

Table of Contents

JavaScript Dec 1995-2009 April	2
JavaScript Can Do :	2
JavaScript Can Not Do:	2
JavaScript From May-2009	2
Using Node JS Platform JavaScript Can Do:.....	2
Node JS	3
Node JS Application	3
Terminal	4
TOOLS.....	4
TERMINAL.....	4
Node.js Module.....	4
Node.js Core Module	4
Http/Https Protocol	5
HTTP REQUEST METHODS	5
Http Client	6
Example Of HTTP Client	6
HTTP Client For Testing	6
Package.json	6
Node.js URL Module	6
Built-in URL Module.....	6
Node.js fs module	7
Node.js fs module essentials operation Asynchronous.....	7
Node.js fs module essentials operation Synchronous.....	8
Synchronous Theory.....	9
Asynchronous Theory.....	10

JavaScript Dec 1995-2009 April

JavaScript created as **client side language/front end language**. Only handle front end logics .
JavaScript has no power to communicate with server.

JavaScript Can Do :

- Interact with temporary storage
- Make interactive web pages
- Interact with local storage
- Sending request for data to server
- Send request to server
- work as an interface between server and user

JavaScript Can Not Do:

- Querying the database
- Operations over databases
- Access/Write a file on server.
- Server Side Request , Response Handle
- Software Backend Operations.

