25-Nov-2021

Node Modules (folder)

Smart phone — Apps · — Amazon, fk
Zomato, swiggy
games
financial Apps

App — 1 universal App.

Modules

Node.JS → Inbuild
Node.JS → externally

externally → npm (node package manger)
externally → yarn

Inbuild Modules

1) Child process :- cp.JS (file name)

require (keyward) ('child-process') (Module name)

cmd
clc → (open Calculator)
Code (open VS Code)
(similar child process work)

```
const cp = require('child-process')
Console.log('Trying to open Calculator');
    cp.execSync('Calc')

Console.log('Calculator opened');
```

Terminal → node cp.JS

```
console.log ('Trying to open Vs Code')
cp.execSync ('Code')
Console.log ('Vs Code Opened')
```

```
cp.execSync ('start chrome')
```

(create for test) cp.execSync ('Start chrome https://www. — .com')

```
→ text.js
  ↳ Console.log ('I am a test file')
```

```
let output = cp.execSync ('node test.js')
    Console.log (output)
```

↓ data is Coming in Buffer form ↓ (means) Binary form

```
console.log ("Output is " + output);
```

(first Concat)

Child process → to Create Sub process.

OS modules

OS.JS (file)

operating System → (for communicate with operating System)

```
          variable          module (package)
const os = require('os')

                        (archite)
Console.log(os.arch()
```

1. console.log(os.arch())
2. Console.log(os.platform())
3. console.log(os.networkInterfaces())
4. Console.log(os.cpus())

Imp File System

fs.JS (file)

// Node Modules are used to perform different tasks as required

// File System Modules :-

// 1:- read , write , update , delete

* → ~~f1.text~~

```
Const fs = require('fs')  (readFileSync is a method to read a file)

let data = fs.readFileSync('f1.txt')

Console.log('This is f1 Content' + data)
```

* Writing to a file

(अगर पहले से है file तो वहा data change हो जाएगा।)

(create हो जाएगी)

```
fs.writeFileSync('F2.txt', 'Hello from F2")
```

writeFileSync(' ', ' ') expect Two things
method is used to put data (write) to a file.
If the file passed doesn't exist it creates a new file and writes to it.

* Update a file (append)

```
fs.appendFileSync('f2.txt', 'this is F2 data')
```

appendFileSync adds data to the previously written file.
Basically it appends a file with the new data passe

* Delete a file by Passing It's path

UnlinkSync()

```
fs.unlinkSync('f2.txt')
Console.log('File removed')
```

Directories

① Create // mkdirSync() is used to creat a new directory.

```
fs.mkdirSync('myDirectory')
```

② Delete // rmdirSync() is used to delete a Directory.

```
fs.rmdirSync('myDirectory')
```

③ path

to check whether a directory exist or not we can use existsSyn method

```
let doesExist = fs.existsSync('myDirectory')
    console.log(doesExist)
```

→ return
① True
② false

If the directory exist the method return True otherwise false

my Directory
↳ 3.txt
↳ 4.txt

④ stats of a path (~~what are the Contents inside~~ details of a folder of a folder) (lstatsync provides us with all the statistics of a directory)

```
let statsOfPath = fs.lstatSync('my Directory')
    console.log(statofPath)
```

⑤ isfile() is a method whether a passed path is a file or not.

```
console.log('isFile?', statsofPath.isfile())
```
— false

⑥ isDirectory() is a method that check whether a passed path is a director or not.

```
console.log('isDirectory?', statsofPath.isDirectory())
```
— True.

Read Directory (readdirSync()) is used to check the Content of a particular folder/directory.

readdirSync()

(7)
```
let folderPath = ' D:\\FJP2 Dev\\Node Modules\\ myDirectory'

let folderContent = fs.readdirSync (folderPath)

    Console.log( folderContent)
    Console.log('directory content' + folder Content)
```

(8) **Copy files**

mydirectory → f3.txt
my Directory2 (Srcfolder, dest folder)

basename ↓ f3.txt

```
let srcfolder = ' D:\\FJP2 Dev\\ Node Modules\\ mydirectory'.

let destfolder = ' D:\\ ______ ______ '.
```


```
const path = require ('path')
```

```
let tobecopiedfileName = path.basename(srcfilePath)
```
// basename method gets the actual name of the file or directory
```
Console.log (tobecopiedfileName) → o/p → f3.txt
```

```
let destPath = path.join (destfolder , tobeCopied fileName)
```
↳(ये JS का fn है)
```
    Console.log (destPath)

fs.copyfileSync ( srcfilePath, destPath)
    Console.log ( "file Copied")
```