Introduction to Node.js

@MaunashJani

JavaScript

- JavaScript is one of the three core technologies of the World Wide Web => HTML + CSS + JS
- JavaScript is the programming language that runs in your browser.
- You can use it to add interactivity and other dynamic features to your website or application.
- All major web browsers have JavaScript engine to execute it.
- With the advent of Node.js, you can also run JavaScript on the server.

JavaScript Engines

THE GOAL OF A JAVASCRIPT IS

"TO GENERATE THE MOST OPTIMIZED CODE IN THE SHORTEST POSSIBLE TIME."

JavaScript Engines

- **V8**—open source, developed by Google, written in C++ powers Google Chrome
- **SpiderMonkey**—the first JavaScript engine, today powers Firefox
- **JavaScriptCore**—open source, marketed as Nitro and developed by Apple for Safari
- **Chakra** —Microsoft Edge
- **JerryScript**—is a lightweight engine for the IoT.
- And few more…

Node.js

- Created by Ryan Dahl in May 27, 2009.
- **Node.js® is a JavaScript runtime** built on Chrome's V8 JavaScript engine.
- **Server-side JavaScript platform**, which allows you to run JavaScript programs, without the browser.
- **I/O and system works** on the server.
- For extensibility, node follows **CommonJS** standard so that modules can be shared between other JavaScript platforms.
- It even allows developer to write **add-ons** for performance critical component.

Server-Side JavaScript (SSJS)

- One of the features that attract developers is that it allows you to use one language, JavaScript, from browser to backend.
- Frontend developer can easily get used to node without much learning curve since the API of node is designed to be familiar to client-side JS programmers.
- For backend developer it:
  - Generate dynamic page content
  - Can create, open, read, write, delete, and close files on the server
  - Can collect form data
  - Can add, delete, modify data in your database

What Can You Do With Node.js

1. Web Applications
   - Real-time applications
   - Event-based web sites
2. REST API
3. Mobile Apps
   - iOS - **NodObjC**
   - Android - **Anode**
4. Desktop Applications
   - Linux/Windows/OS X - **node-webkit**

Who Uses Node.js

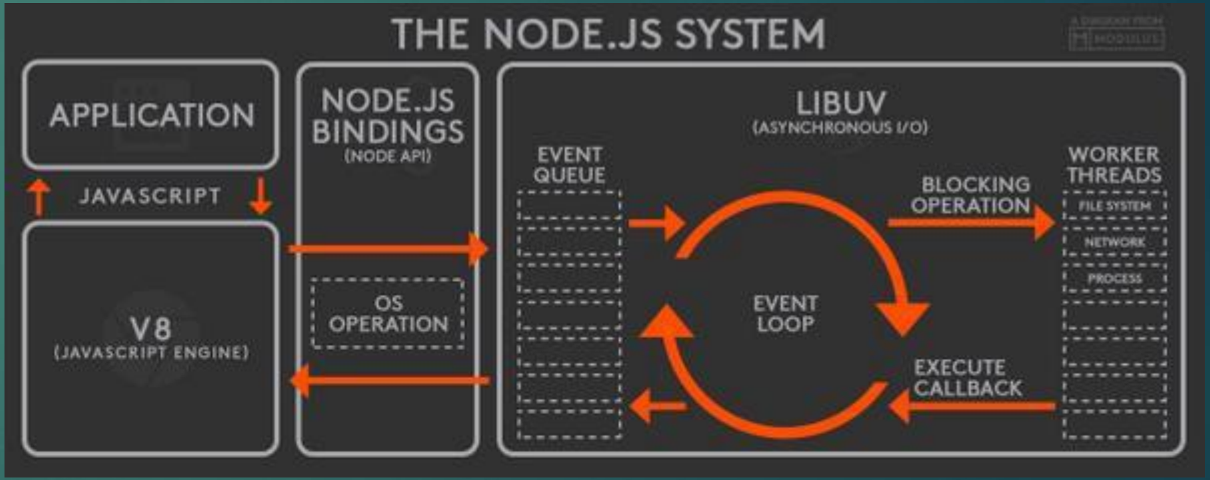
- Netflix
- eBay
- LinkedIn
- Microsoft
- Yahoo
- Walmart
- Uber
- PayPal
- NASA
- Groupon

Design Goals

- No function should direct perform I/O.
- To receive info from disk, network, or another process there must be a callback.
- Stream everything; never force the buffering of data.
- Have built-in support for the most important protocols: TCP, DNS, HTTP
- Single Thread:
  - Node.js is single thread; all applications run on a single thread and it never spawns on other threads.
  - Developers don't need to deal with concurrency, cross-thread operations, variable locking, and so on.

