Node.Js

10 April 2019 04:28 PM

==========
==========
https://www.w3schools.com/nodejs/nodejs_intro.asp
What is Node.Js?

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server
- Node.js uses **asynchronous** programming!
- Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

==========
What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

==========
Git Repo https://github.com/harsh007kumar/NodeJS
==== About Node.js ==========

> Download **NodeJS** from https://nodejs.org/en/
  - [LTS ver- Long Term Support] codenamed after elements in periodic table (Ex-Carbon)
  - Direct Link https://nodejs.org/en/download/releases/ (ver **Erbium**)
> **npm** - Node package management
  - https://www.npmjs.com
  - (Like NuGet gives assemblies for .NET) it allows you to download packages from command line
> Use below command to check node & npm version installed on your machine :

```
C:\Users\harkuma>node -v
v12.16.2

C:\Users\harkuma>npm -v
6.14.4
```

==== Package Management ==========

> How to Install an package
  - Ex- "Node.js web-server"
  - | Use => | npm install express |
  - Here <express> is name of the package, replace it with whichever package you want to install.
> To get details of about any package (ex- what its used for)
  - Use https://www.npmjs.com/package/fresh
    Here <fresh> is package name, replace it with any package name you want to get detail about.
> "package-lock.json" is like timeline of everything installed on project.
  - Meant only for Developers and end user does see this file as it isn't published.
> Some modules include executables (Ex- Start-up script to run in web-server)
  - Global installed executables added to path
    - Should be installed "-g" flag.
    - If user is going to run program directly from command line, they should install it globally.
  - Locally installed
    - Ex- Build command which execute a minifier script.
    - Installed in node_modules/.bin directory (so it could be referenced relatively)
> CreatingYourOwnPackage
  - Simple create a blank folder and follow below steps after running **npm init**

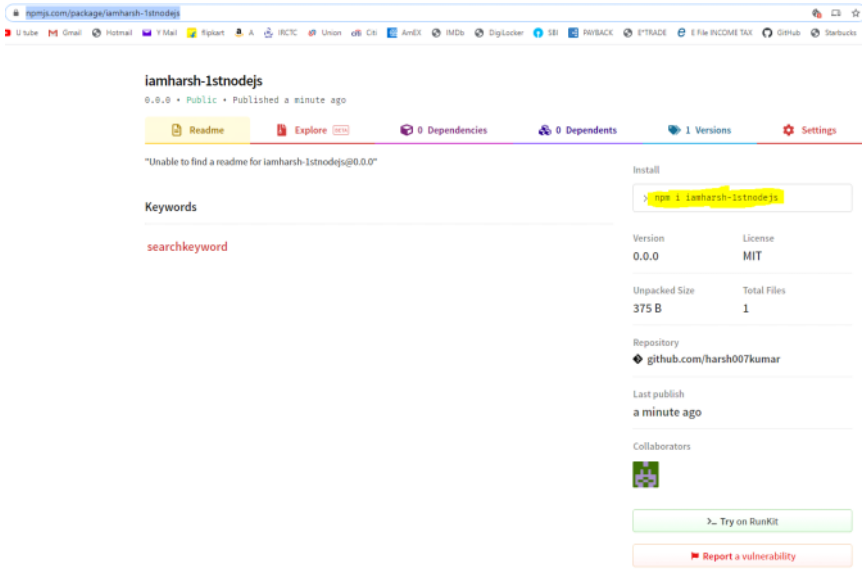
```
Select Node.js command prompt
C:\Users\harkuma\source\repos\NodeJS>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs) iamharsh-1stnodejs
version: (1.0.0) 0.0.0
description: creating first Node.js package
entry point: (index.js)
test command: testthisplease
git repository: https://github.com/harsh007kumar
keywords: searchkeyword
author: Harsh<harsh007kumar@gmail.com>
license: (ISC) MIT
About to write to C:\Users\harkuma\source\repos\NodeJS\package.json:

{
  "name": "iamharsh-1stnodejs",
  "version": "0.0.0",
  "description": "creating first Node.js package",
  "main": "index.js",
  "scripts": {
    "test": "testthisplease"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/harsh007kumar"
  },
  "keywords": [
    "searchkeyword"
  ],
  "author": "Harsh <harsh007kumar@gmail.com>",
  "license": "MIT"
}


Is this OK? (yes) yes

C:\Users\harkuma\source\repos\NodeJS>
```

