**Simple NodeJS server**

NodeJS run a single JS file. Which is the main entry point of a NodeJS server. This file mainly we can use index.js or server.js. this is convention nothing else.

We will pass the index.js file to the node. And node will run the file. Like: node index.js

No need to write index.js. no need to provide extension. By default node keep the extension .js

**Basic Node Application Structure:**

/\*

 \* Title: Basic Node app example

 \* Description: Simple node application that print random  quotes per second interval

 \* Author: Masud Rana

 \* Date: 23/10/2023

 \*/

// Dependencies [own module, external module, node module]

const mathLibrary = require("./lib/math");

const quotesLibrary = require("./lib/quotes");

// App object - Module scaffolding

const app = {};

// Configuration

app.config = {

  timeBetweenQuotes: 1000,

};

// Function that prints a random quote

app.printAQuote = function printQuote() {

  // get all the quotes

  const allQuotes = quotesLibrary.allQuotes();

  // get the length of all the quotes

  const numberOfQuotes = allQuotes.length;

  // pick a random number between 0 and numberOfQuotes

  const randomNumber = mathLibrary.getRandomNumber(1, numberOfQuotes);

  // get the at that position in the array (minus one)

  const selectQuote = allQuotes[randomNumber - 1];

  // print the quote to the console

  console.log(selectQuote);

};

// function that loops indefinitely, calling the printAQuote function as it goes

app.indefiniteLoop = function indefiniteLoop() {

  setInterval(app.printAQuote, app.config.timeBetweenQuotes);

};

// Invoke the loop

app.indefiniteLoop();

**Note:** we should not reassign function parameter. Better we declare variable and then assign function parameter to this variable and work with this variable.

* Always use scaffolding object and export this object. So that we can easily access one object and export all the necessary file to the one object file

That is the example of good code, beautiful code, readable code.

Best practice, structured practice