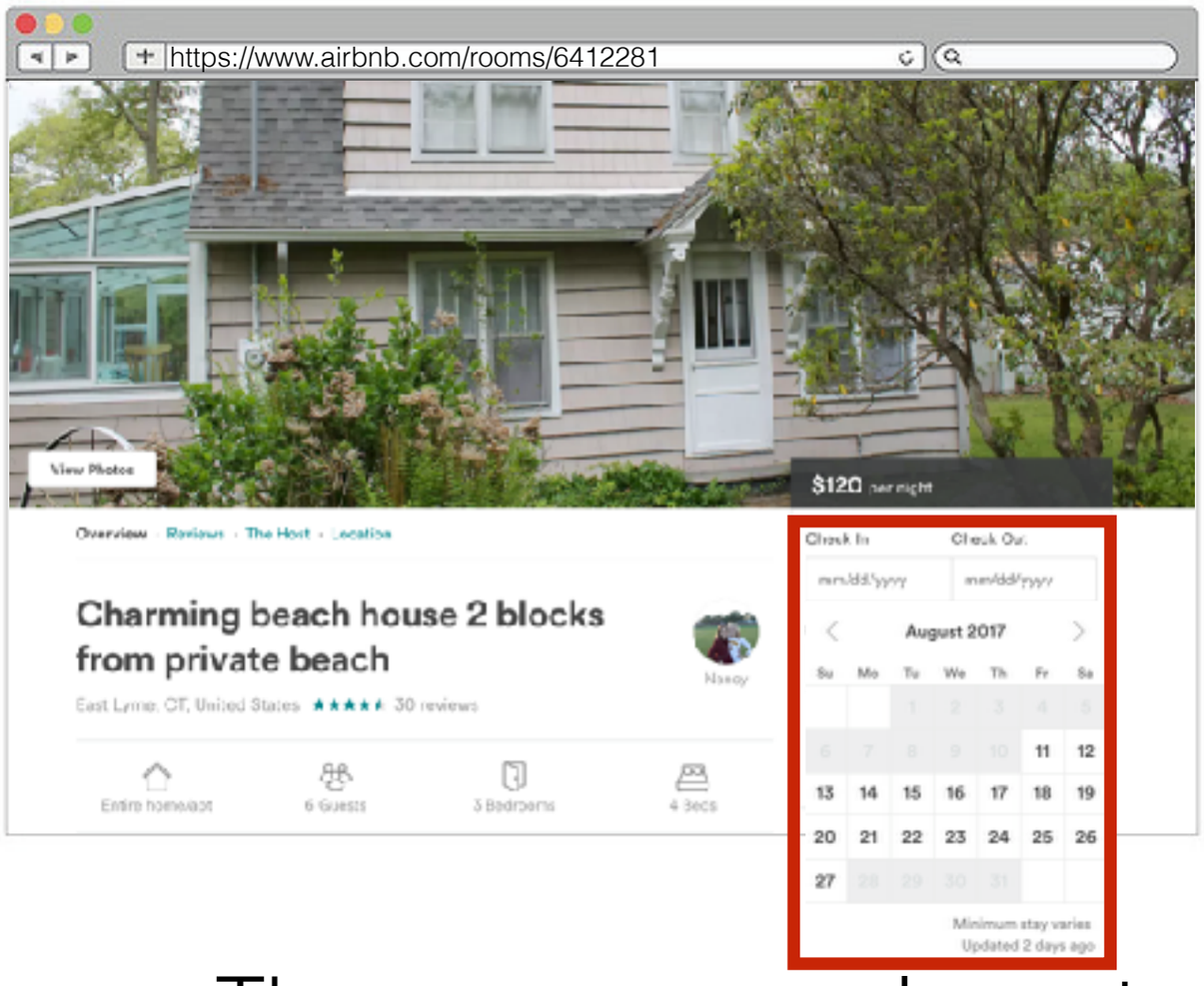
- Browser's HTML view
- Python! (requests, beautifulsoup)

Code

Programmatically interacting with HTML

It’s a bit more complicated…

Web browser



Server connected to the internet

The server can also returns some javascript, a full programming language that the browser will execute.

Javascript

- Javascript allows websites to perform complex actions and interact with the HTML.

- It makes scraping web pages harder because it is not easy to mimic the browser's javascript execution.

- BUT: javascript is often used to interact with formatted data through APIs!

Tools

- Browser's HTML view
- Python! (requests, beautifulsoup)
- APIs

APIs

{“listing_id”: 6412281,
“bedrooms”: 3,
“bathrooms”:2,
“ratings”: 4.5}

HTTP request

Formatted data
e.g. json

Server connected
to the internet

Application Programming Interface

Tools

- Browser's HTML view
- Python! (requests, beautifulsoup)
- APIs
- Browser's network view