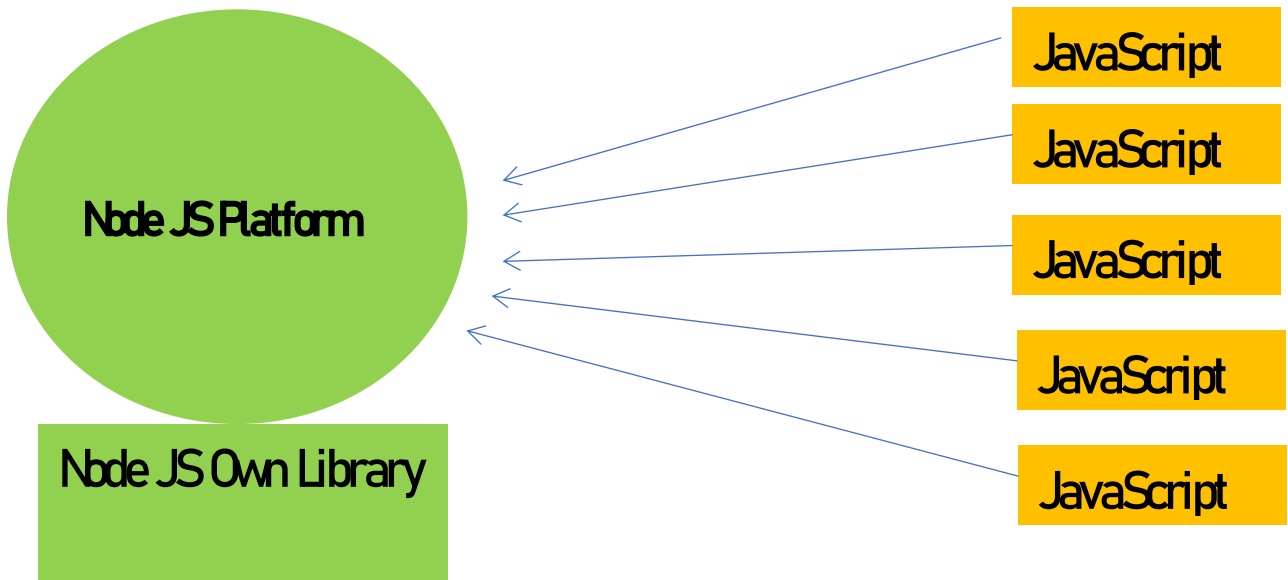
JavaScript From May-2009

Ryan Dahl Create An Awesome Platform May -2009, This is Node JS

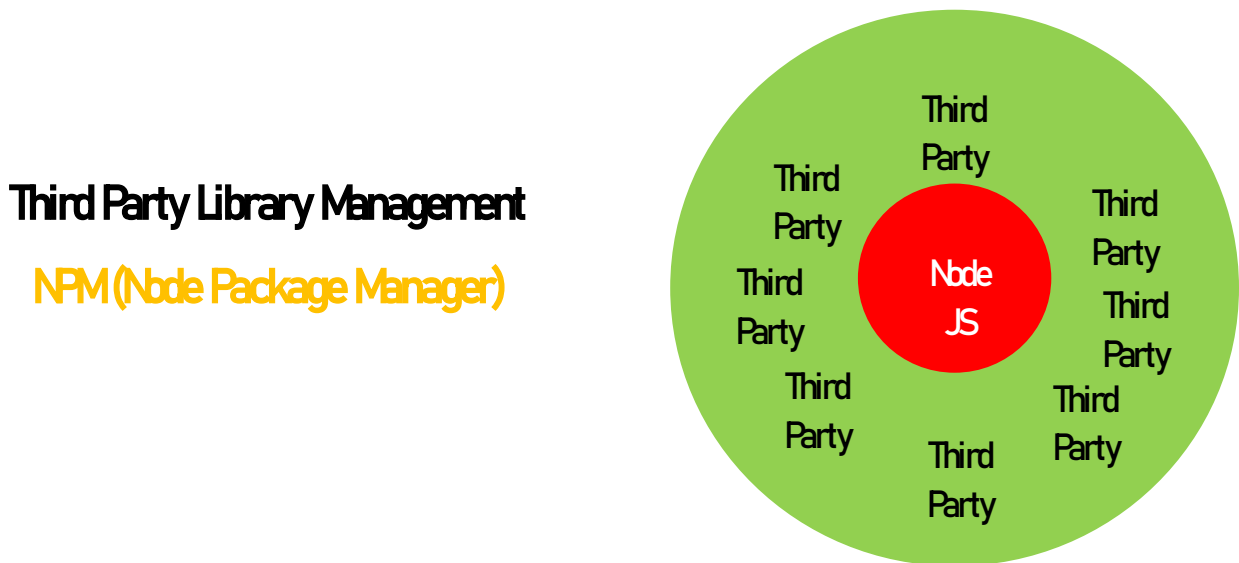
Using Node JS Platform JavaScript Can Do:

- Querying the database
- Operations over databases
- Access/Write a file on server.
- Server Side Request , Response Handle
- Software Backend Operations.

Node JS



Node JS Application



Terminal

TOOLS

- Node.js
- VS Code
- Webstrom

TERMINAL

- Windows CMD
- Windows PowerShell
- VS Code Terminal Option
- Webstrom Terminal
- Node.js Terminal

Node.js Module

Module in Node.js is a simple / complex functionality organized in single or multiple JavaScript files.

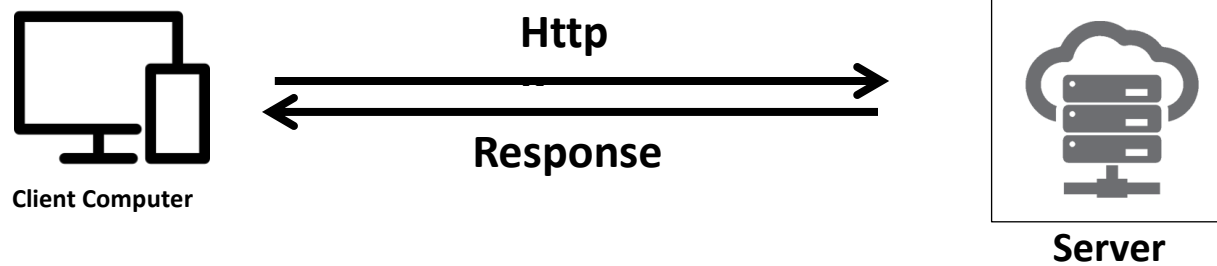
Node.js includes three types of modules:

