**Node-Express-Mongo Theory**

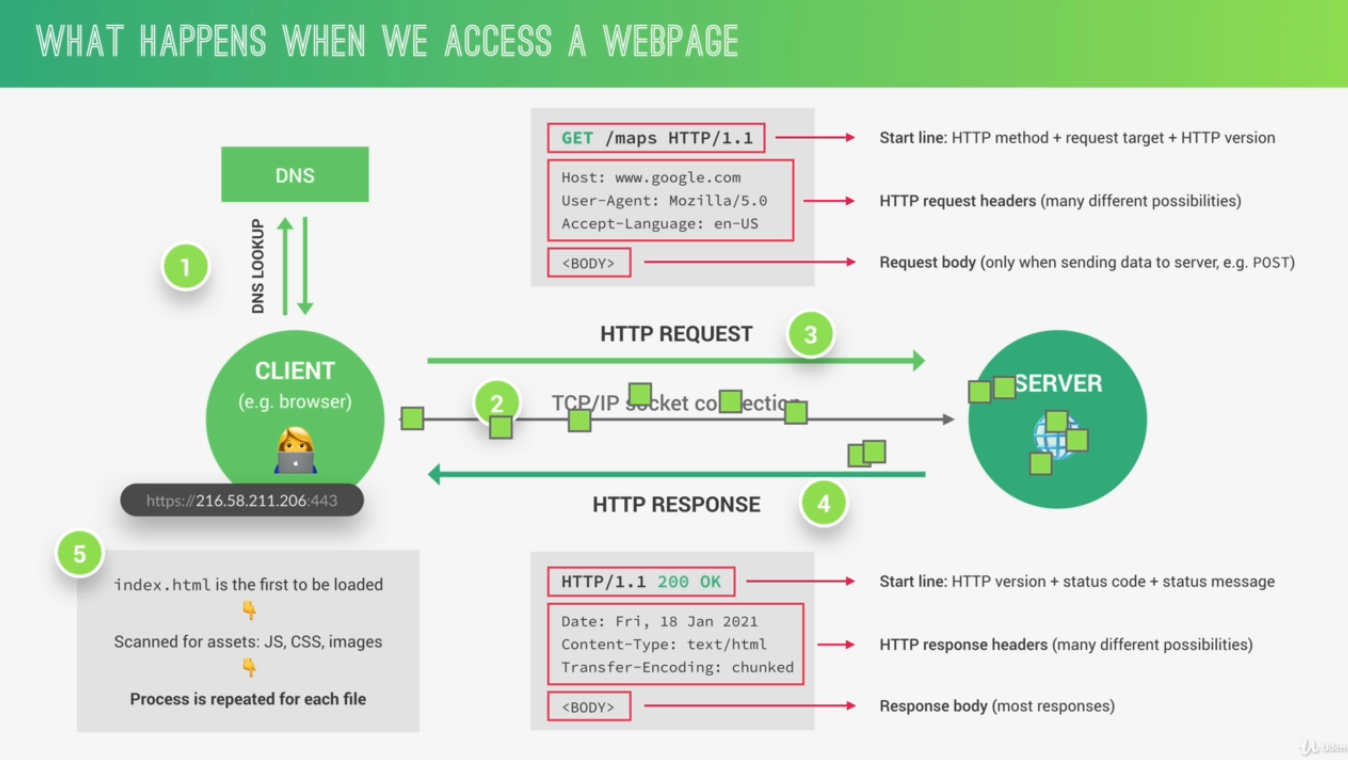
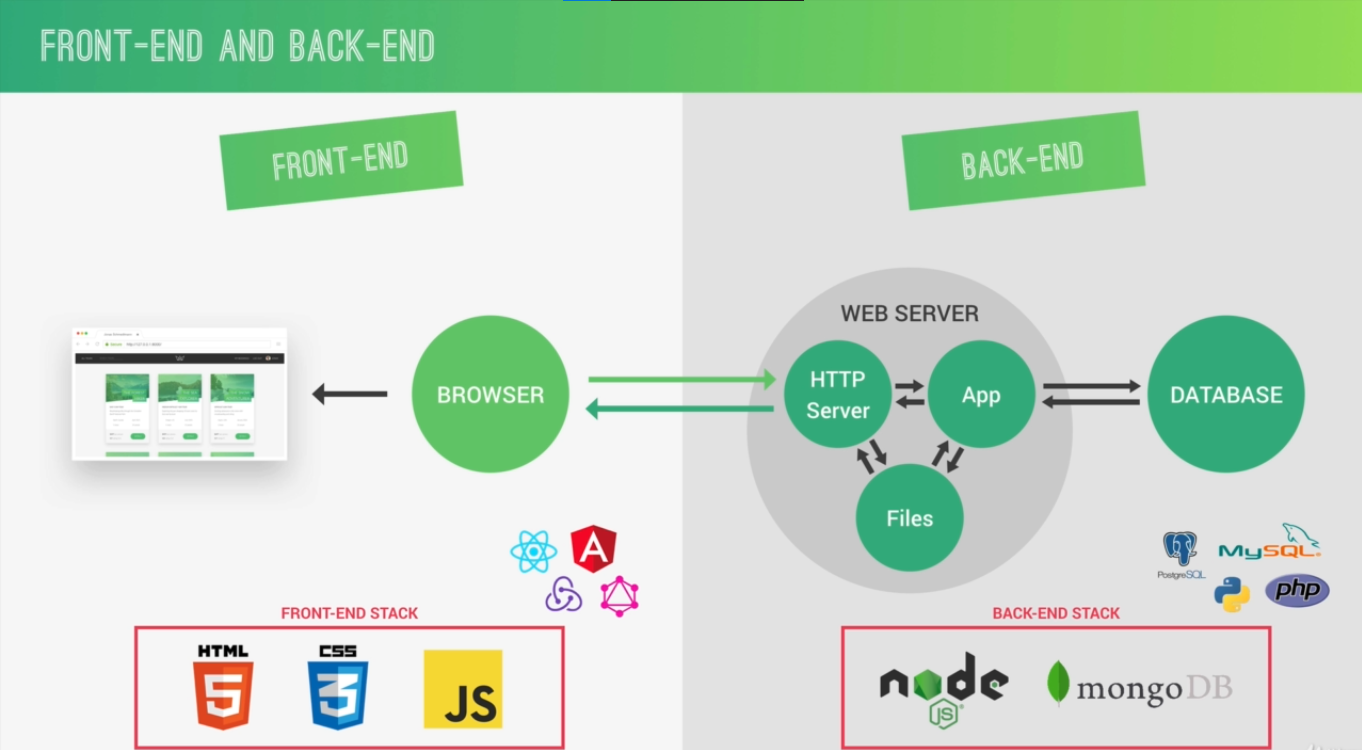
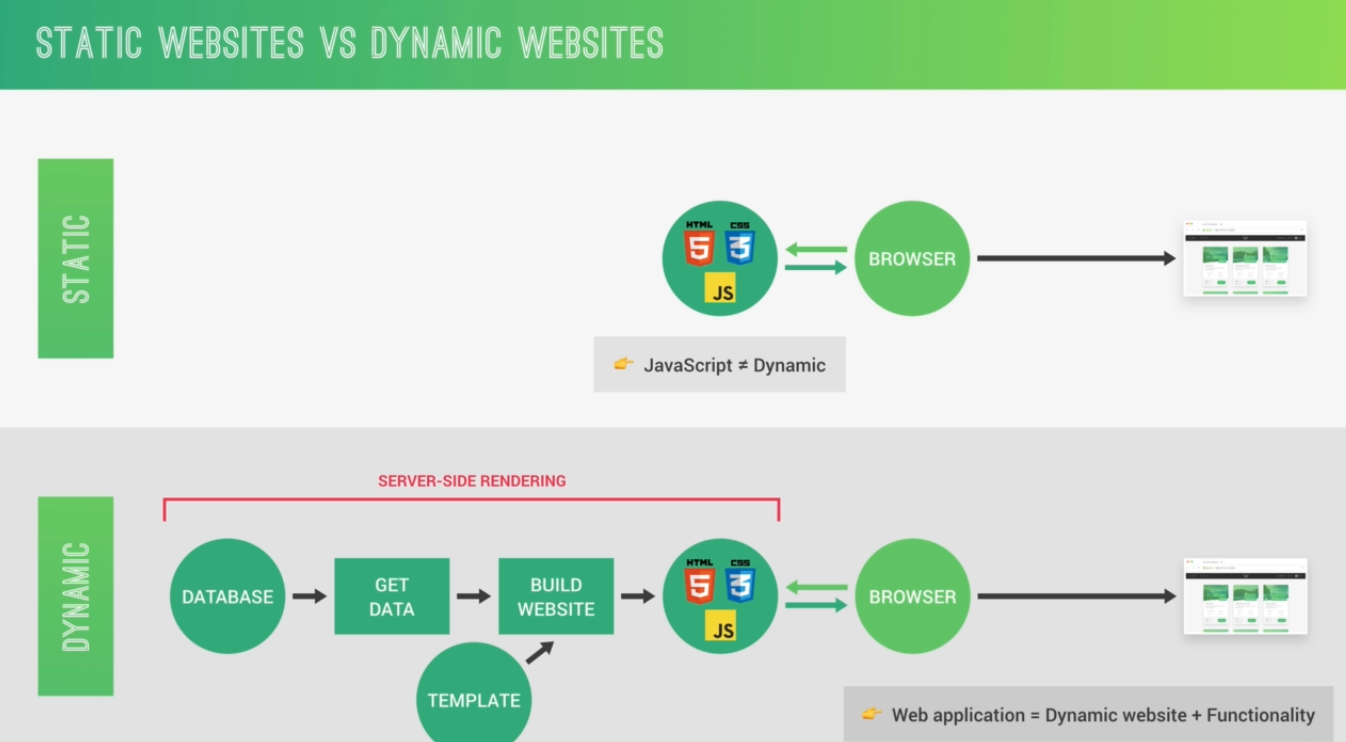
1. Node is a JavaScript runtime built on Google’s open source V8 JavaScript engine.
2. Node.JS pros:
   1. Single-threaded, based on event driven, non-blocking I/O model
   2. Perfect for building fast and scalable data-intensive apps;
   3. Some big companies uses node in production(Netflix, uber, PayPall, ebay…)
   4. JavaScript across the entire stack: faster and more efficient development;
   5. NPM: huge library of open-source packages available for everyone for free
   6. Very active developer community
3. Use Node.JS for:

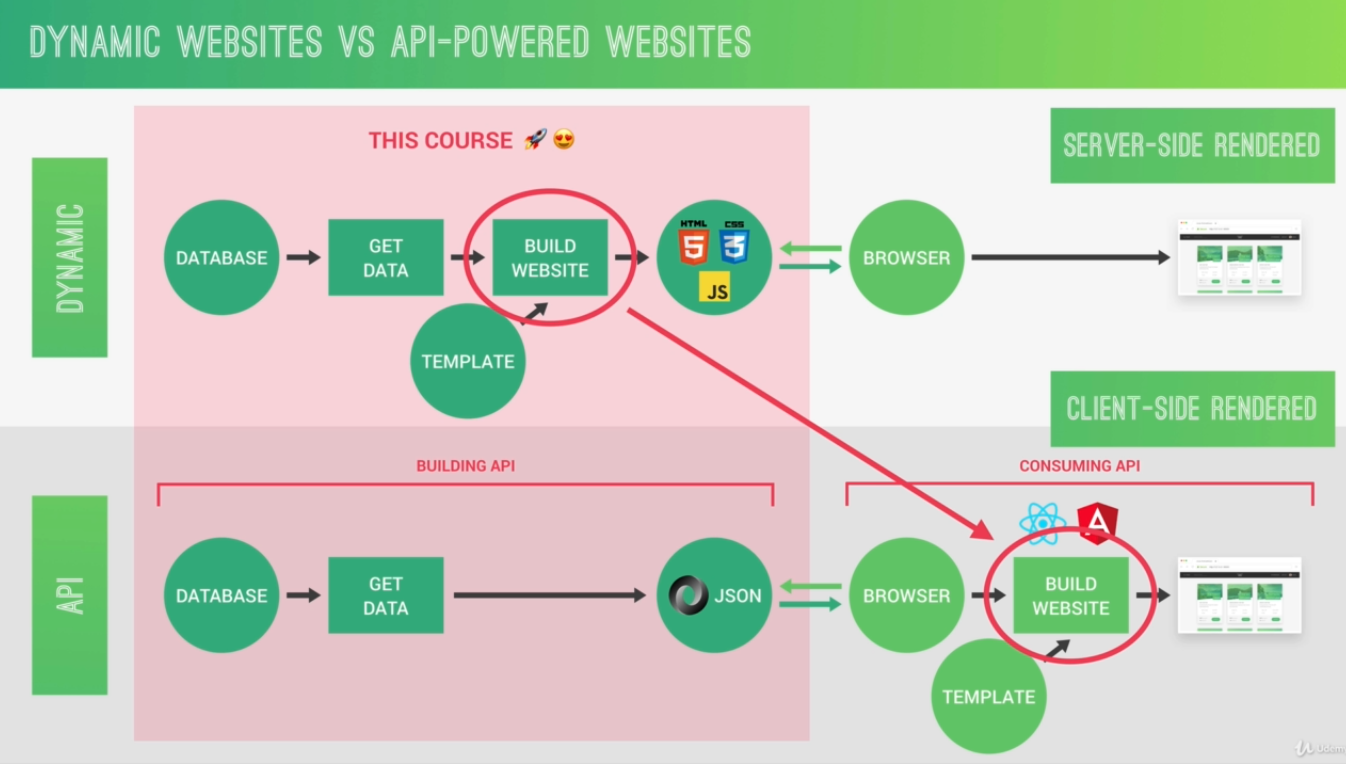
- API with database behind it (preferably NoSQL);

