Run time to run js outside the browser

Es6 🡪 callbacks, promises, aync-await

Global in node :   
 \_\_dirname 🡪 path to current directory

\_\_filename 🡪 file name

module 🡪 info about current module

require 🡪 function to use modules

process 🡪 info about env where program being executed

Build in modules:

os

fs

path

http

fs : readFileSync, writeFileSync, readFile, writeFile

Asynchronous is task will start simultaneously

Alternatives for above methods are promises ,async and await

http:

to create server and do server operations.

Npm : (Node Package Manager)

Install : npm install will install node\_modules folder

Installing packages: npm install nodemon or npm I nodemon

Installing globally : npm install -g nodemon

Event Loop : The event loop is what allows Node.js to perform non-blocking I/O operations — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.

Asynchronous methods :

setTimeOut():

setTimeout(() =>{

    console.log("second (3) one");

},0)

setInterval()

Async patterns :

Promise(),async(),await()

Event-Driven :

Require(‘events’)

Streams :

Writable

Readable

Duplex

Transform

Express :

MIME types

Error Codes

Get,post,delete,put,listen,all,use

Cool thing in express is **express.static(<folder path>)**

This will take all the style files (js,css,images,html..) and add to the application just we need to give folder path where all the file are. We don’t need to configure explicitly.

API vs SSR

API 🡪 setting an http interface to interact with data , data is sent as json(javascript object notation) to send response we use res.json() method.

SSR 🡪setting templates, send template, res.render()

req.params

<https://hn.algolia.com/api>

logger – middleware?

Morgan 🡪 app.use(morgan('tiny'))

Gatsby

Package-lock.json --🡪 requirement?

Great community

Package.json will have dependencies details

# MongoDB(NoSQL)

mongodb://localhost:27017

SQL vs NoSQL

In SQL table called as collection in mongo db

Row/Record 🡪 Document

Column 🡪 Field

We can store duplicate data so we don’t need to use relations.

In SQL Data uses schemas, Relations 🡪 NoSQL is schemeless, very few relations

Data is distributed across multiple tables 🡪 Data is typically merged/ nested in a few collections

Horizantal scalling is difficult/impossible,Vertical scalling is posible 🡪 Both horizantal and Vertical scalling is posible

Limitations for lots of (thousands) read & write queries per second 🡪 Great perfarmance for mass read and write requests

### SQL :

To work with mysql install mysql2 package(npm install --save mysql2)

### Sequelize:

It is like ORM (Object Relational Mapping Library) for NodeJS.

To work with sequelize install sequelize package (npm install --save sequelize) and it will need mysql2 package too.

Usage :

1. Install the package   
    npm install --save sequelize mysql2
2. Import the sequelize package  
    const Sequelize = require('sequelize');
3. Create instance

venkatesh@1993

**venkateshv1993 🡪 postman**

**venkateshlucky 🡪 git || venkatesh@1993**