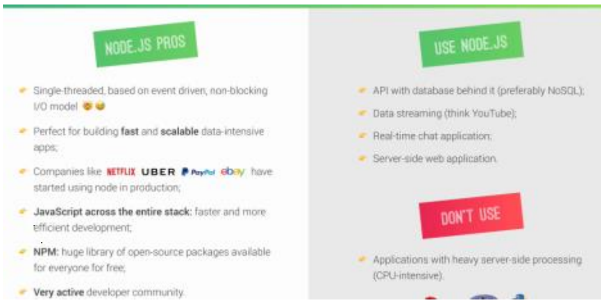
Node

16 February 2022 08:26 AM

हम जानते है कि Html, css and js सिर्फ browser पर run होती है पर हम इसको outside browser पर run करा रहे तो v8 engine आता है जो google ने बनाया है। V8, js को run करने के लिए Environment देता है।




```
D:\nodeProject\testingNode\nodeRevision>node
Welcome to Node.js v14.18.2.
Type ".help" for more information.
>
Array                 ArrayBuffer           Atomics               BigInt
BigInt64Array         BigUint64Array        Boolean               Buffer
DataView              Date                  Error                 EvalError
FinalizationRegistry  Float32Array          Float64Array          Function
Infinity              Int16Array            Int32Array            Int8Array
Intl                  JSON                  Map                   Math
NaN                   Number                Object                Promise
Proxy                 RangeError            ReferenceError        Reflect
RegExp                Set                   SharedArrayBuffer     String
Symbol                SyntaxError           TextDecoder           TextEncoder
TypeError             URIError              URL                   URLSearchParams
Uint16Array           Uint32Array           Uint8Array            Uint8ClampedArra
WeakMap               WeakRef               WeakSet               WebAssembly
_                     _error                assert                async_hooks
buffer                child_process         clearImmediate        clearInterval
clearTimeout          cluster               console               constants
crypto                decodeURI             decodeURIComponent    dgram
diagnostics_channel   dns                   domain                encodeURI
encodeURIComponent    escape                eval                  events
fs                    global                globalThis            http
```

→ Node

→ press 'tab' सब देखने के लिए

```
> String.
String.__proto__            String.hasOwnProperty       String.isPrototypeOf        String.propertyIsEnumerable
String.toLocaleString       String.valueOf

String.apply                String.arguments            String.bind                 String.call
String.caller               String.constructor          String.toString

String.fromCharCode         String.fromCodePoint        String.length               String.name
String.prototype            String.raw

> String.
```

after tab check all string function for string.

```
> Number.
Number.__proto__            Number.hasOwnProperty       Number.isPrototypeOf        Number.propertyIsEnumerable
Number.toLocaleString       Number.valueOf

Number.apply                Number.arguments            Number.bind                 Number.call
Number.caller               Number.constructor          Number.toString

Number.EPSILON              Number.MAX_SAFE_INTEGER     Number.MAX_VALUE            Number.MIN_SAFE_INTEGER
Number.MIN_VALUE            Number.NEGATIVE_INFINITY    Number.NaN                  Number.POSITIVE_INFINITY
Number.isFinite             Number.isInteger            Number.isNaN                Number.isSafeInteger
Number.length               Number.name                 Number.parseFloat           Number.parseInt
Number.prototype
```

→ Similar like Strings

check Numerical functions

```
> Array.
Array.__proto__             Array.hasOwnProperty        Array.isPrototypeOf         Array.propertyIsEnumerable
Array.toLocaleString        Array.valueOf

Array.apply                 Array.arguments             Array.bind                  Array.call
```

```
> Array.
Array.__proto__           Array.hasOwnProperty      Array.isPrototypeOf       Array.propertyIsEnumerable
Array.toLocaleString      Array.valueOf

Array.apply               Array.arguments           Array.bind                Array.call
Array.caller              Array.constructor         Array.toString

Array.from                Array.isArray             Array.length              Array.name
Array.of                  Array.prototype
```

→ Similar to check all function

→ Node → open Repl (REPL)

index.js — 1-node-farm

```
const fs = require('fs');

const hello = 'Hello world';
console.log(hello);
```

→ first Code

```
const fs = require('fs')
console.log(fs.readFileSync('./t.txt'))
```

→ उसके से read करता है

```
<Buffer 68 74 74 70 73 3a 2f 2f 67 69 74 68 75 62 2e 63 6f 6d 2f 6a 6f 6e 61 73 73
 74 6d 61 6e 6e 2f 63 6f 6d 70 6c 65 74 65 2d 6e 6f 64 65 2d ... 10 more bytes>
```

Buffer देता

```
const fs = require('fs')
console.log(fs.readFileSync('./t.txt','utf-8'))
```