Node.js Architecture

- Node.js follows Single Threaded with Event Loop Model.
- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
- Node.js comprises of two main component **core & its modules.**



Source: Google Images

@MaunashJani

- As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications.
- Code like this

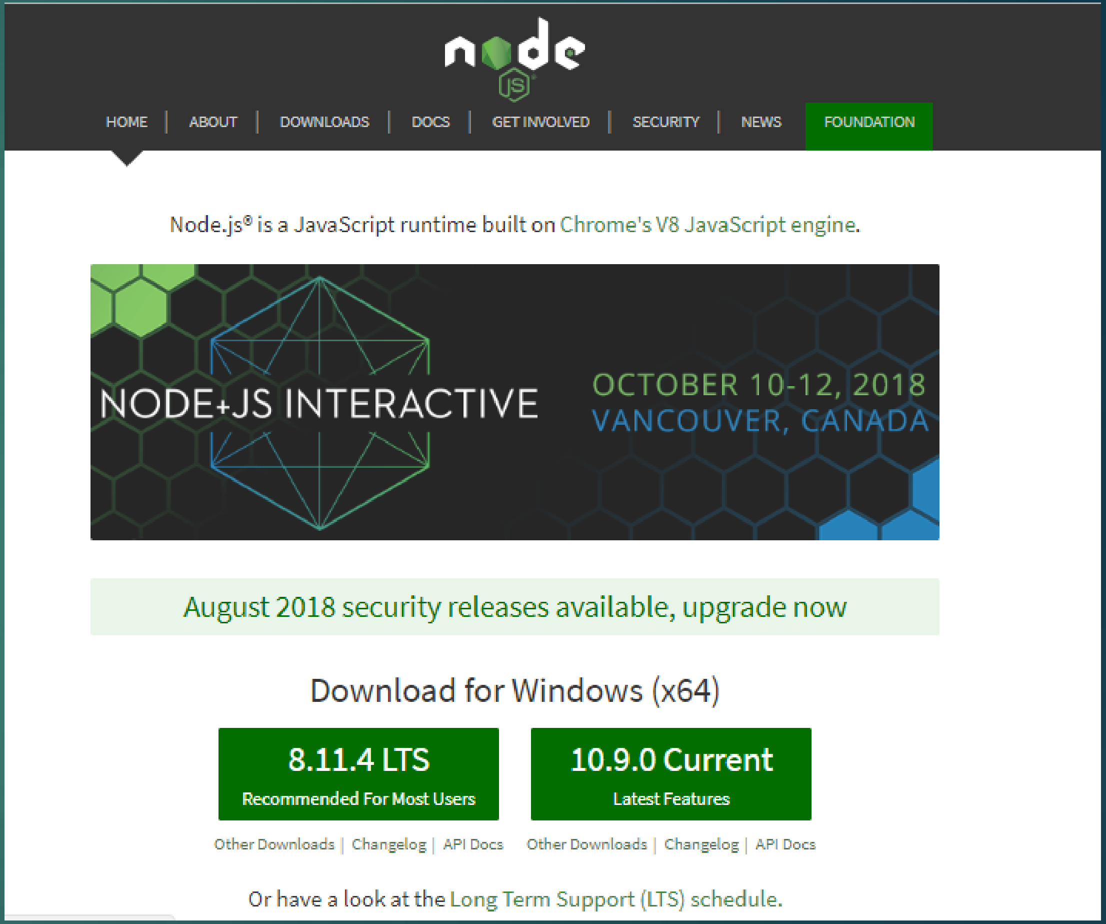
```
var result = db.query("select..");
//result either blocks the entire  process multiple execution stacks.
```

- But a line of code like this allows the program to return to the event loop immediately.

```
db.query("select..", function (result) {
        // use result
});
```

Get Started

- Download & Install - https://nodejs.org
- Text Editor – Visual Studio Code


HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

NODE+JS INTERACTIVE
OCTOBER 10-12, 2018
VANCOUVER, CANADA

August 2018 security releases available, upgrade now

Download for Windows (x64)

8.11.4 LTS
Recommended For Most Users
10.9.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

Source: https://nodejs.org

@MaunashJani

Basic Commands

- node
  - Starts node console
  - node <.js filename>
  - node –v
  - node --help
  - CTRL + C – End node console
  - CTRL + L – Clear node console
  - Node comes with a REPL that is accessible by running Node from the command line.

