Long ques of node js

Que 14:

Node is server-side platform. It is built on Google Chrome's JavaScript Engine

(VS Engine), It was developed by Ryan Dahl in 2009 and its latest version is 16.13.1 (includes npm 8.1.2)

The definition of Node is given by its official documentation is as follows-

"Node is is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data- intensive real-time applications that run across distributed devices

" Node.js is an open source, cross-platform runtime environment.

It is used for developing server-side and networking applications.

Node.js applications are written in JavaScript. It can be run inside the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a huge library of different JavaScript modules Such libraries make the development of web applications easier.

Node.js - Runtime Environment + JavaScript Library

Features of Node.js

Asynchronous : All APIs of Node is library is asynchronous. This means that they are non-blocking. Server based on Node.js never waits for an API to return data. The server moves to the next API after calling it.

Event Driven : The event notification of Node is helps the server to get a response from the previous API call

Very Fast : Since it is built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

Single Threaded : Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way

Highly Scalable : Node.js uses a single threaded program and the same program can provide service to many requests making it highly scalable

No Buffering : Applications of Node.js never buffer any data. They just give data as output in chunks.

License : Node.js is released under the MIT license.

Que 15:

HTTP Module

Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP).

To include the HTTP module, use the require() method:

```
var http = require('http');
```

The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.

The createServer() method is used to create an HTTP server:

Example:

```
var http = require('http');
//create a server object:
http.createServer(function (req, res) {
res.write('Advanced Web Technology!'); //write a response to the client
res.end(); //end the response
}).listen(3000); //the server object listens on port 3000
```

The function passed into the http.createServer() method, will be executed when someone tries to access the computer on port 3000.

Que 16:

Components of node.js:

Node JS contains many components to develop, test and deploy applications.

Following are the list of Node JS components:

- Node CLI
- NPM
- package.json
- Node Modules
- Development Tools and Frameworks

Node CLI:

Node JS has a CLI (Command Line Interface) to run the basic commands and script files.

This component gets installed by default while installing Node JS Platform.

We can open command prompt and type basic JavaScript commands or even files on the command prompt.

NPM :

NPM stands for Node Package Manager. NPM is used to install, update, uninstall and configure Node JS modules/packages very easily.

When we install Node JS Base Platform, it installs only few components, modules, and libraries like Node CLI, NPM etc. Later on, we can use NPM to upgrade Node JS with our required modules.

To check npm version, run "npm -v command on Node CLI.

package.json

"package.json" is a plain text file in ISON format. It is used to manage our application required module dependencies. We should place this file in our application root folder. It defines information like our application name, module dependencies, module versions etc. This configurations file is very important and requires more time to explain in detail. We will discuss it in detail with some examples in coming posts. Myownapp package.json file;

```
{
        "name": "myownapp",
        "version": "1.0",
        "dependencies": {
        }
}
```

Node Modules :

Node JS is a modular platform. Each functionality of it is implemented by a separate module or package. It has some core modules like npm, install, uninstall, update etc and rest all modules are third-party modules.

When we install Node JS Platform, by default only one module is installed i.e.npm module. We need to use "npm" command to install required modules one by one.

Development Tools and Frameworks:

Many companies have developed some tools and framework to ease and reduce the overhead of Node JS applications since the Node JS Platform has become very popular in developing Data-Sensitive Real-time and Network applications.

Que 17:

Request object methods:

1. Accepts() : This method checks if the specified content types are acceptable, based on the request’s Accept HTTP header field.
2. Get() : This method returns the specified HTTP request header field.

3. Is () : This method returns true if the incoming request’s "Content-Type" HTTP header field matches the MIME type specified by the type parameter.
4. Param(): This method returns the value of param name when present.

Response object method:

1. End() : used to end the response process. It signals that server that the response Is complete.
2. Append() : used to append data to the HTTP response header field
3. Get() : used to provide HTTP response header specified by field.
4. Set() : set the response’s HTTP header field.

Que 18:

User defined modules:

User defined modules are created locally in your node.js application. You can create you own modules, and easily include them in your application.

Example:

Calc.js

```
    exports.add = function(x.y){
        return x+ y;
    };
```

The exports keyword is used to make properties and methods available outside the module file. Since this file provides attributes to the outer world via exports, another file can use its exported functionality using the require() function.

Implementation.js

```
var calculator = requires(‘cals.js’)

var a = 10;

var b = 20;

console.log(“addition of the numbers is ” + calculator.add(a,b);
```

Que 19:

The file system module of node js has following method for creating new file.

fs.appendFile()

fs.open()

fs.writeFile()

1. fs.open()

fs.open() method is used to open a file for different purpose depending on the flag.

Following is the syntax of the method to open a file in asynchronous mode-

```
fs.open(path, flags, mode), callback)
```

In the above syntax:

path - filename including path that needs to be opened.

flags - behavior of the file to be opened.

mode - It sets the file mode (permission), but only when the file was created. It defaults to 0666, readable and writeable.

callback - This is the callback function which gets two arguments (err,fd). err- error while performing any operation on file and fd is file data

2. fs.writefile():

method is used to create a new file if it does not exist. It will over-write the file if the file already exists.