- Core Modules
- Local Modules
- Third Party Modules

Node.js Core Module

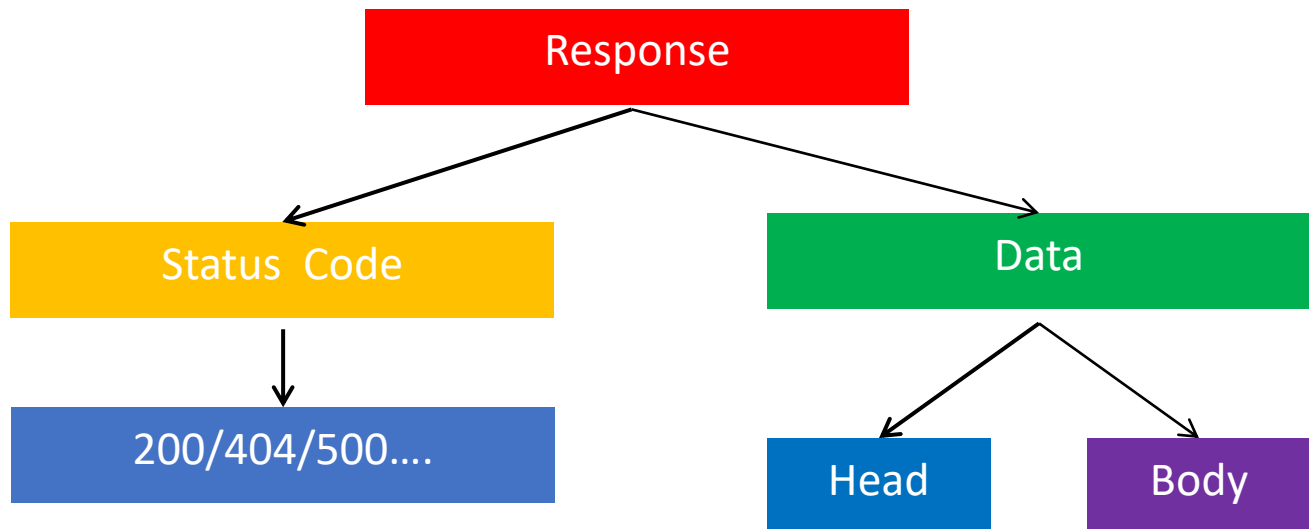
Core Module	Description
http	http module includes classes, methods and events to create Node.js http server.
url	url module includes methods for URL resolution and parsing.
querystring	querystring module includes methods to deal with query string.
path	path module includes methods to deal with file paths.
fs	fs module includes classes, methods, and events to work with file I/O.
util	util module includes utility functions useful for programmers.

Http/Https Protocol



HTTP REQUEST METHODS

- GET()
- DELETE()
- POST()
- PUT ()
- OPTIONS()
- HEAD()



Http Client

যাহা সার্ভারে HTTP Request পাঠায় এবং Response Receive করে, তাকেই HTTP Client বলে

Example Of HTTP Client

- Axios
- Fetch
- JQuery Ajax
- cURL
- Volly
- Retrofit
- RestSharp

HTTP Client For Testing

- Postman
- Fiddler

Package.json

Application এর নথি

- Holds various metadata relevant to the project
- handle the project's dependencies
- Used identify the project
- Used to hold application information

Node.js URL Module

Built-in URL Module

- The URL module splits up a web address into readable parts.
- `http://rabbil.com/blog.html?year=2020&month=july;`

Node.js fs module

- Node.js includes **fs module** to access physical file system.
- The fs module is responsible for all the **asynchronous or synchronous** file I/O operations.
- Synchronous: Loading.... Loading... you wait for it to finish....After finish you can move.
- Asynchronous: No Loading... No Loading ... no need to wait.... you can move.

