Introduction To Forging API Requests

Main Ways Websites Get Data

Server Side Rendering

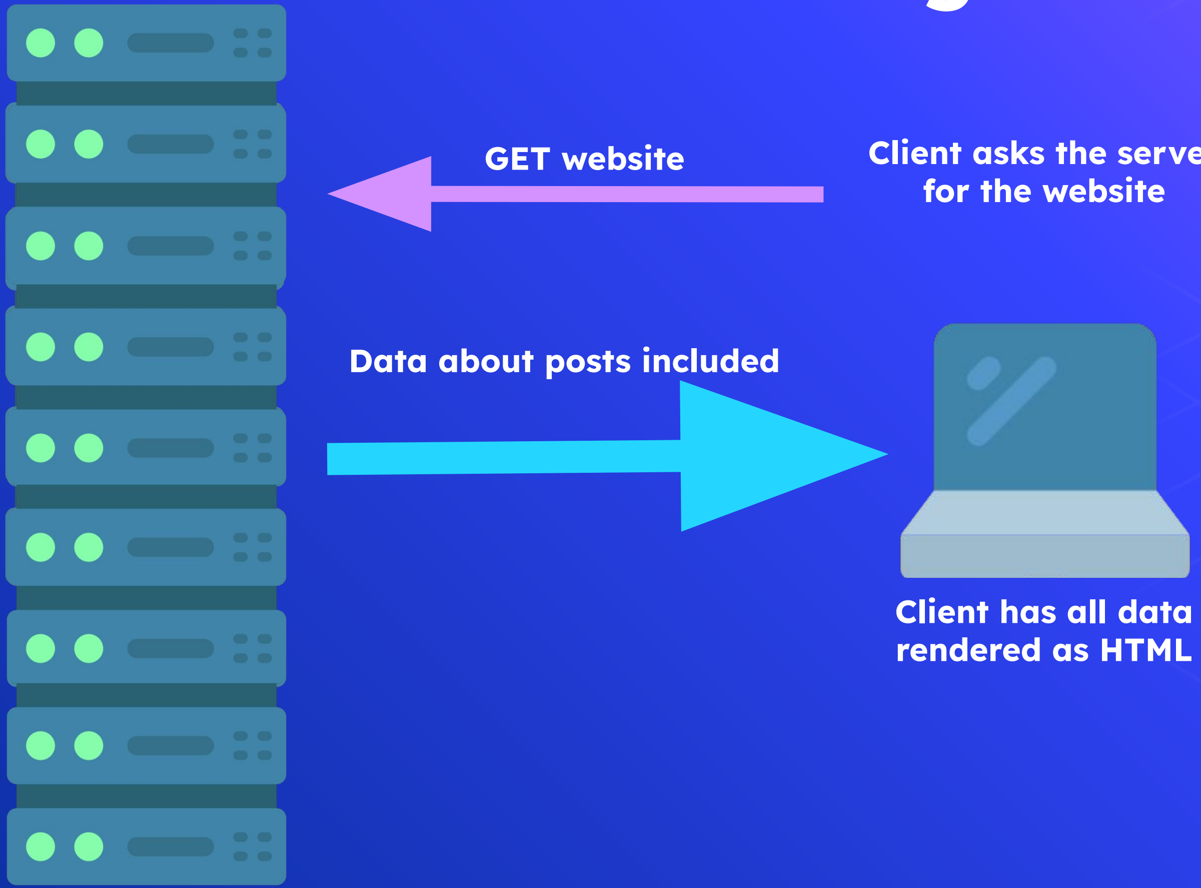
- Data is sent as part of the HTML file to the requester
- Each new request for data requires a page reload