→ utf-8 से

```
[nodemon] starting `node index index.js`
https://github.com/jonasschmedtmann/complete-node-bootcamp
```

read text.

```
const textOut = `This is what we know about the avocado: ${textIn}.\nCreated on ${Date.now()}`;
fs.writeFileSync('./txt/output.txt', textOut);
console.log('File written!');
```

→ Write file

→ location जहाँ file create होगी

→ file में जो ~~कि लिखना~~ है।

Const http = require('http')

```
// SERVER
const server = http.createServer((req, res) => {
  console.log(req);
  res.end('Hello from the server!');
});

server.listen(8000, '127.0.0.1', () => {
  console.log('Listening to requests on port 8000');
});
```

→ ये Host है/(localhost)

→ ये port है।

Const http = require('http')
Const url = require('url')

```
// SERVER
const server = http.createServer((req, res) => {
  const pathName = req.url;

  if(pathName === '/' || pathName === '/overview') {
```

→ Routing with url

```
////////////////////////////////////
// SERVER
const server = http.createServer((req, res) => {
  const pathName = req.url;

  if(pathName === '/' || pathName === '/overview') {
    res.end('This is the OVERVIEW');
  } else if (pathName === '/product') {
    res.end('This is the PRODUCT');
  } else {
    res.writeHead(404, {
      'Content-type': 'text/html',
      'my-own-header': 'hello-world'
    });
    res.end('<h1>Page not found!</h1>');
  }
});

server.listen(8000, '127.0.0.1', () => {
```

→ Routing with url

→ Custom header भी डाल सकते है

→ Header हमेशा Response के पहले जाता है

```
// SERVER
const server = http.createServer((req, res) => {
  const pathName = req.url;

  if (pathName === '/' || pathName === '/overview') {
    res.end('This is the OVERVIEW');
  } else if (pathName === '/product') {
    res.end('This is the PRODUCT');
  } else if (pathName === '/api') {

    fs.readFile(`${__dirname}/dev-data/data.json`, 'utf-8', (err, data) => {
      const productData = JSON.parse(data);
      res.writeHead(200, { 'Content-type': 'application/json'});
      res.end(data);
    });

  } else {
    res.writeHead(404, {
```

→ अगर file send करने होते तो

→ read file and send

```
// SERVER
const data = fs.readFileSync(`${__dirname}/dev-data/data.json`, 'utf-8');
const dataObj = JSON.parse(data);

const server = http.createServer((req, res) => {
  const pathName = req.url;

  if (pathName === '/' || pathName === '/overview') {
    res.end('This is the OVERVIEW');
  } else if (pathName === '/product') {
    res.end('This is the PRODUCT');
  } else if (pathName === '/api') {
    res.writeHead(200, { 'Content-type': 'application/json'});
    res.end(data);
  } else {
```

→ इन दोनों मे हम file send कर रहे है। पर ऊपर वाले मे बार-बार Read होगा फिर send होगा और इसमे Read एक बार होगा send जितने बार need होगी उतनी बार होगा

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                         1: node
jonas.io  1-node-farm  npm install nodemon --save-dev
            ) ⠼ finalize:minimatch: sill finalize /Users/jonas.io/Desktop/1-node-farm/node_modules/minimatch
```

→ install development dependenc

```
>npm i nodemon --global
mon -> C:\Users\17ics\AppData
```

→ install as global

```
ies": {
"^1.18.11"
```

→ ye bug fixed show करता है

→ feature done करता है (minor)

→ (major) → पूरा change कर दिया

```
npm outdated
npm install slugify@1.0.0
oadExtraneous: sill removeObsoleteDep
```

→ install Specific Version @no

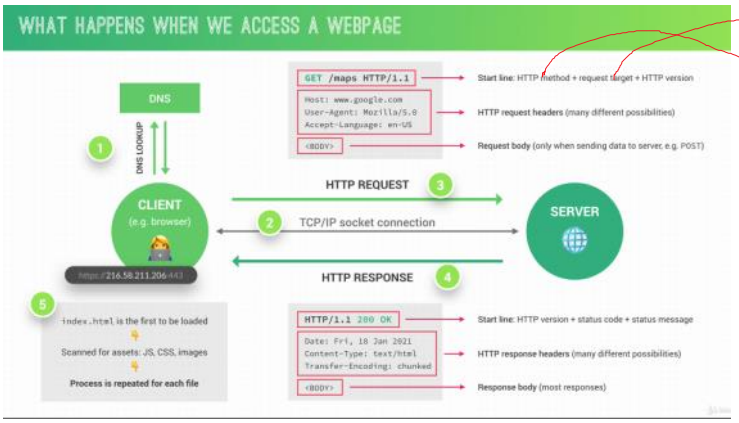
ies

