

# Node.js

# Understand Client & Server

**Client**

asks for  
services

Request



**Server**

performs  
services



Response

*A client sends requests to server (Any server), which then sends responses back to the client.*

**Browser**  
asks for  
services

Request



**Web  
Server**  
performs  
services



Response

A browser is an *HTTP client* because it sends requests to an *HTTP server* (Web server), which then sends responses back to the client.

# Browser

asks for  
services

HTTP  
Request



# Web Server

performs  
services

Response

HTTP



HTML/CSS

Bootstrap

JavaScript

ReactJS

AngularJS

jQuery

# Browser

asks for  
services

HTTP  
Request



# Web Server

performs  
services

Response



HTTP

HTML/CSS

JavaScript

Angular

Node.js

expressjs

mongoose

MongoDB

# What is Node.JS?

# Node.js

**Node.js** is an open-source, cross-platform runtime environment for running server-side Web applications.

It is C++ program built on top of **Google's Chrome V8** engine.

**V8** is Google's open source high-performance **JavaScript engine**, written in C++ and used in **Google Chrome**, the open source browser from Google.

Ryan Dahl, Developer from Joyent, initially created Node.js on top of Google V8 capabilities.

Try on your machine, open command prompt, type

```
>node --version
```



# Node Core Architecture

JavaScript

**Node Standard Library**

C/C++

**Node Bindings**  
(Socket, http, etc)

**V8**

**Thread Pool**  
(libeio)

**Event Loop**  
(libev)

**DNS**  
(c-ares)

**Crypto**  
(OpenSSL)

# Node.js

**npm** - Node Package Manager, a tool to install third party developed javascript modules in to your project.

[npmjs.com](https://npmjs.com) maintains registry of several thousands of very useful javascript modules developed by developers elsewhere can be readily used in your own javascript projects

Try on your machine, open command prompt, type

```
>npm --version
```

# Node.js

## npm init

This command initializes Node application project

Try on your machine, open webstorm and create your development folder, open terminal, type

```
>npm init
```

npm init command initializes node app and store preferences in **package.json**. New **package.json** file will be generated by npm init

# Node.js

**npm install <module-name>**

npm install command downloads the module and place them under **node-modules** folder in your application directory

```
>npm install express
```

Let us add express module to our application. You can add any third party module to your project using "npm install <module-name>"

Now, in your code you can start using module by using import (es6) or require(es5)

# index.js

```
//index.js

let firstName = "Varma";
let lastName = "Bhupatiraju";

console.log(`Hello ${firstName} ${lastName}, WELCOME to Node World`);
```

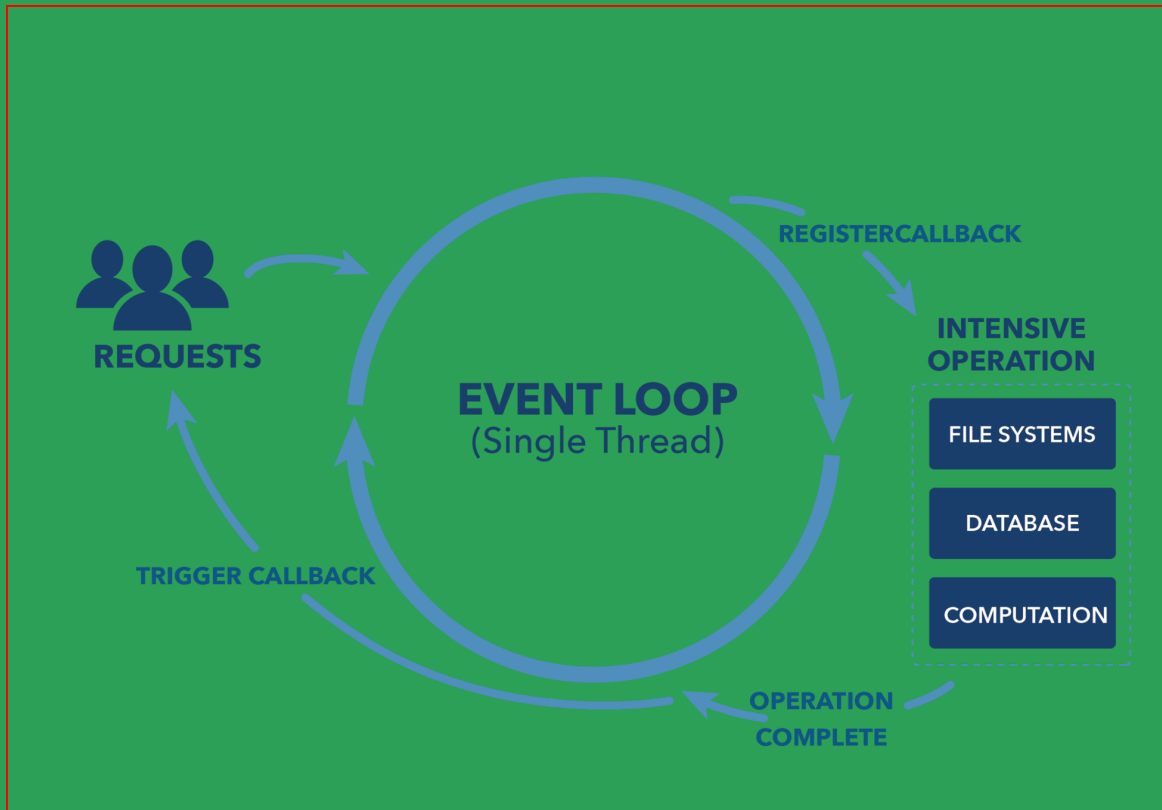
```
>npm start
```

```
>node index.js
```

# Why Node.js?

- Non-Blocking I/O
- Super fast (V8)
- Vibrant Eco System (npm)
- Reuse code between browser and server

# Non Blocking I/O



# How Node can be used?

- Simple Command Line Program
- HTTP Server (Running Website)
- HTTP Server (Running REST API)
- HTTP Server w/ Sockets (As Chat Server)



# Simple Command Line Program

```
//find.js - let's say we want to find a specific file inside a directory.

var fs = require('fs');

var search = function(dir, needle) {
  if(!fs.existsSync(dir)) {
    return console.log('Directory ' + dir + ' does not exist.');
```

```
  }

  var haystack = fs.readdirSync(dir), path, stats;
  for(var s = 0; s < haystack.length; s++) {
    path = dir + '/' + haystack[s];
    stats = fs.statSync(path);

    if(stats.isDirectory()) {
      search(path, needle);
    } else if(path.indexOf(needle) >= 0) {
      console.log(path);
    }
  }
};

search(process.argv[2], process.argv[3]);
```

```
>node find.js ~/projects dino/app.js
```

# HTTP Server w/ Website

server.js

```
fs = require("fs");
host = "127.0.0.1";
port = 1337;
express = require("express");

app = express();

// serve static files in ROOT/public folder
app.use(express.static(__dirname + "/public"));

app.get("/", function(request, response){ //root dir
  response.send("Hello!!");
});

app.listen(port, host);
```

server.js

# HTTP Server w/ API

server.js

```
fs = require("fs");
host = "127.0.0.1";
port = 1337;
express = require("express");

app = express();

// serve static files in ROOT/public folder
app.use(express.static(__dirname + "/public"));

app.get("/", function(request, response){ //root dir
  response.send("Hello!!");
});

app.listen(port, host);
```

server.js