Node Js

Moshge tutorial eka

Node eka start karanne node filename.js

//browser(client side) eke thiyenne window nodejs wala global

//global.console.log("global is like window");

//module concept

//1.every node project atleast one module

//2.module encapsulate things

//3.anything add to export will accessible through other modules

//console.log(module);

//add module to  this module

//require('./filename') this folder

//require('./filename') parent folder

//require('./subfolder/filename')

/\*

function godak nam object ekak widihata

danna module.export ekata

mehema

 module.export.objectName=functionName;

 only one function nam kelinma assign ok

module.export=functionName;

me paththe use wenne object name eka hodama wade functionName == objectName karana eka goodpractise ekak

 \*/

//const logger = require("./logger").default;

//logger.log("ishan");

/\*

java wala psvm wage meke module wrapper function ekak thiyenawa

(function (exports, require, module, \_\_filename, \_\_dirname) { code tika })()

meka An immediately invoked function expression (or IIFE, pronounced "iffy")

meka aththada kiyala balamu

\*/

console.log(\_\_filename);

console.log(\_\_dirname);

/\*

built in modules use karanakota ../ or ./

ganne nathuwa normal require('path')

wage dunnoth eka built in module kiyla node aduragannawa \*/

/\*

path module eka path gana info

dena module ekak \*/

const path = require("path");

var pathobj = path.parse(\_\_filename);

console.log(path);

console.log(pathobj);

/\*

os module eka os eka gana info usefull

ram eka, uptime wage dewal \*/

const os = require("os");

console.log(os.totalmem() / 1024 \*\* 3);

console.log(os.freemem());

/\*

file system gana thiyena module eka

thama fs module eka

eken files manipulation karanna puluwan \*/

const fs = require("fs");

const files = fs.readdirSync("./");

console.log(files);

fs.readdir("./", function(error, files) {

  error ? console.log(`error ${error}`) : console.log(files);

});

/\*

event class eka

java wala  event listner ekata samanai \*/

/\* //Register a listner

emitter.on("messageLogged", arg => {

  console.log("listner is called", arg);

});

//Raise an event

emitter.emit("messageLogged", { id: 12313 });

 \*/

//must register before raise

const Logger = require("./logger");

const logs = new Logger();

//REgister listner

logs.on("messageLogged", e => {

  console.log("listner called", e);

});

//after register emit by log method

logs.log("messageLogged from practise");

//emit eken message ekak emit karanawa

// eee emit karapu message register karapu

//listner allagannawa/\*

/\*

ඔන් මෙතඩ්  එකෙන්  කියනවා එයාගේ   එකට  ඔන්න  ලිස්ට්න්  කිරීම පටන්ගන්න කියලා

එමිට්  එකට් කෝල් එකක්  එකක් ආවම ඔන් එකෙ ලිස්ට්නර්ව කෝල් වෙනවා

ඔන් and එමිට් method names සමානවෙන්න ඕනේ

.on("messageLogged"

 this.emit("messageLogged"

\*/

Logger class eka thiyena module eka

var url = "http://mylogger.io/log";

const EventEmitter = require("events");

class Logger extends EventEmitter {

  log(message) {

    //send an gttp request

    console.log(message + "received by logger");

    //Raise an event event ekak emit karanna balaporoththuwen inna kenek meya

    this.emit("messageLogged", { id: 123123 });

  }

}

module.exports = Logger;

///////////////////////////////////

//http module important ma module eka

const http = require("http");

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

  if (req.url === "/") {

    res.write("Hellow world");

    res.end();

  }

  if (req.url === "/api/courses") {

    res.write(JSON.stringify([1, 2, 3, 4, 5]));

    res.end();

  }

});

server.on("connection", socket => {

  console.log("new connection");

});

server.listen(3000);

console.log("listning to port");

**# This kiyala haduwoth node wala method**

**Awul yanawa**

app.get("/ass", analyzeThis);

function analyzeThis(request, response) {

  response.send("data");

}

//This wenuwata this dammoth okkk

**# save node pakage globaly**

**npm install -g body-parser**

**#link already installed ones**

If you're using npm >=1.0, you can use npm link <global-package> to create a local link to a package already installed globally. (Caveat: The OS must support symlinks.)

However, this doesn't come without its problems.

npm link is a development tool. It's awesome for managing packages on your local development box. But deploying with npm link is basically asking for problems, since it makes it super easy to update things without realizing it.

As an alternative, you can install the packages locally as well as globally.

For additional information, see

**Traversy ge tutorial eka**

**Changes in chrome and node**



