

NodeJS Documentation Explained

Part - 1

Overview

_____Many Developers who are working in NodeJS or are starting their NodeJS journey, it has been observed that they have three mindsets in general i.e.:

1. To build any NodeJS Application, they have to use NPM Modules.
2. NodeJS Documentation ???, who will read this!!!
3. Node Talks ???, feeling like an alien language to me!!!

What Problem Developers Face ?

1. Developer's common phrase is, "We can't develop this/that feature as I'm unable to find any NPM Module".
2. Unnecessary code is part of the code with Big Bull "Node_Modules".
3. Bundling the Image will lead to a very large size. If we compare the basic express server and http server, then express server code is greater than 2000 times of http server code.
4. Code to be updated whenever any NPM Module changes their method or syntax in a new version, else it's a "Blocker".

Problem is because ?

1. Lack of awareness of NodeJS inbuilt modules.
2. Lacking in Core Concepts of CS Fundamentals, like OS, CN, DS, etc.
3. Lacking in interrelation of Fundamentals and NodeJS Modules.

Solution

Yes, I'm gonna make a bridge of the Core Fundamentals and Interrelation of these fundamentals with NodeJS Modules while explaining the NodeJS Documentation.

Like for example,

1. Start the server using ExpressJS vs Start server with Node's HTTP Module
 - a. Express is made on top of HTTP
2. HTTP module is based on "net.server" which uses EventEmitter internally which internally uses sockets (core UNIX Fundamentals)
 - a. Many Modules are built on top of EventEmitter only

Let's Start

The NodeJS Documentation has 3 types of documentation:

1. API Reference Documentation
2. ES6 Features
3. Guides

API Reference Documentation

- Contribution
 - Submit a bug
 - Submit Contribution
- Stability Index
 - 0 : **Deprecated**
 - 1 : **Experimental**
 - 2 : **Stable**
 - 3 : **Legacy**
- JSON Output
- System Calls Man Pages
-

APIs

| API Name | Description |
|----------------------|--------------------------------------------------------|
| assert | Provides a set of assertion tests |
| buffer | To handle binary data |
| child_process | To run a child process |
| cluster | To split a single node process into multiple processes |
| crypto | To handle OpenSSL cryptographic functions |
| dgram | Provides implementation of UDP datagram sockets |
| dns | To do DNS lookups and name resolution function |
| domain | To handle errors |
| events | To handle events |

| | |
|----------------|--------------------------------------------------|
| fs | To handle the file system |
| http | To make nodejs act as an http server |
| https | To make nodejs act as an https server |
| net | To create servers & clients |
| os | Provides information about the OS |
| path | To handle file paths |
| punycode | A character encoding scheme |
| querystring | To handle URL query string |
| readline | To handle readable streams one line at a time |
| stream | To handle streaming data |
| string-decoder | To decode buffer objects into strings |
| timers | To execute a function after a given number of ms |
| tls | To implement TLS and SSL protocols |
| tty | Provides classes used by a text terminal |
| url | To parse URL strings |
| util | To access utility function |
| v8 | To access information about v8 |
| vm | To compile JS code in a vm |
| zlib | To compress or decompress files |
| | |