AJAX

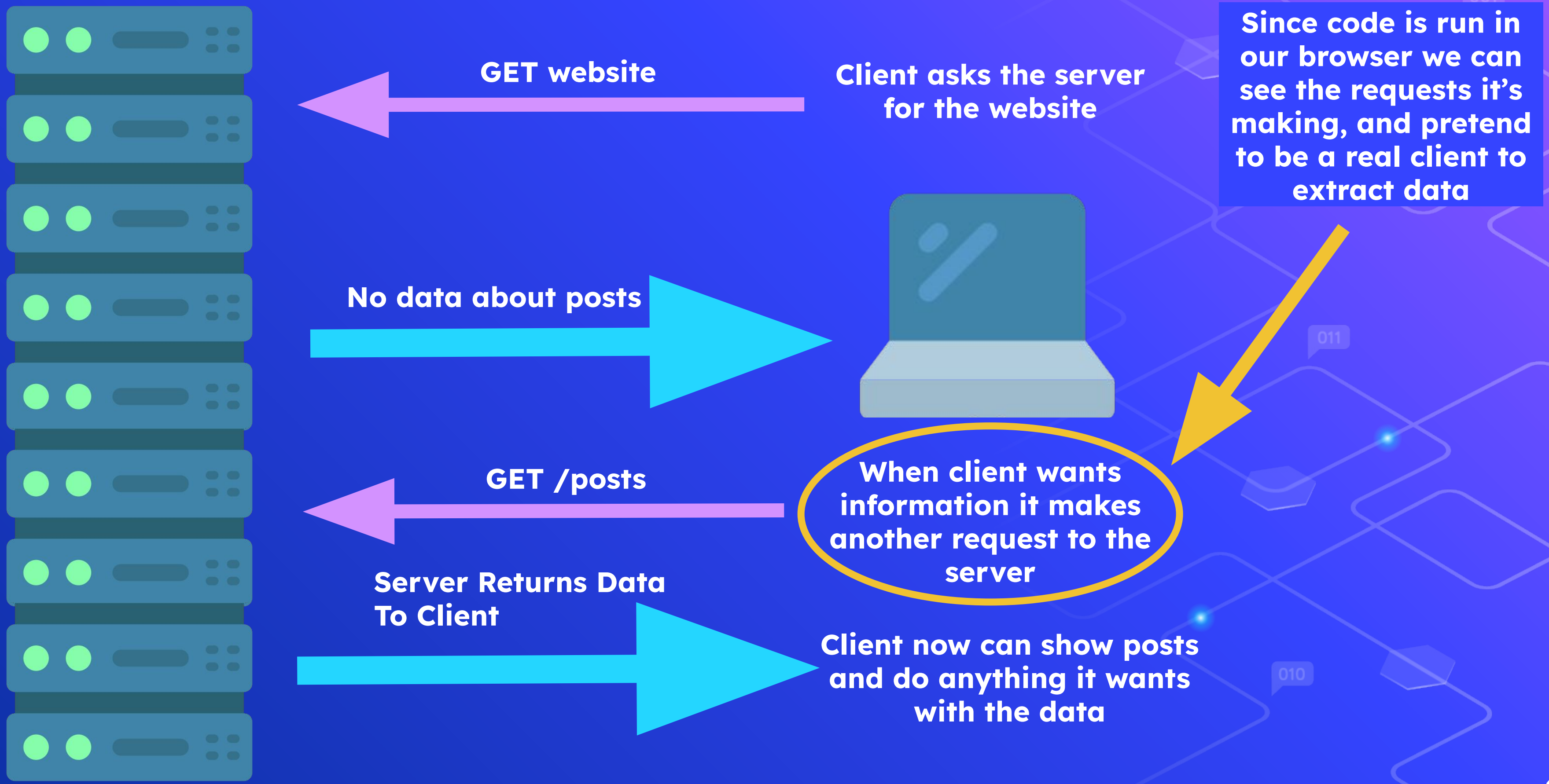
- Client is able to request new information from a server
- Allows client to update itself without refreshes

Note: This is to a simplification, but this is all you really need to know for this specific lesson

Server Side Rendering


GET website
Client asks the server for the website
Data about posts included
Client has all data rendered as HTML

AJAX



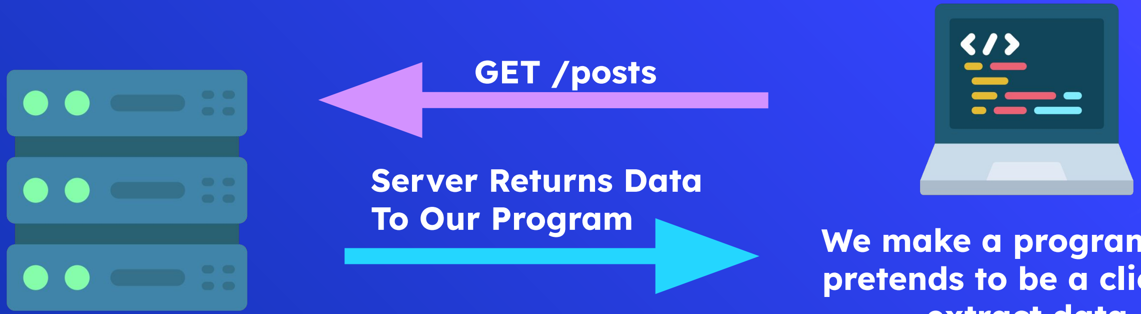
Client asks the server for the website

Since code is run in our browser we can see the requests it’s making, and pretend to be a real client to extract data

When client wants information it makes another request to the server

Client now can show posts and do anything it wants with the data

How Do We Exploit This?



We make a program that pretends to be a client to extract data

Forging Requests

The idea is to write programs that pretend to be a legitimate client. This way you can programmatically extract data at scale.

Advantages To Forging Requests

- These APIs can be easier to scrape at scale
- They may contain extra information you can't see in the website
    - Similar to how Missouri teachers had their SSNs leaked (The Verge)
- Less data returned means quicker requests (and less data transfer fees)

Disadvantages To Forging Requests

- Some websites constantly update their APIs
  - Extra work has to be done to keep up with these changes
- Can be hard to emulate human behavior to avoid captchas and other blocking mechanisms

Thanks!

Any questions?

Twitter: @david_teather

Everything Else: @davidteather

Credits

- Presentation template by SlidesCarnival
- Photographs by Unsplash