Node.js

Node.js is an asynchronous event driven server-side Javascript runtime environment. It was developed by Ryan Dahl in 2009. Node was designed for building scalable network dependent applications. A rich library of JavaScript modules are also provided for this platform that simplifies the development of web applications.

Features:

Node.js is asynchronous such that when multiple requests come in it handles them concurrently. Moreover, when a request has been made, Node will move on to the next request even when the previous has not been finished yet. Node will send back data in chunks rather than buffering.

Node is event driven because as soon as its server starts running, it initiates all variables, declares functions and waits for an event to occur. Node.js’ Event Loop listens for events and triggers a callback function.

Where to Use Node.js

* I/O bound applications
* Data Intensive Real-Time Applications
* JSON APIs based applications
* Data Streaming applications
* Single Page Applications

Environment Setup

If you are willing to set up your development environment for Node.js then you need a text editor, to develop your own Node.js applications, and the Node.js binary installable.

The Node.js installable can be downloaded from the Official website of Node.js

<https://nodejs.org/en/>

Run the msi file



or



Run the msi files and it will automatically include the Node.js to your machine’s PATH environment variable.

Components of a Node.js application

An application developed with Node.js consists of 3 important components:

1. Modules – modules are imported for use in your source code for easier and faster coding. Modules are loaded by using the require directive.
2. Server – the server will be the one listening to the requests that clients will make.
3. Request and Response mechanism – when the server listens to a client’s request it generates a response as defined by the programmer in reply to the request.

Sample Code

const module = require(‘http’);

const server = module.createServer();

const host = ‘localhost’;

const port = 8000;

server.on(‘request’, requestHandler);

server.listen(port, host, ()=> {

console.log(‘The server is running!’);

}

)

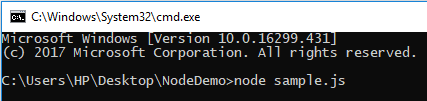
function responseHandler(request, response){

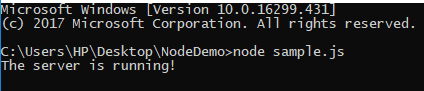
console.log(‘You have responded to the request’);

response.end();

}

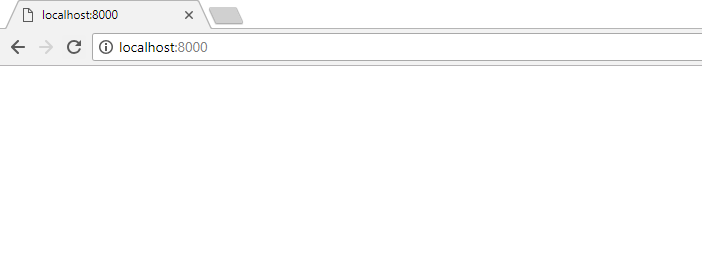
To start the server, go to the directory of your file then run your terminal there. Issue the command :





Making a Request to the Server

In any browser, open localhost:8000 and observe following result:





Node.js – NPM

The Node Package Manager (NPM) provides 2 main functionalities:

* Node.js packages and modules available via online repositories
* Utility to install packages using a machines command line, also it can do version management and Node.js packages’ mdependency management.

Installing a Module using NPM

To install a module in a command line interface issue the command



Example of a web framework module is the express module.



After installation you can already use the module by including this to your .js file:



Uninstalling a module

To uninstall a Module just issue the following command:



To verify if you have successfully uninstalled a module you can issue the command:



Updating a module

To update a module