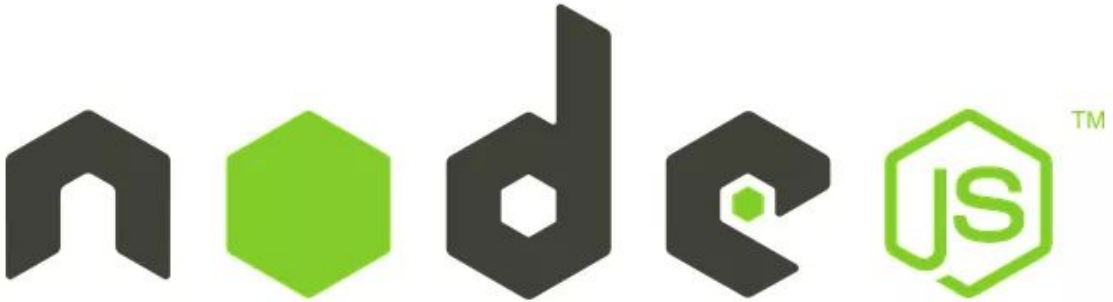


Introduction to Server Side

Week 4, Day 1

What is a server

As the name implies, it’s built to serve, and guess who is typically served?

It’s a computer, designed to be efficient in serving requests.

It’s connected to the Internet 24/7.

A server also could send requests to other servers just as the client does.

Server-side code requirements

To write server-side code, we need a general purpose programming language that can do all of the following:

- Run a web server
- Read from and write to the database
- Read from and save files
- Handle authentication
- ...

There are many languages that can do this: PHP, Java, Ruby, Python...

What is nodeJS

JavaScript was designed for the browser and does not support writing server-side code out of the box.

NodeJS is a run-time environment that allows you to develop the server side of your application using JavaScript.

What makes NodeJS unique and especially useful to a full-stack developer is that you can use the same language on the client side (browser), and server-side (back-end).

Warning

- JavaScript running on the server-side isn’t exactly the same as JavaScript running in the browser

- You won’t have access to the browser APIs like local storage, window(DOM) and others

- However you will access to other technologies, like database and your code will be **hidden** from the client.

Static or Dynamic Server

- Behind GitHub pages that you have been using in the last 3 weeks there is also a server, but it only serves static files… meaning it will only respond with the file as it is
- So if your server needs to respond with dynamic content, content from the database, etc… it needs to be a dynamic server
- Since GitHub pages doesn’t support that we need to look for other alternatives
- Heroku is a web hosting provider, more or less like Amazon web service or Google cloud platform

Can’t my PC do the job?

Yes it can, and you will always use your PC as a local server in the development environment.

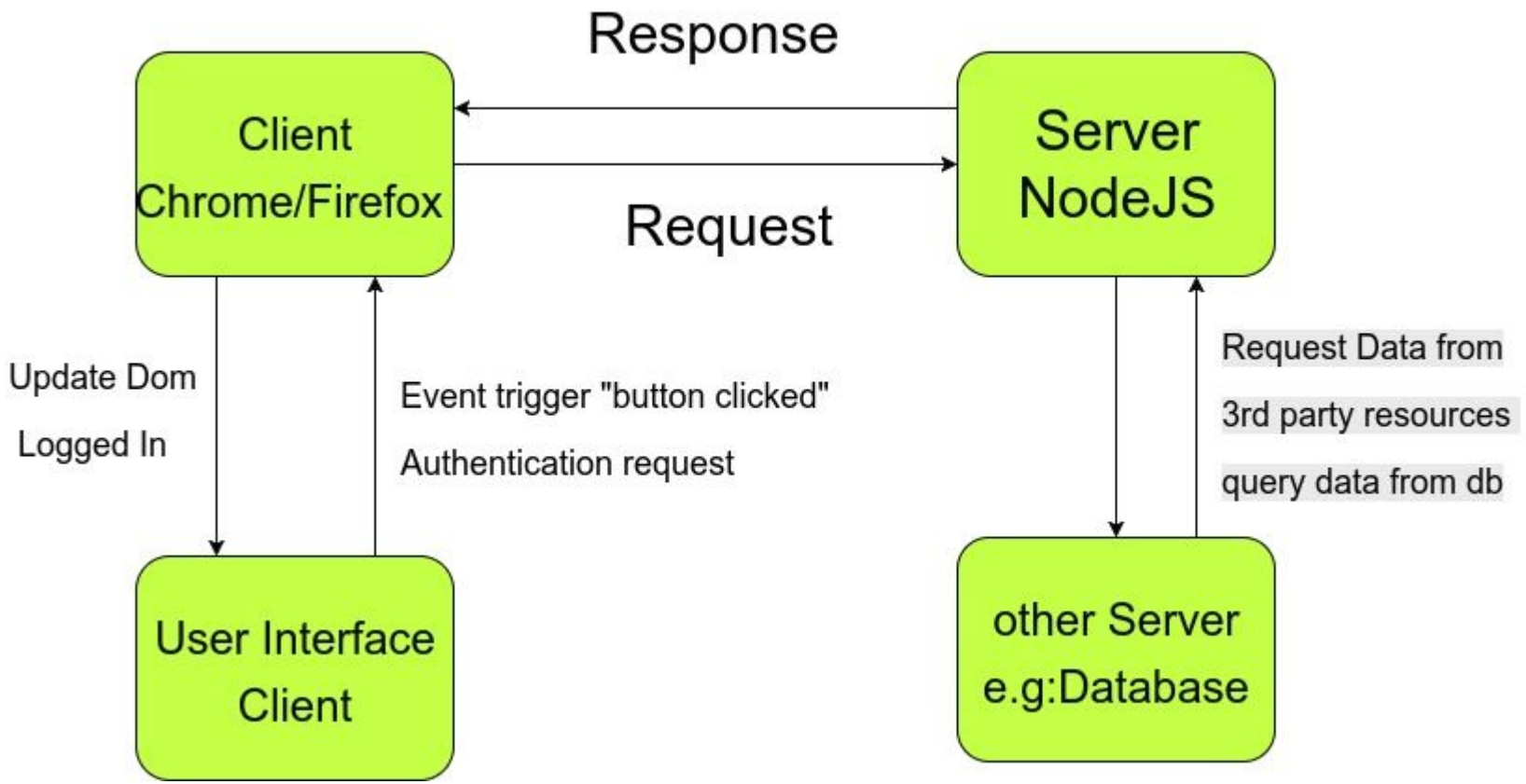
In production environment, it’s not reliable, you would need to keep your PC connected to internet 24/7 and you would need to ask your ISP for a static IP address.

And if your PC restarts suddenly… your website/service will be down!

How the Server-Side of an app works

The server-side of an app handles all client requests and sends responses

- Each client request has a url, method, headers, cookies, body… and more.
- The server checks the url, then it asks a handler (function) to handle the request. The process of matching handlers with the url is called **routing**.
- Then another function may check for other metadata (e.g. depending on if the client is requesting public or private data).
- The server finally responds to the client.


Response
Client
Chrome/Firefox
Server
NodeJS
Request
Update Dom
Logged In
Event trigger "button clicked"
Authentication request
Request Data from
3rd party resources
query data from db
User Interface
Client
other Server
e.g:Database