DOP: / /2023 **DOS: / /2023**

Experiment No: 0

Title: Build Hello World App in Node.js

Theory:

What is Node.js:

Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use. It can be downloaded from this link https://nodejs.org/en/.

Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

Features of Node.js:

Following is a list of some important features of Node.js that makes it the first choice of software architects.

- **Extremely fast**: Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.
- **I/O is Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.
- **Single threaded**: Node.js follows a single threaded model with event looping.
- **Highly Scalable**: Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.
- **No buffering**: Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.
- **Open source**: Node.js has an open-source community which has produced many excellent modules to add additional capabilities to Node.js applications.
- **License**: Node.js is released under the MIT license.

Install Node.js on Windows:

To install and setup an environment for Node.js, you need the following two softwares available on your computer:

1. Text Editor.
2. Node.js Binary installable

Text Editor:

The text editor is used to type your program. For example: Notepad is used in Windows, vim or vi can be used on Windows as well as Linux or UNIX. The name and version of the text editor can be different from operating system to operating system.

The files created with text editor are called source files and contain program source code. The source files for Node.js programs are typically named with the extension ".js".

The Node.js Runtime:

The source code written in source file is simply JavaScript. It is interpreted and executed by the Node.js interpreter.

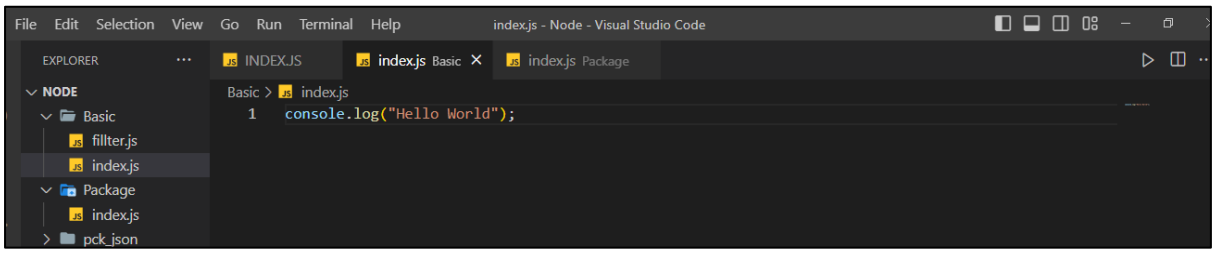
How to download Node.js:

You can download the latest version of Node.js installable archive file from https://nodejs.org/en/

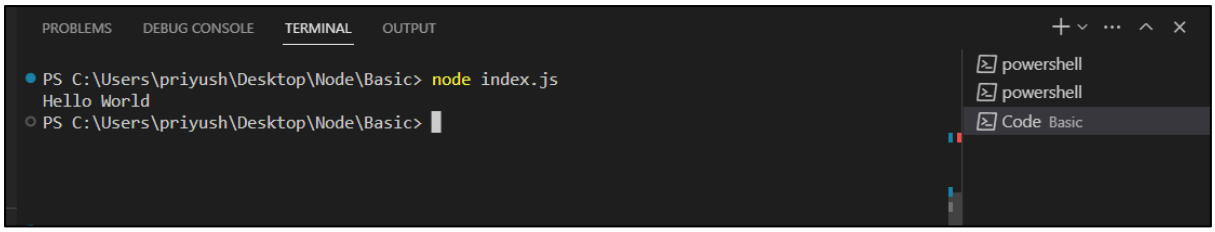
Hello World App in Node.js

Input:

Console base:



Output:

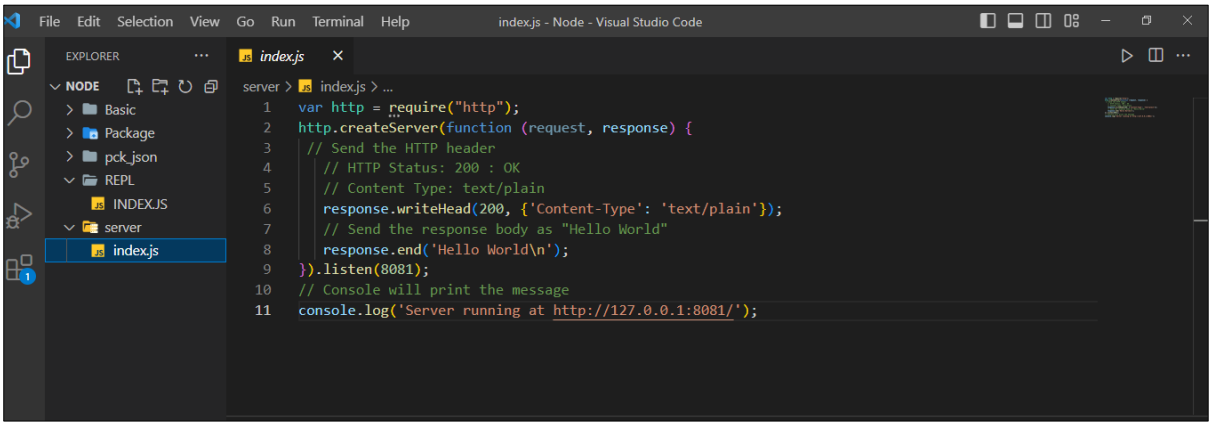


Node.js web-based Example

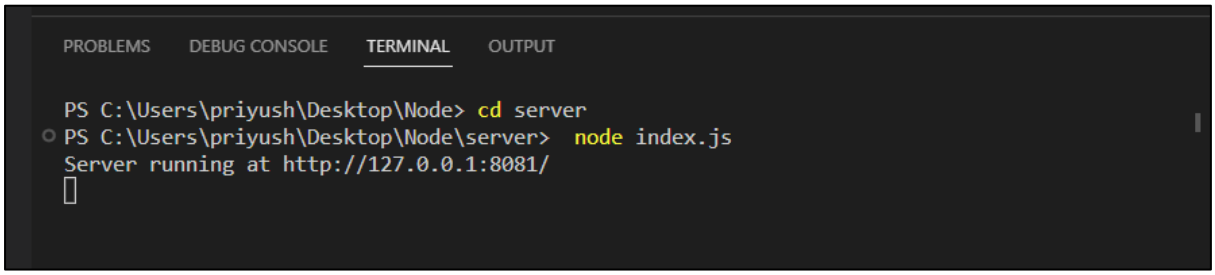
A node.js web application contains the following three parts:

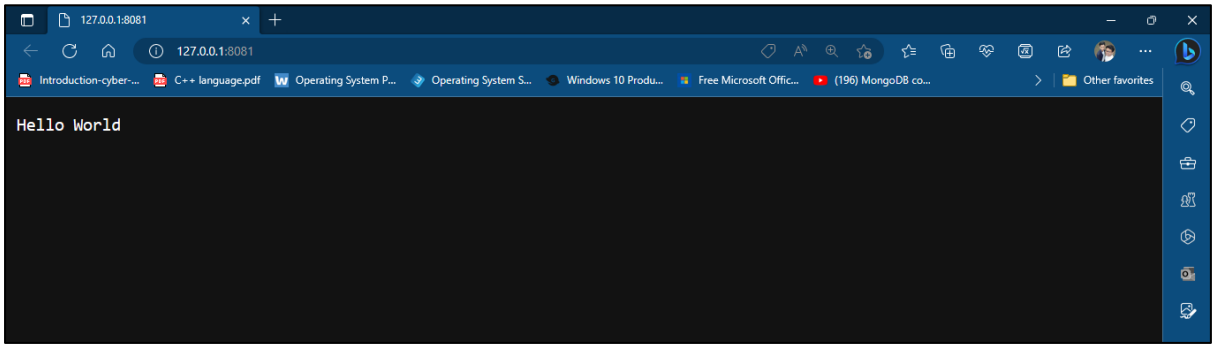
1. **Import required modules**: The "require" directive is used to load a Node.js module.
2. **Create server**: You have to establish a server which will listen to client's request similar to Apache HTTP Server.
3. **Read request and return response:** Server created in the second step will read HTTP request made by client which can be a browser or console and return the response

Input:



Output:





Conclusion: - We Successfully implement Installation of NodeJS in windows, small code for programs like Hello World.

DOP: / /2023 **DOS: / /2023**

Experiment No: 0

Title: Modules in Node.js (Networking, File system, Web module).

Theory:

Networking Module:

Node.js Net

Node.js provides the ability to perform socket programming. We can create chat application or communicate client and server applications using socket programming in Node.js. The Node.js net module contains functions for creating both servers and clients.

Node.js Net Example

In this example, we are using two command prompts:

1. Node.js command prompt for server.
2. Window's default command prompt for client.

Syntax: var net = require('net');

server:

File: net_server.js

```
const net = require('net');
var server = net.createServer((socket) => {
  socket.end('goodbye\n');
}).on('error', (err) => {
  // handle errors here
  throw err;
});
// grab a random port.
server.listen(() => {
  address = server.address();
  console.log('opened server on %j', address);
});
```

client:

File: net_client.js

```
const net = require('net');
const client = net.connect({port: 50302}, () => {//use same port of server
  console.log('connected to server!');
  client.write('world!\r\n');
});
client.on('data', (data) => {
  console.log(data.toString());
  client.end();
});
client.on('end', () => {
  console.log('disconnected from server');
});
```

Node.js File System (FS)

In Node.js, file I/O is provided by simple wrappers around standard POSIX functions. Node File System (fs) module can be imported using following syntax:

```
Syntax: var fs = require("fs")
```

Node.js FS Reading File

Every method in fs module has synchronous and asynchronous forms.

