Page No.
Date

Theory Assignment 1

Name : Fozeem Patel
Rollno : 44
Semester: 7th
Subject : 701
Date : 24/07/2023

Page No.
Date

Q:1 Nodejs : Introduction, features, execution architecture

→ Introduction

- Its a cross-platform runtime environment and library for running JavaScript applications outside the browsers.
- It is used for creating server-side and networking web applications.
- Its open source and free to use.
- Basic modules of Nodejs are written in JavaScript
- NodeJs is Run time Environment
- Its an Built in module and scalable network applications.
- It uses an event-driven, non blocking I/O model that make it lightweight and efficient.

Node.Js = C++ + Js Libraries
+
V8

→ Features

o Extremely fast
- its built on Google chrome's V8 Javascript Engine, so its library is very fast in Code execution

o I/O is Asynchronous and Event Driven
- Libraries are asynchronous in NodeJs. Nodejs based server never waits for an API to return data. The server moves to the next

API after calling it and a notification mech of Events of Nodejs helps server to get a response from the previous API Call.

- Single threaded
  - it follows a single threaded model with event looping

- Highly scalable
  - more user interaction is available.

- No buffering
  - it cuts down the overall processing time while uploading audio and video files, its applications never buffer any data.

- Open source
  - its has an open source community which has produced many excellent modules and add additional capabilities to Nodejs application

- License
  It is related under the MIT license.

Date

→ Execution Architecture

o Event Loop
- Nodejs execution model is the event loop. The event loop is responsible for handling I/O operations asynchronously and efficiently
- It allows Node Js to process multiple requests Concurrently without blocking the execution of other tasks

o Event-driven and Non-blocking I/O:
- It registers event listeners for specific events and responds to them when they occur
- This approach is in contrast to traditional synchronous programming where each operation would block the execution until it Completes
- Non-blocking is a critical aspect of Nodejs. When an I/O operation is initiated, Nodejs does not wait for it to complete Instead it continues executing other tasks while listening for the I/O operation to finish

o Callbacks
- they are functions that are passed as arguments to other functions and are executed later, typically upon the completion of an asynchronous operation.
- They plum a central role in the event-driven

architecture of Nodejs

- Single threaded Event loop
  - it runs on a single thread, which might seem counterintuitive for handling concurrent connections,
  - the event loop's non-blocking nature enables Nodejs to achive high concurrency without the need for multiple threads.
  - This way the event loop remains free to handle I/O operation effic

- Libuv
  - its a library that provides the platform-independent event loop implementation for nodejs
  - it handles the low level I/O operations and abstracts the underlying operating system facilities

- C++ Bindings
  - Certain parts of Nodejs core which such as V8 and libuv are written in C++

Q. 2 Note on modules with example.

- Modules are an essential concept that enables code organization and reusability modules are encapsulated pices of code that can be used to define functions, classes or variables that can be easily imported and utilized in other part of your application.

- NodeJs supports two typs of modules:
  - Core Modules (built-in modules)
  - User defined Modules (Custom modules
  - Th

- Core Modules
  - it has many built-in do modules that are part of the platform and come with Nodejs installation.
  - these modules can be loaded into the program by using the required function

Syntax
```
Const module = require ('module_name');
```

- The require() function will return a JavaScript type depending on what the particular module returns.

ex

```
Const http = require ('http');
Const fs = require('fs');

Const server = http.createServer (function(req,res)
    Console.log ("Req received =>" + req.url);
});
Server.listen(5000);
Console.log ("Listening on port no 5000");
```

Some Core modules are given below

- http
- assert
- fs
- Path
- Process
- os
- querystring
- url

- User defined
- Third-party modules are modules that are available online using the Node package Manager
- These modules can be installed in the project folder or globally
- Some of the popular third-party modules are mongoose, express, angular and React

ex

```
npm install express
npm install mongoose
npm install -g
```

Q:3 Note on package with example

- it provides online repositories for node.js package which are searchable on
- it also provides command line utility to install Nodejs package do version management and dependency management of Nodejs packages.
- NPM is a short form of Node Package manager, which is the world largest software registry.
- The open source web project developers use it the Nodejs project for installing packages in the project dependency management and event version management.

Components of npm

- website:
The npm official website is used to find package for your project, create and set up profiles to manage and access privet and public

- Command Line Interface:
The CLI runs from your computers terminal to interact with npm package and respositories

- Registry:
it is a large and public database of JavaScript project and meta-info.

Q: 4 use of Package.json and Package-lock.json

O Package.json

- it is a crucial part of NodeJs.

1 Dependency Management
- one of the primary purpose of package.json is to manage projects dependencies. it lists all the external package that the project depends on, along with their versions.

2 Project Information
- file Containg metadata about the projed such as the projet name description, version author, license and other relevent information

3 Scripts
- it allows you to define Customs scripts that can be executed using the npm run Command

4 NPM Command
- file enables you to specify default behaviors for certain NPM Commands For example, you can set the default behavior for npm start.

Date

⊙ Package-lock.json

1 Dependency Version Locking
- this manage reads the package-lock.json file to ensure that the exact version of dependencies specified in that file are installed.

2 Dependency Tree
- it contains a detailed tree-like representation of the project entire dependency tree. This include both the direct and indirect dependencies.

3 Reproducible Builds
- different developers working on the same project can ensure they have the same dependencies installed with the same version.

