Node JS

Contents

[1 Node JS Introduction 3](#_Toc425935169)

[2 Environment Setup 3](#_Toc425935170)

[3 NPM (Package manager) 3](#_Toc425935171)

[4 Callback concept 3](#_Toc425935172)

[5 Event loop 3](#_Toc425935173)

# Node JS Introduction

Node JS is a JavaScript based framework built on Google’s JavaScript version 8 engine. It is used to develop intensive I/O application e.g. video streaming, single page application like web applications.

Node JS is open source and used widely in the market.

Node JS provides various JavaScript modules which eases the web development.

# Features of Node JS

Following are the features of the Node JS

* **Asynchronous and event driven**

All APIs of Node JS are asynchronous i.e. non-blocking. It essentially means that Node JS based server never waits for an API to return data. Server moves to the next API after calling it and a notification mechanism of Node JS helps to get the response from the previous API call.

* **Very Fast**

**Node JS library is very fast in code execution.**

* **Single threaded but highly scalable.**

**Node JS uses a single thread model with event looping. Event mechanism helps server to respond in a non-blocking ways and make server highly scalable as opposed to traditional servers which creates limited number of threads to handle request.**

**Node JS use single threaded program and same program serves number of requests than traditional server.**

# Environment Setup

To setup Node JS environment, you require following two things.

* Text editor
* Node Js installable

Install the Node Js according to the operating system. We are the setting up the Node Js for Windows 7, so we will follow all the Node Js commands for the Windows 7 OS through out the document.

To check Node Js is setup on machine or not; simply run on command prompt Node --version command, you will get the version.

Before creating actual "Hello, World!" application using Node.js, let us see the parts of a Node.js application. A Node.js application consists of following three important parts:

* **Import Require Module:** We use **require** directive to load Node Js modules.
* **Create Server:** A server which will listen to client requests.
* **Read Request and Return Response:** Server of Node Js will response to the request created by the web application or console applications.

# Creating Node Js Application

## Step 1 Import Required Module

We use require directive to load http module and store returned HTTP instance into http variable as follows:



## Step 2 Create Server

At next step we use created http instance and call **http.createServer ()** method to create server instance and then we bind it at port 8081 using listen method associated with server instance

Pass it a function with parameters request and response. Write the sample implementation to always return "Hello World".

Code is enough to create an HTTP server which listens i.e. wait for a request over 8081 port on local machine.

## Step 3 Testing Request and Response

Let’s combine step 1 and 2 into file **main.js**

**Main.js**



Now execute the main.js, in command prompt you can see the log **Server running at http://127.0.0.1:8081**

# Node Js REPL Terminal

REPL stands for Read, Eval, Print and Loop. It represents computer environment like console in windows or shell in Linux/Unix where command is entered and system responds with the output in interactive mode.

Node is bundled with REPL environment and it performs the following desired task

* Read

Reads the user input, parse the input in JavaScript’s data structure and stores in memory.

* Eval

Takes and evaluates data structure.

* Print

Prints the result

* Loop

Loop the above command until user types Ctrl+c twice.

## Starting REPL

# NPM (Package manager)

# Callback concept

# Event loop