Hello World

- The basic hello world application in Node.js is something like this:
- > node
- > console.log("Hello World");
- Output:

```
Hello World
Undefined
```

- Writing the line in a file named index.js the execution will be:
- > node index.js
- Output:

```
Hello World
```

Node.js file

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js“
- Node loads the code and after compiling it, Node executes it from top to bottom, registering the callbacks as needed.
- The script has access to various global objects that are useful for writing our applications. Some of them are:
  - __dirname, __filename, console, module, require()

What is a Module in Node.js?

- Node.js uses a module architecture to make simpler the creation of complex applications.
- Modules are like to libraries in C.
- Each module contains a set of functions related to the "subject" of the module.
- For example, the *http* module contains functions specific to HTTP.
- Node.js has a set of built-in modules which you can use without any further installation. http, https, url, fs, events, etc.

Include Modules

- To include a module, use the require() function with the name of the module:
- var http = require('http’);
- Now the application has access to the HTTP module.
-
```
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.end('Hello World!');
}).listen(3000);
```

Create Your Own Modules

- A module that returns the current date and time:

-
  ```
  exports.myDateTime = function () {
      return Date();
  };
  ```
- Use the exports keyword to make properties and methods available outside the module file.
- Save the code above in a file called “demomodule.js"

Include Your Own Module

```
var http = require('http');
var dt = require(‘./demomodule');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write("The current date and time is: " + dt.myDateTime());
    res.end();
}).listen(3000);
```

Node.js as a File Server

- var fs = require('fs’);
- Common use for the File System module:
- Read files - **fs.readFile();**
- Create files
- Update files
- Delete files
- Rename files

Creating a file

- The File System module has methods for creating new files:
  - fs.appendFile()
  - fs.open()
  - fs.writeFile()

-
  ```
  var fs = require('fs’);
  fs.appendFile(‘sample2.txt', 'Hello content!', function (err) {
    if (err) throw err;
    console.log('Saved!');
  });
  ```

Creating a file

- Create a new, empty file using the open() method:

-
  ```
  var fs = require('fs');
  fs.open(sample3.txt', 'w', function (err, file) {
    if (err) throw err;
    console.log('Saved!');
  });
  ```
- Create a new file using the writeFile() method:

-
  ```
  var fs = require('fs');
  fs.writeFile(sample4.txt', 'Hello content!', function (err) {
    if (err) throw err;
    console.log('Saved!');
  });
  ```

Update a file

- The File System module has methods for updating files:
- fs.appendFile()
- fs.writeFile()

Delete a file

```
var fs = require('fs');

fs.unlink(‘sample4.txt', function (err) {
  if (err) throw err;
  console.log('File deleted!');
});
```

Rename a file

```
var fs = require('fs');

fs.rename(‘sample4.txt’, ’sample4_renamed.txt', function (err) {
  if (err) throw err;
  console.log('File Renamed!');
});
```

@MaunashJani

NPM

- NPM is a package manager for Node.js packages, or modules.
- www.npmjs.com hosts thousands of free packages to download and use.
- The NPM program is installed along with Node.js
- A package in Node.js contains all the files you need for a module.
- Modules are JavaScript libraries you can include in your project.
- Download a Package:

```
> npm install package_name
```

- Using a Package

```
var pkg = require(‘package_name');
```

Node.js Upload Files

- The Formidable Module
- > npm install formidable
- var formidable = require('formidable’);
- Example - demo16.js

Popular Modules

- Express – Web application framework
- Request – Simplified HTTP Client
- Grunt
- Bower
- Underscore
- Passport
- Nodemailer
- Node MySQL
- Socket.io – Helps make real-time applications
- Mongoose – MongoDB Object Modeling
- Jade – Template engine
- restify – REST API framework
- And many more…

References

- https://www.youtube.com/watch?v=ztspvPYybIY
- https://www.w3schools.com/nodejs/
- https://code.tutsplus.com/tutorials/nodejs-for-beginners--net-26314
- https://www.tutorialspoint.com/nodejs/nodejs_first_application.htm
- https://www.youtube.com/playlist?list=PLTjRvDozrdlydy3uUBWZlLUTNpJSGGCEm
- https://www.youtube.com/playlist?list=PL4cUxeGkcC9gcy9lrvMJ75z9maRw4byYp
- https://www.youtube.com/playlist?list=PLC3y8-rFHvwhco_O8PS1iS9xRrdVTvSIz
- https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBMdkKFn3HasZnnAqVjzHn_
- http://howtonode.org/
- https://nodeschool.io/

Thank You

Maunash Jani

Software Developer

@Genius_Lynx

@MaunashJani

maunash@geniuslynx.com