> Publishing your package

- **Npm login**

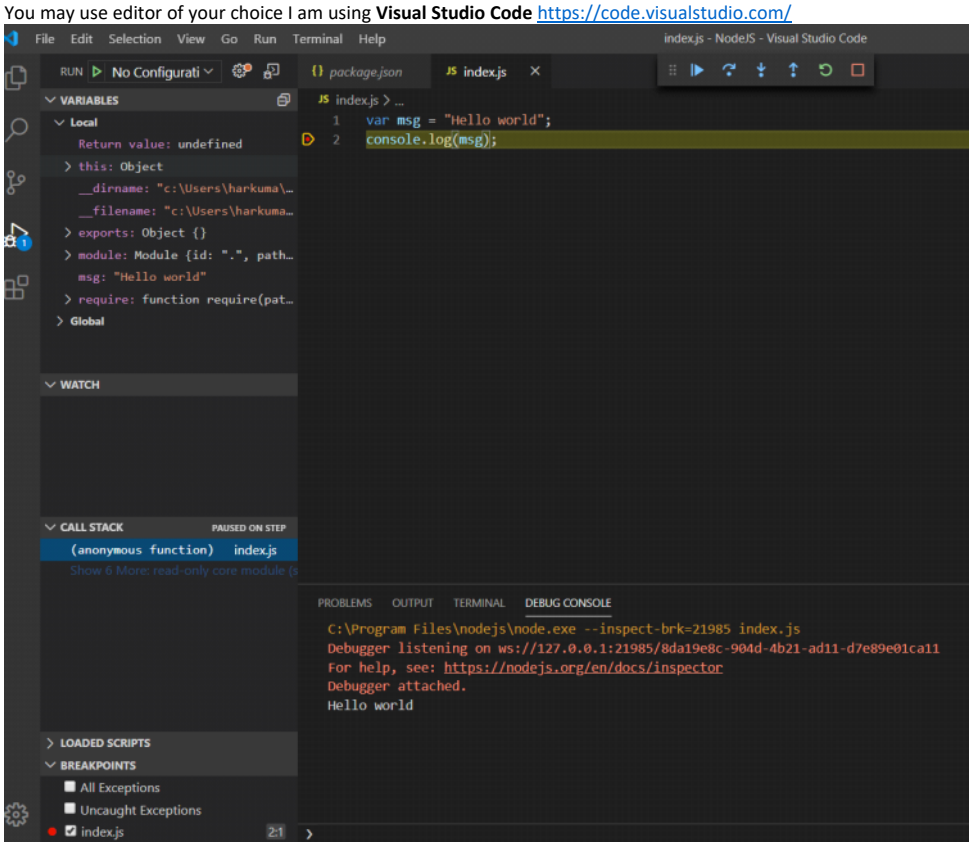
```
C:\Users\harkuma\source\repos\NodeJS>npm whoami
npm ERR! code ENEEDAUTH
npm ERR! need auth This command requires you to be logged in.
npm ERR! need auth You need to authorize this machine using `npm adduser`

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\harkuma\AppData\Roaming\npm-cache\_logs\2020-04-23T19_49_12_010Z-debug.log

C:\Users\harkuma\source\repos\NodeJS>npm login
Username: harsh007kumar
Password:
Email: (this IS public) harsh007kumar@gmail.com
Logged in as harsh007kumar on https://registry.npmjs.org/.

C:\Users\harkuma\source\repos\NodeJS>npm publish
npm notice
npm notice package: iamharsh-1stnodejs@0.0.0
npm notice === Tarball Contents ===
npm notice 375B package.json
npm notice === Tarball Details ===
npm notice name:          iamharsh-1stnodejs
npm notice version:       0.0.0
npm notice package size:  324 B
npm notice unpacked size: 375 B
npm notice shasum:        dc891ae045b9276a9f776ffcb7ae5a40be4231ea
npm notice integrity:     sha512-6yTcb5UcLM2cw[...]TodO7n5n5RTsg==
npm notice total files:   1
npm notice
+ iamharsh-1stnodejs@0.0.0

C:\Users\harkuma\source\repos\NodeJS>
```

