NODE JS

Aruna S
Department of
Computer Science and Engineering

NODE JS

Module System

S. Aruna
Department of Computer Science and Engineering

NODE JS

Node JS Modules

- https://nodejs.org/dist/latest-v12.x/docs/api/

  To see all the available modules

- Few modules are inbuilt globally available.

  Ex: Console module, Timer Module

- Many modules need to be explicitly included in our application

  Ex: File System module

Such modules need to be required at first in the application

NODE JS

Node JS Timer Module

- This module provides a way for functions to be called later at a given time.
- The Timer object is a global object in Node.js, and it is not necessary to import it

Method	Description
clearImmediate()	Cancels an Immediate object
clearInterval()	Cancels an Interval object
clearTimeout()	Cancels a Timeout object
ref()	Makes the Timeout object active. Will only have an effect if the Timeout.unref() method has been called to make the Timeout object inactive.
setImmediate()	Executes a given function immediately.
setInterval()	Executes a given function at every given milliseconds
setTimeout()	Executes a given function after a given time (in milliseconds)
unref()	Stops the Timeout object from remaining active

NODE JS

NPM and installing modules

- NPM is a package manager for Node.js packages, or modules if you like.
- www.npmjs.com hosts thousands of free packages to download and use.
- The NPM program is installed on your computer when you install Node.js

  A package in Node.js contains all the files you need for a module.

  Modules are JavaScript libraries you can include in your project.

Example:

```
D:\nodejs>npm install validator
```

- validator package is downloaded and installed. NPM creates a folder named "node_modules", where the package will be placed.
- To include a module, use the **require()** function with the name of the module

```
var val = require('validator');
```

NODE JS

Importing your own modules

- The module.exports is a special object which is included in every JavaScript file in the Node.js application by default.
- The module is a variable that represents the current module, and exports is an object that will be exposed as a module.
- So, whatever you assign to module.exports will be exposed as a module. It can be
  - Export Literals
  - Export Objects
  - Export Functions
  - Export Function as a class

NODE JS

Importing your own modules

You can create your own modules, and easily include them in your applications. The following example creates a module that returns a date and

```
exports.myDateTime = function () {
    return Date();
};
```

Use the exports keyword to make properties and methods available outside the module file.

```
var date = require('./myfirstmodule.js');
console.log(date.myDateTime());
```

```
PS C:\Users\DELL\Desktop\WTII\Node Examples\notes-app> node app.js
Sat Sep 12 2020 12:03:34 GMT+0530 (India Standard Time)
```

NODE JS

Package.json file

- All npm packages contain a file, usually in the project root, called package.json
- This file holds various metadata relevant to the project.
- It is used to give information to npm that allows it to identify the project as well as handle the project's dependencies.
- It can also contain other metadata such as a project description, the version of the project in a particular distribution, license information, even configuration data - all of which can be vital to both npm and to the end users of the package.
- The package.json file is normally located at the root directory of a Node.js project.

NODE JS

A Sample Package.json file

```
{
    "name": "sample",
    "version": "1.0.0",
    "description": "Learning Express",
    "main": "index.js",
    "dependencies": {
      "body-parser": "^1.19.0",
      "builtin-modules": "^3.1.0",
      "express": "^4.17.1",
      "mongodb": "^3.6.1",
      "npm": "^6.14.6"
    },
    "devDependencies": {},
    "scripts": {
      "test": "hi"
    },
    "author": "Aruna",
    "license": "ISC"
  }
```

NODE JS

CHALK and nodemon Module

https://www.npmjs.com/package/chalk

- The Chalk Module Can Be Used With The Console's String Interpolation in Node. Js.
- The Chalk module allows you to add styles to your terminal output.

https://www.npmjs.com/package/nodemon

- **Nodemon** is a utility that will monitor for any changes in your source and automatically restart your server.
- Just use **nodemon** instead of node to run your code

THANK YOU

Aruna S
Department of
Computer Science and Engineering
arunas@pes.edu