```
>npm outdated
```

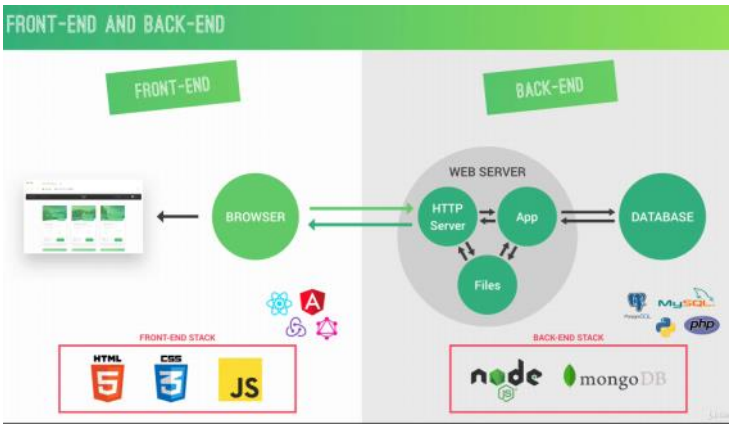
→ check outdated dependencies

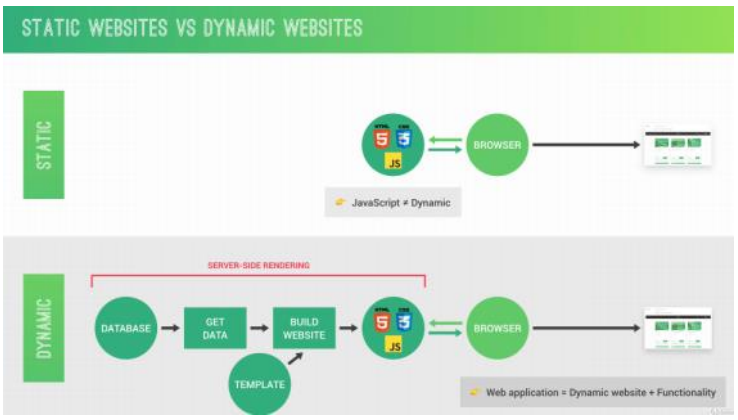
```
TERMINAL
npm update slugify
0 No repository field.

udited 2237 packages in 1.291s
```

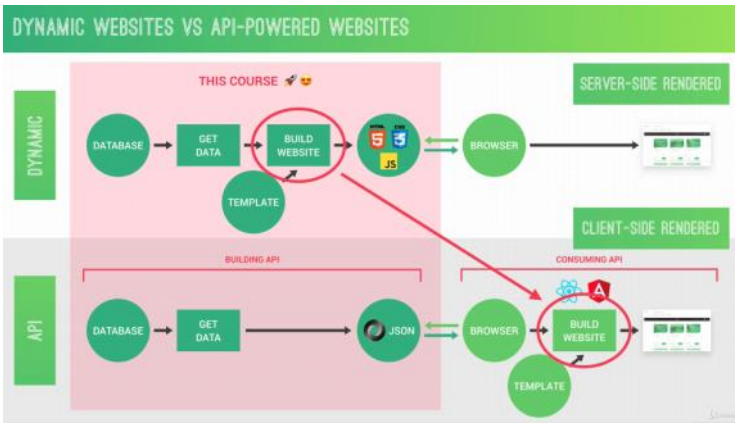
→ package update

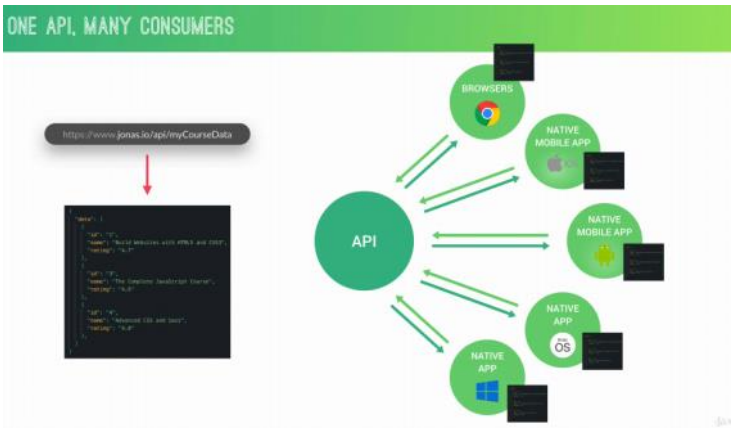


→ google.com/map → Resource.
→ Get / Post / put






DYNAMIC WEBSITES VS API-POWERED WEBSITES
THIS COURSE
SERVER-SIDE RENDERED
DYNAMIC
DATABASE
GET DATA
BUILD WEBSITE
TEMPLATE
BROWSER
CLIENT-SIDE RENDERED
BUILDING API
CONSUMING API
API
DATABASE
GET DATA
JSON
BROWSER
BUILD WEBSITE
TEMPLATE


ONE API, MANY CONSUMERS
https://www.jonas.io/api/myCourseData
API
BROWSERS
NATIVE MOBILE APP
NATIVE MOBILE APP
NATIVE APP
NATIVE APP