

Node.js

TABLE OF CONTENTS

01 > What is Backend

02 > What is http request

03 > Intro to Node.js

04 > Global variables & Modules

05 > Export and Require

06 > Intro to NPM

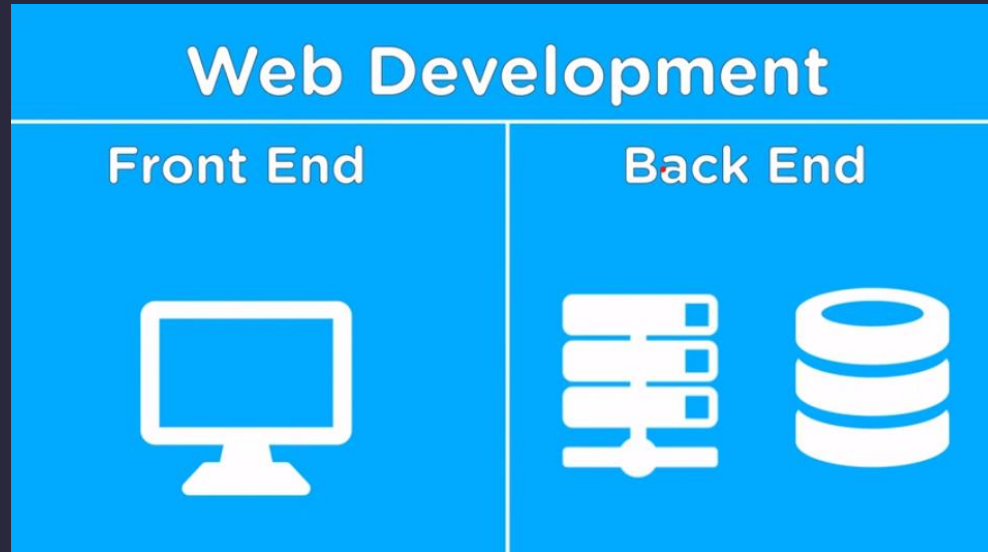
07 > Create a node.js project

01

What is Backend

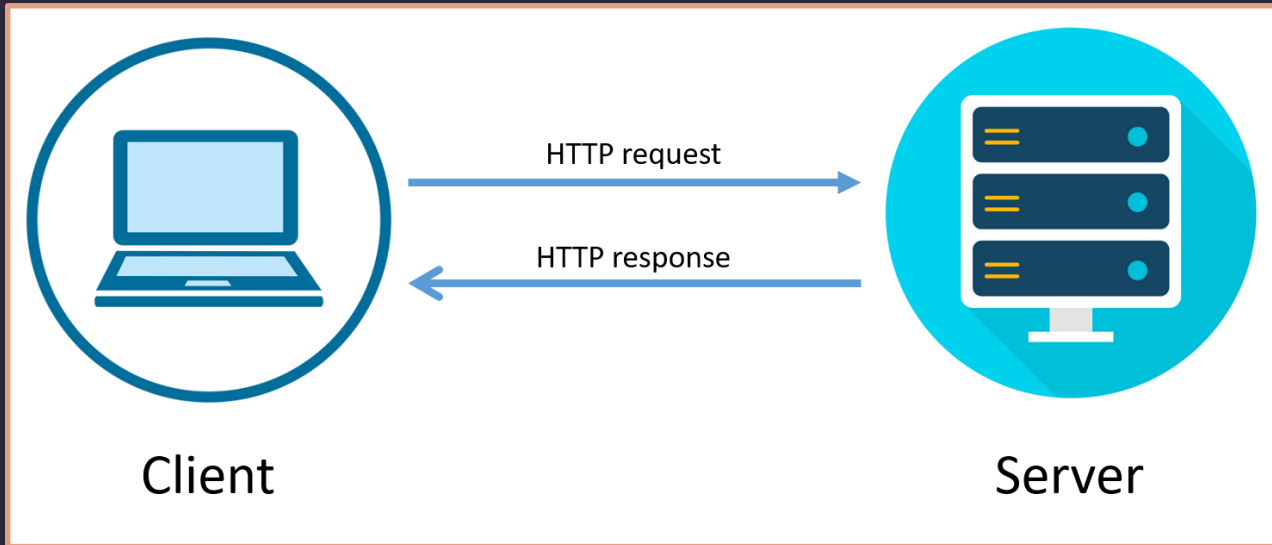
What is Backend?

- Website is made of two main parts



What is Backend?

- Request and Response Cycle



What is Backend?

- AKA Server Side.
- Consists of the Server.
- Saves and manages your data.
- Which provides data on request.
- The Server receives http request from Frontend in the form of URL.
- From the URL, server can get almost all the things it needs in order to process a request.

```
http://example.com/path?query=value
```

02

What is http request

What is HTTP request?