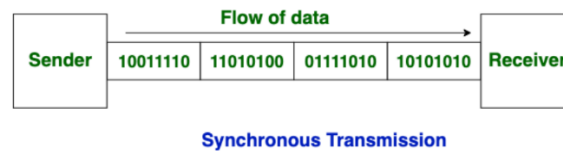
Node.js fs module essentials operation Asynchronous

Method	Description
fs.readFile(fileName [,options], callback)	Reads existing file.
fs.writeFile(filename, data[, options], callback)	Writes to the file. If file exists then overwrite the content otherwise creates new file.
fs.rename(oldPath, newPath, callback)	Renames an existing file.
fs.exists(path, callback)	Determines whether the specified file exists or not.
fs.unlink(path, callback);	File delete
fs.appendFile(file, data[, options], callback)	Appends new content to the existing file.
fs.open(path, flags[, mode], callback)	Opens file for reading or writing.
fs.mkdir(path[, mode], callback)	Creates a new directory.
fs.rmdir(path, callback)	Removes an existing directory.
fs.readdir(path, callback)	Reads the content of the specified directory.

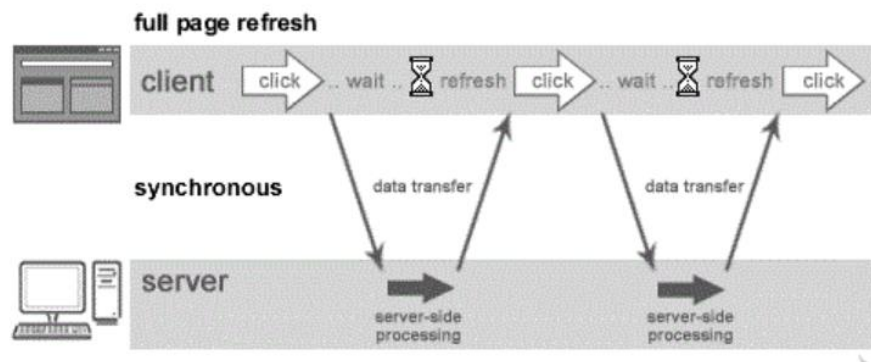
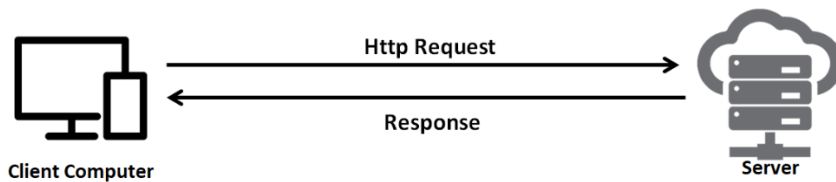
Node.js fs module essentials operation Synchronous

Method	Description
fs.readFileSync(fileName [,options])	Reads existing file.
fs.writeFileSync (filename, data[, options])	Writes to the file. If file exists then overwrite the content otherwise creates new file.
fs.renameSync(oldPath, newPath)	Renames an existing file.
fs.existsSync (path)	Determines whether the specified file exists or not.
fs.unlinkSync (path);	File delete
fs.appendFileSync (file, data[, options])	Appends new content to the existing file.
fs.openSync (path, flags[, mode])	Opens file for reading or writing.
fs.mkdirSync (path[, mode])	Creates a new directory.
fs.rmdirSync (path)	Removes an existing directory.
fs.readdirSync (path)	Reads the content of the specified directory.

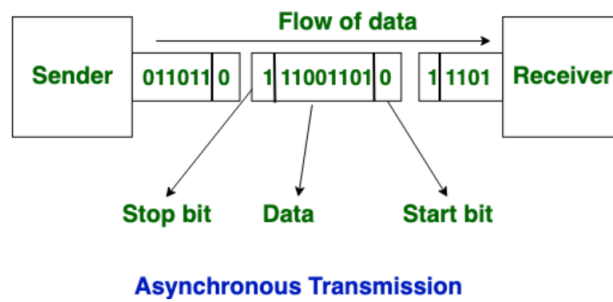
Synchronous Theory



- Data is sent in form of blocks or frames.
- This transmission is the full duplex type.
- There is no gap present between data
- Synchronous transmission is fast.
- More efficient and more reliable for large amount of data.



Asynchronous Theory



- This transmission is the half duplex type transmission
- Start bits and stop bits are added with data.
- Data is sent in form of byte or character.
- Asynchronous transmission is slow.
- There is present gap between data.

