Node.js

What is Node.js?

* Node.js is an open source server environment
* Node.js is free
* Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* Node.js uses JavaScript on the server

A common task for a web server can be to open a file on the server and return the content to the client.

Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

What Can Node.js Do?

* Node.js can generate dynamic page content
* Node.js can create, open, read, write, delete, and close files on the server
* Node.js can collect form data
* Node.js can add, delete, modify data in your database

What is a 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"

**REPL**

**Read  
Evaluation  
Print  
Loop**

Enter REPL by entering:

node

For viewing all possibilities of a keyword:

Press tab twice

Now, it will work like the console in Chrome Developer Tools

Type .exit to exit

Or press Ctrl + C twice

Node has various libraries. These libraries can be accessed using require method.

The various inbuilt libraries can be seen at <https://nodejs.org/api/modules.html>

Example:

**fs.copyFileSync(src, dest[, mode])**[**#**](https://nodejs.org/api/fs.html#fs_fs_copyfilesync_src_dest_mode)

Added in: v8.5.0

* src [<string>](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#String_type) | [<Buffer>](https://nodejs.org/api/buffer.html#buffer_class_buffer) | [<URL>](https://nodejs.org/api/url.html#url_the_whatwg_url_api) source filename to copy
* dest [<string>](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#String_type) | [<Buffer>](https://nodejs.org/api/buffer.html#buffer_class_buffer) | [<URL>](https://nodejs.org/api/url.html#url_the_whatwg_url_api) destination filename of the copy operation
* mode [<integer>](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#Number_type) modifiers for copy operation. **Default:** 0.

Synchronously copies src to dest. By default, dest is overwritten if it already exists. Returns undefined. Node.js makes no guarantees about the atomicity of the copy operation. If an error occurs after the destination file has been opened for writing, Node.js will attempt to remove the destination.

mode is an optional integer that specifies the behaviour of the copy operation. It is possible to create a mask consisting of the bitwise OR of two or more values (e.g. fs.constants.COPYFILE\_EXCL | fs.constants.COPYFILE\_FICLONE).

* fs.constants.COPYFILE\_EXCL: The copy operation will fail if dest already exists.
* fs.constants.COPYFILE\_FICLONE: The copy operation will attempt to create a copy-on-write reflink. If the platform does not support copy-on-write, then a fallback copy mechanism is used.
* fs.constants.COPYFILE\_FICLONE\_FORCE: The copy operation will attempt to create a copy-on-write reflink. If the platform does not support copy-on-write, then the operation will fail.

const fs = require('fs');

const { COPYFILE\_EXCL } = fs.constants;

// destination.txt will be created or overwritten by default.

fs.copyFileSync('source.txt', 'destination.txt');

console.log('source.txt was copied to destination.txt');

// By using COPYFILE\_EXCL, the operation will fail if destination.txt exists.

fs.copyFileSync('source.txt', 'destination.txt', COPYFILE\_EXCL);

NPM

Node Package Manager: use packages for your project which are prebuilt by people previously

Initialize

npm init