4 Faster Installations

5 Security and Integrity
- file includes cryptographic hashes for each installed package

Q.5 Nodejs Packages

- Express
  - A fast and minimalist web application framework for building APIs and web applications

- lodash:
  - A utility library that provides helpful functions for working with arrays, objects, strings and more

- axios
  - A popular HTTP client that makes it easy to send HTTP requests from Nodejs or the browser

- bcrypt
  - A library for hashing passwords and providing a secure way to store sensitive info

- Jsonwebtoken
  - A library for generating and verifying JSON Web Tokens used for authentication

- Mongoose
  - An object modeling tool for MongoDB, making it easy to work with MongoDB database,

- Socket.io
  - A real time WebSocket library for enabling

bidirectional communication between client and server

- async
  - A utility library for handling asynchronous operation in JavaScript

- nodemailer
  - A module for sending email from Node.js application

- morgan
  - A HTTP request logger middleware for Node.js

- body-parser:
  A middleware for parsing incoming request bodies in Express application

- chalk
  - A library for adding colors to the Console output

- request
  - A simple HTTP request client for Node.js

Page No.
Date

Q:6 npm introduction and Commands with its use.

- npm is a short form of Node package manager, which is the world's largest software registry. The open source web project developers use it from entire world to share and borrow packages.
- The npm also act as a Command-line utility for the Node.js project for installing package in the project dependency manage and even version management.

→ Components of npm

1 website
- The npm official website is used to find packages for your project, create and set up profiles to manage and access private and public packages.

2 Command line Interface
- The CLI runs from your Computer's terminal to interact with npm package and repositories

3 Registery
- The registry is a large and public database of JavaScript project and metadata information. you can use any supported npm that

want or even your own, you can even use someone else's registry as per their terms of use

→ Use of npm is given below

- Package Installation
- Dependency Management
- Package Initialization
- Script Execution
- version management
- Global Package
- Publishing Package
- Security Scanning
- Version Updates
- Package searching

Q:7 Describe use and working of following Nodejs packages, important properties and methods and relevent programs

o Url Package

- This package is particularly useful when working with URL in Nodejs Application. it allows you to parse incoming URL and construct new URL.
- you might use the url.parse method to extract query parameters from the URL of an incoming HTTP request.

→ Properties
url.href
url.protocol
url.host
url.hostname
url.port
url.pathname

→ Methods

o url.parse(urlString[, parseQueryString[, slashesHost]])

o url.format(urlObject)

o url.resolve(from, to)

Date

o Process

The process object is especially useful for gathering runtime information and Controlling the behavior of the Node.js process.
- Accessing Command-line Arguments
- Environment Variables
- Exiting the process
- Event Handling

→ Properties
process.argv
process.env
process.pid
process.platform
process.version
Pro

Ex
```
process.on('exit', (code) => {
    console.log('Exiting with code ${code}');
});
```

o readline

Ex
```
const readline = require('readline');
const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});
```

```
rl.question("What is your name? ", (name) =>
    { console.log(`Hello, ${name}!`);
      rl.close();
    });
```

- this module is commonly used for building interactive command-line interfaces in Node.js applications, allowing developers to interact with users in a user-friendly way

→ Properties

- readline.createInterface(options)
  - This method creates a new Interface for reading input from the user.

→ Functions

- readline.Interface.question(query, callback)
- readline.Interface.close();

0 fs

- this is used to read from a file, write to a file and list the contents of a directory asynchronously
- it provides an easy-to-use interface for interacting with files and directories, making it crucial part of many Node.js Apps

Page No. 5
Date

eg: Const fs = require('fs');

```
fs.readfile('file.txt', 'utf8', (err, data) => {
    if(err)
    { Console.error('error', err); }
    else {
        console.log('File Content', data);
    }
});
```

→ Properties

fs.constants
- An object containing related to file system operations such as file access module.

→ Functions

- fs.readFile(Path [options] callback)
- fs.writeFile(file, data [options] callback)
- fs.readdir(path [options], callback)
- fs.unlink(path, callback)
- fs.mkdir(Path [options], callback)

Date

Events

- -'his creates a custom event emitter and handle events
- it used for implementing event driven architecture, custom event handling and decoupling different parts of the application
- it allows different components or modules to communicate and react to events making it a powerful tool for building flexible and scalable applications

→ Properties

- events.EventEmitter

→ Methods

- EventEmitter.on(event, listener)
- EventEmitter.once(event, listener)
- EventEmitter.emit(event [argument])
- EventEmitter.off(event, listener)

Console

- it is an essential tool for logging information, debugging and inspecting data during the development and testing phase of Node.js applications.

Page No.
Date

it helps developers understand what is happening in the code, identify issues and monitor the application's behavior.

→ Methods

Console.log
Console.info
Console.error
Console.warn
Console.dir
Console.time
Console.timeEnd
Console.trace

○ buffer

- it is widely used in scenarios where raw binary needs to be processed, such as handling network protocols, reading and writing files and working with stream.
- developers should be cautious when using Buffer due to potential for security vulnerabilities such as buffer overflows.
- it is essential to handle binary data carefully

→ Methods
Buffer.alloc

Buffer.from
Buffer.isBuffer
buf.toString
buf.toJSON
buf.compare

0 http

ex

```
Const http=require('http');
Const server=http.createServer((req,res)=
    res.writeHead(200,{'Content-Type'});
    res.end('Hellow');
});
server.listen(3000,()=>{
    Consde.log("Listening 3000");
});
```

→ Properties

http.METHODS
http.STATUS_CODES

→ Method

http.createServer
http.get
http.request

Date

c) V8

- you can access some V8-related functionalities through the process global object, specifically the property.
- it is developed by Google and is also used in the chrome web browser.
- it is responsible for executing JavaScript code in Node.js and provides many low-level features, such as memory management, garbage collection and Just-in-Time Compilation.
- we have to specify requirements related to the V8 engine or need to access low level details.