Asynchronous methods take a last parameter as completion function callback. Asynchronous method is preferred over synchronous method because it never blocks the program execution where as the synchronous method blocks.

Let's take an example:

Create a text file named "input.txt" having the following content.

File: input.txt

```
Hello NodeJS
```

Let's take an example to create a JavaScript file named "main.js" having the following code:

File: main.js

```
var fs = require("fs");
// Asynchronous read
fs.readFile('input.txt', function (err, data) {
  if (err) {
     return console.error(err);
  }
  console.log("Asynchronous read: " + data.toString());
});
// Synchronous read
var data = fs.readFileSync('input.txt');
console.log("Synchronous read: " + data.toString());
console.log("Program Ended");
```

Node.js Open a file

Syntax :fs.open(path, flags[, mode], callback)

Parameter explanation:

Following is the description of parameters used in the above syntax:

- **path**: This is a string having file name including path.
- **flags**: Flag specifies the behavior of the file to be opened. All possible values have been mentioned below.
- **mode**: This sets the file mode (permission and sticky bits), but only if the file was created. It defaults to 0666, readable and writeable.
- **callback**: This is the callback function which gets two arguments (err, fd).

Node.js Web Module :

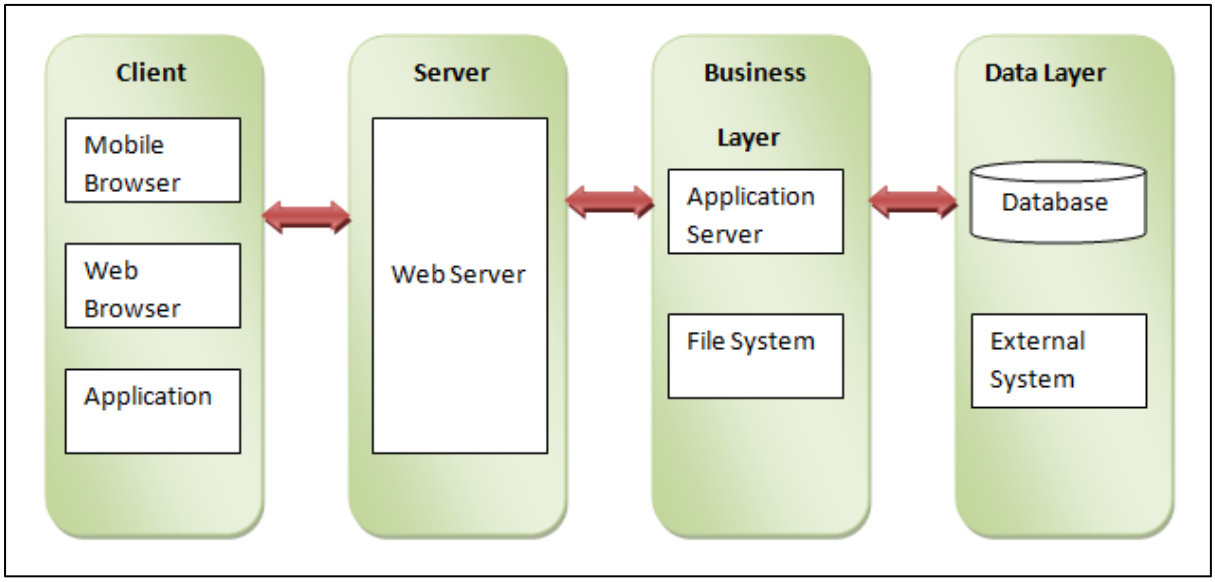
What is Web Server:

Web Server is a software program that handles HTTTP requests sent by HTTP clients like web browsers, and returns web pages in response to the clients. Web servers usually respond with html documents along with images, style sheets and scripts.

Web Application Architecture

A web application can be divided in 4 layers:

- **Client Layer**: The Client layer contains web browsers, mobile browsers or applications which can make HTTP request to the web server.
- **Server Layer:** The Server layer contains Web server which can intercepts the request made by clients and pass them the response.
- **Business Layer**: The business layer contains application server which is utilized by web server to do required processing. This layer interacts with data layer via data base or some external programs.
- **Data Layer**: The Data layer contains databases or any source of data.



Creating Web Server using Node.js syntax:

```
1. var http = require('http');
2. var fs = require('fs');
3. var url = require('url');
4. // Create a server
5. http.createServer( function (request, response) {
6.    }
```

Conclusion: - We Successfully implement and understanding Modules in Node.js