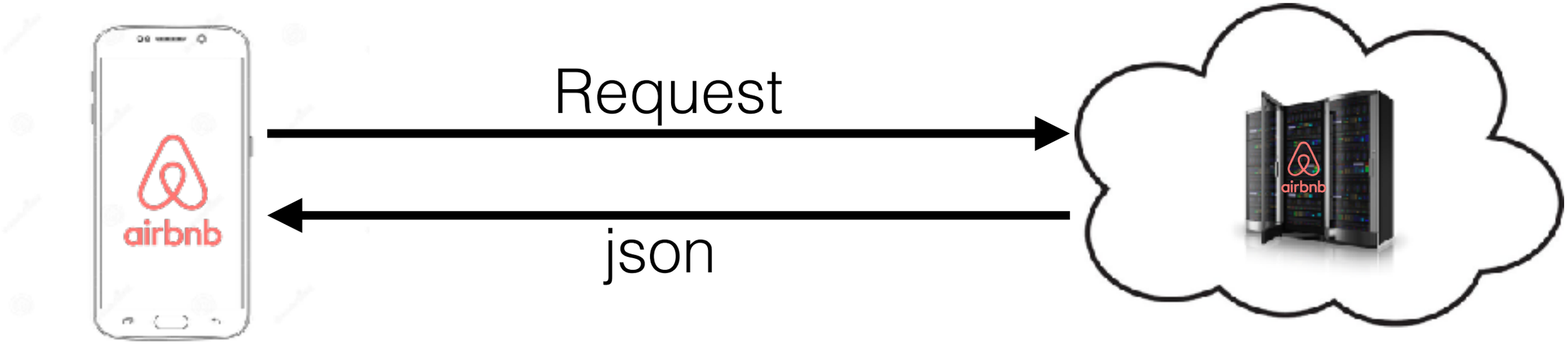
Demo

Discovering APIs with a browser's network view

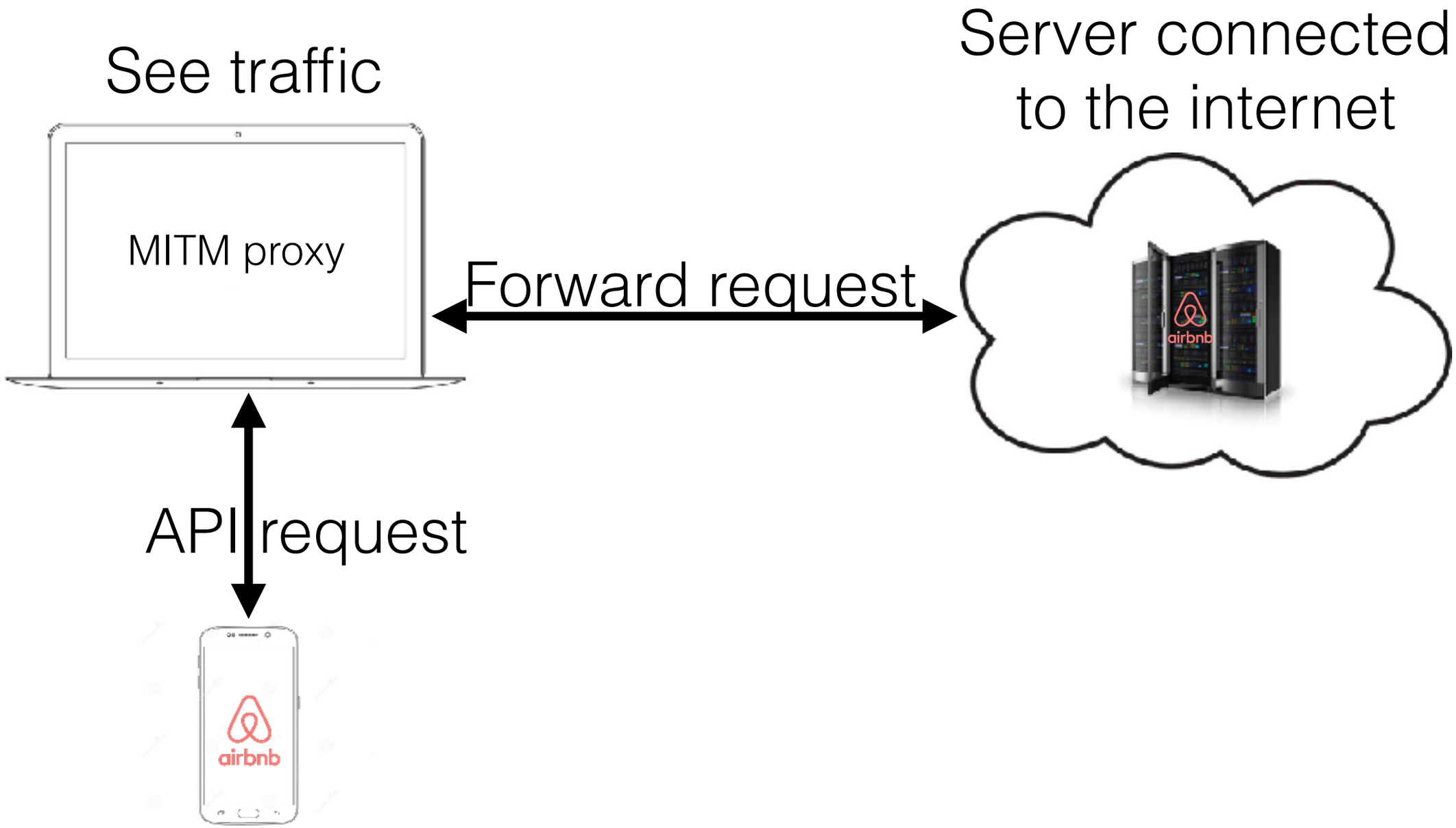
Code

Querying APIs

Mobile apps


Server connected
to the internet
Request
json
airbnb
airbnb

Mobile apps & MITM proxy



Demo

Discovering APIs used by mobile apps with MITM proxy

Tools

- Browser's HTML view
- Python! (requests, beautifulsoup)
- APIs
- Browser's network view
- MITM proxy and smartphone

Open APIs

- Sometimes web services want to give access to their data: it usually happens through an official API
- Example: NYT articles

What if there is no API?

We can use automated browsers, like:

- Selenium (python bindings)
- PhantomJS (in javascript, headless. Can be used with python through selenium)
- Mecanize (no javascript)
- …

Tools

- Browser's HTML view
- Python! (requests, beautifulsoup)
- APIs
- Browser's network view
- MITM proxy and smartphone
- Automated browsers

Demo

selenium