- Data streaming (think Youtube);

- Real-time chat application

-Server-side web application

1. Don’t use Node.JS for the heavy server-side processing (CPU intensive) like image processing, video manipulation, file compressing etc.
2. Node is Single Threaded so if you use synchronous code it means that any other operation needs to wait for the first one to be finished
3. Use Asynchronous functions for executing operations. Using callback function in Node.JS
4. To avoid callback hell we can use Promises or Async/Await
5. Dependencies are of two types: regular dependencies where the code depends on the dependency and the dev-dependencies that help the developers with tools and helpers for the developers(like webpack, nodemon etc.). Nodemon is good for listening for the changes and restarting the server for us instead of using “node index.js”.
6. If we want to use a local script in the terminal we need to declare them into the scripts. We can only use global dependencies directly in the terminal
7. What is a “slug”? It is a part of a URL that identifies a particular page on a website in an easy-to-read form. A slug is the part of the URL that explains the page’s content. (slugify used)
8. Code for dependencies versions. Ex: \*^~1.18.10 (1 is the major version, 18 is the minor version, and the 10 is the patch version). The “~” stands for only patch releases(which is safer), the “^” is for all the patch and minor releases. There is also an option to update to all version using “\*”
9. Util commands : npm outdated, npm update(that doesn’t work updating the package.json, but package-lock is working in npm 6+), npm I package@version, npm update <packageName>, npm -rm -r node\_modules, npm uninstall <packageName>, npm install
10. Request-response architecture
11. Frontend-backend
12. Static vs dynamic websites
13. Dynamic vs Api-powered websites



1. Dynamic websites are computing the data on the serverside and the website is build there based on the actions the users makes. The server makes the pages and send them back to the browser to be displayed. The Api powered website is getting the data from a database, and expose it to the frontend through JSON and the browser gets the data and you can build the website on the frontend alone. You can make an API server and it can exist alone to be consumed by different consumers.
2. Api severs 