Syntax:

```
fs.writeFile(filename, data[, options], callback)
```

- path-string having the filename including path.
- data -The string or buffer to be written into the file.
- options - It's an object which will hold (encoding, mode, flag). By default. encoding is utf8, mode is octal value 0666. And flag is 'w'.
- callback - This is the callback function which gets a single parameter err that returns an error in case of any writing error.

3. fs.appendFile()

fs.appendFile() method is also used to create a new file if it does not exist. It will append the content in the file if it already exists.

Syntax:

```
fs.appendFile(filename, data[, options], callback)
```

path - string having the filename including path.

data -The string or buffer to be written into the file.

options - It's an object which will hold (encoding, mode, flag). By default. encoding is utf8, mode is octal value 0666. and flag is 'a'.

callback - This is the callback function which gets a single parameter err that returns an error in case of any writing error.

Que 20:

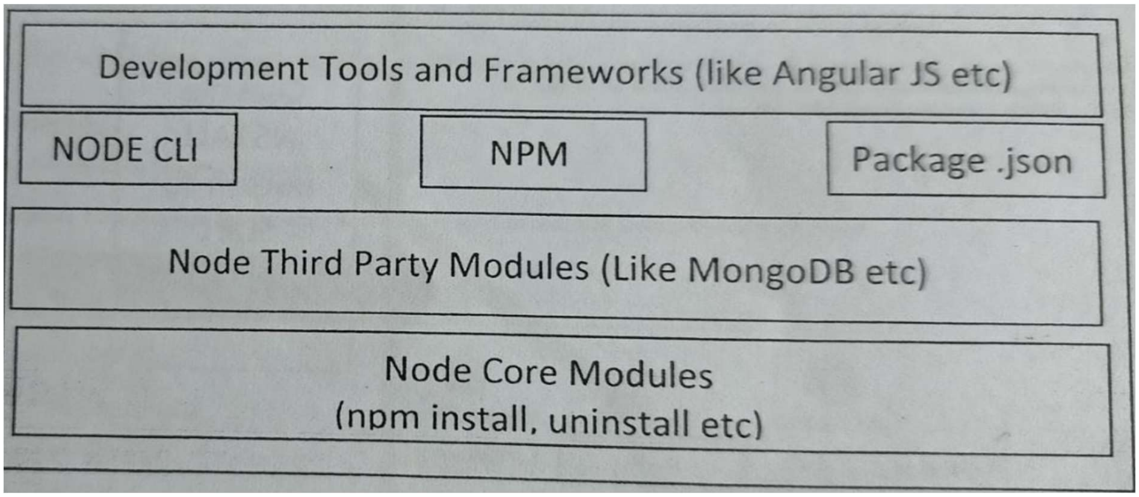
Node.js as a Web Server

Web server is required to access the web pages of any web application. The web server handles all the http requests coming from the web application. For example, Apache web server is used for PHP web applications, and IIS web server is used for ASP.NET web applications.

Node.js allows to create own web server that handle incoming HTTP requests asynchronously.Node.js has a built-in module called HTTP. This module enables Node.js to send data over HTTP. The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.

Que 34:

Node js platform diagram:


Development Tools and Frameworks (like Angular JS etc)
NODE CLI
NPM
Package .json
Node Third Party Modules (Like MongoDB etc)
Node Core Modules
(npm install, uninstall etc)

Node JS components:

- Node CLI
- NPM
- package.json
- Node Modules
-

Node CLI:

Node JS has a CLI (Command Line Interface) to run the basic commands and script files.

This component gets installed by default while installing Node JS Platform.

We can open command prompt and type basic JavaScript commands or even files on the command prompt.

NPM :

NPM stands for Node Package Manager. NPM is used to install, update, uninstall and configure Node JS modules/packages very easily.

When we install Node JS Base Platform, it installs only few components, modules, and libraries like Node CLI, NPM etc. Later on, we can use NPM to upgrade Node JS with our required modules.

To check npm version, run "npm -v command on Node CLI.

package.json

"package.json" is a plain text file in ISON format. It is used to manage our application required module dependencies. We should place this file in our application root folder. It defines information like our application name, module dependencies, module versions etc. This configurations file is very important and requires more time to explain in detail. We will discuss it in detail with some examples in coming posts. Myownapp package.json file;

```
{
    "name": "myownapp",
    "version": "1.0",
    "dependencies": {
    }
}
```

Node Modules :

Node JS is a modular platform. Each functionality of it is implemented by a separate module or package. It has some core modules like npm, install, uninstall, update etc and rest all modules are third-party modules.

When we install Node JS Platform, by default only one module is installed i.e.npm module. We need to use "npm" command to install required modules one by one.

Que 35:

Query string:

A query string is the portion of a URL where data is passed to a web application and/or back-end database. The reason we need query strings is that the HTTP protocol is stateless by design. For a website to be anything more than a brochure, you need to maintain state (store data). There are a number of ways to do this: On most web servers, you can use something like session state server-side. On the client, you can store via cookies. Or in the URL, you can store data via a query string.

On the world wide web, all URLs can be broken down into the protocol, the location of the file (or program) and the query string. The protocol you see in a browser is almost always HTTP; the location is the typical form of the hostname and filename (for example, www.techopedia.com/somefile.html), and the query string is whatever follows the question mark sign ("?").

For example, in the URL below, the bolded area is the query string that was generated when the term "database" was searched on the Techopedia website.

//www.techopedia.com/search.aspx?**q=database§ion=all**