- An HTTP request is made by a client. To a named host.
- The aim of the request is to access a resource on the server.
- To make the request, the client uses components of a URL (Uniform Resource Locator).
- Which includes the information needed to access the resource.

Lets breakdown a typical URL – HTTP

- HTTP stands for Hyper Text Transfer Protocol.
- Request-response protocol for the web for information sharing.

`http://example.com/path?query=value`

Protocol

HTTP

Non-Encrypted

HTTPS

Encrypted

Lets breakdown a typical URL – Host

- AKA domain name.
- It tells the internet which server needs to send the response to.

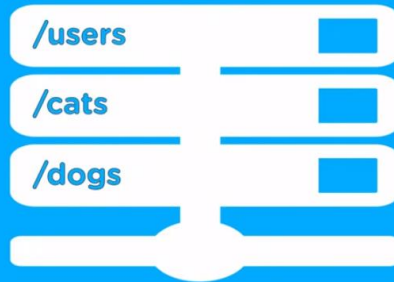


Lets breakdown a typical URL – Path

- Tells the server what the client wants.
- Tells which section of the code on the server should be run in order to get the correct response.

`http://example.com/path?query=value`

Path



Lets breakdown a typical URL – Query string

- Is used by the specific section of the server to alter the response.
- The query string is broken down into specific query parameters.
- Which can augment the way server response to a request for a specific path.

`http://example.com/path?query=value`

Query String

Key	Value
query	value
search	dogs
fullscreen	false

03

What is Node.js



Introduction to Node.js

- Node.js is an open source.
- Server environment.
- It is actually neither a framework or a library - as with traditional application software, but a runtime environment.
- A runtime environment (RTE) contains Web API's that a developer can access.

04

What are Global variables and Modules



Global Variables in Node.js

- 1) `__dirname` - path to current directory.
- 2) `__filename` - file name.
- 3) `require` - function to use module (CommonJS)
- 4) `module` - info about current module (file)
- 5) `process` - info about .env where the program is being executed.
- 6) There are more modules....



Modules in Node.js

- Own Modules:
 - Every file in node.js project that a developer adds is referred as **module**.
 - Many of the modules are exported from their files so they can be used in other project files.
- Node built in Modules:
 - OS - Provides information about computer's operating system.
 - PATH - Provides a way of working with directories and file paths.
 - FS - Allows you to work with the file system on your computer.
 - HTTP - Allows Node.js to transfer data over the HTTP.
 - There are more modules..

05

What are Export and require



Export and Require

- **Export:**
 - Is used when you want to make something available somewhere else in your code.
 - You can export array, object, function.
 - `Module.exports.items = ['item1', 'tem2']`
- **Require:**
 - Is used to use something that is available to use from the outside of the file.
 - `require('./filename')`

06

Intro to NPM



Intro to NPM

- Utility for interacting with **packages** e.g. package installation, package uninstallation.
- It is an **online repository** for the publishing of open-source Node.js projects.
- Important commands:
 - npm - global
 - npm -- version.
 - local dependency - to install something only for this particular project.
 - npm i <packageName>
 - npm install -g <packageName>



Package.json

- Manifest file (stores important info about project/ package).
- There are **three** ways of creating this file:
 - Manual approach - create package.json in root directory, create properties etc.
 - npm inti - step by step, you can also press enter to skip the options.
 - npm init -y - to set all options default.



Package (version numbers)

- The version of a package is represented in three numbers e.g. 4.12. 18
 - 1st **major** change
 - When this changes that means there are some breaking changes.
 - 2nd **minor** change
 - That means it is backward compatible.
 - 3rd **patch** change
 - For the bug fix.



Dependencies in Node.js

- Project dependencies
 - Dependencies for project e.g. bootstrap
- DevDependencies:
 - Dependencies for development e.g. Nodemon

06

Create a node.js project



Installing Nodemon

- It simply restarts the node application.
- Whenever it observes the changes.
- In the file present in the working directory of your project.



scripts in package.json file

- Change current start script to - **“start”**: **“node app.js”** and run project using npm start.
- Add another command to script -**“dev”**: **“nodemon app.js”** |run project as npm run dev.
- If you want to run nodemon on **npm start** then change the start **“script”** to
 - **“nodemon app.js”**



Setting up Version Control System

- Initialize git and GitHub repo of node project.
- Add .gitignore file:
 - Add node_modules file in .gitignore file.



Let's create server

```
const http = require("http");

const server = http.createServer((req, res) => {
  if (req.url === "/") {
    res.write("Welcome to the home the page");
    res.end();
  }
});

server.listen(5000, () => {
  console.log("Server listening on port 5000...");
});
```

<QnA>

>

Thanks!

<

Regards: Hassan Bilal