- **Npm publish**
- To Check which id you are logged on use **npm whoami**
- To check where your package is published use earlier mentioned format
  - https://www.npmjs.com/package/<yourPackageName>



==== JavaScript fundamentals ==========

You may use editor of your choice I am using **Visual Studio Code** https://code.visualstudio.com/



- We have variables on the top left
- Call Stack on bottom left
- And Console on center bottom of screen
- To add debug point use single click
- F5 to run the program

> **Defining prototypes and classes in Node.js**
  - Node.js supports mostly ES6 ES2015 convention

```
package.json   index.js   helloWorld.js
index.js > HelloWorld3 > sayHello
1   /////////////// Example 1
2   var msg = "Hello world";
3   console.log(msg);
4
5   /////////////// Example 2
6   sayHello();
7   function sayHello()
8   {
9       console.log("Hello World 2 - standalone function");
10  }
11  /////////////// Example 3
12  function HelloWorld3(){}
13  HelloWorld3.prototype.sayHello = function()
14  {
15      console.log("Hello World 3 - function part of prototype");
16  }
17  new HelloWorld3().sayHello();
18  /////////////// Example 4
19  class HelloWorld
20  {
21      sayHello()
22      {
23          console.log("Hellow World 4");
24      }
25  }
26  new HelloWorld().sayHello();
27  /////////////// Example 5
28  const meaningLessName= require('./helloWorld');     // assiging value to constant using require|| We dont mentioned extension of the file
29  new meaningLessName().sayHello();

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE
C:\Program Files\nodejs\node.exe --inspect-brk=33059 index.js
Debugger listening on ws://127.0.0.1:33059/5823824e-3e10-49fc-85d0-5ab0c30dfba7
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello world
Hello World 2 - standalone function
Hello World 3 - function part of prototype
Hellow World 4
Hellow World 5 - from class present declared in helloWorld.js file
```

> **Using Babel to trans pile from ES6 (ES2015)**
  - **npm install --save-dev babel-cli**
  - **npm install --save-dev babel-preset-es2015**

    Note : --save-dev signifies it is only required by the developer working on this platform.
  - Now update package.json to include **bable and inform that we would be writing code in ES2015**

>

```
EXPLORER
OPEN EDITORS
  package.json
  index.js src
  helloWorld.js src
NODEJS
  .compiled
  node_modules
  src
    helloWorld.js
    index.js
  package-lock.json
  package.json

package.json   index.js   helloWorld.js
package.json > {} scripts > compile
{
  "name": "iamharsh-1stnodejs",
  "version": "0.0.0",
  "description": "creating first Node.js package",
  "main": "index.js",
  "scripts": {
    "test": "testthisplease",
    "compile":"./node_modules/.bin/babel src --out-dir .compiled --source-maps --watch"
  },
  "babel":{
    "presets":[
      "es2015"
    ]
  },
  "repository": {
    "type": "git",
    "uri": "https://github.com/harsh007kumar"
```

  - We also add below under scripts :
    - `"compile":"./node_module/.bin/babel src --out-dir .compiled --source-maps --watch"`
    - `//watch informs to keep running the code in background and every time code is changed its silently trans-piled/recompiled`
  - Now we can run below from command line which will keep compiling our code
    - npm run compile

```
C:\Users\harkuma\source\repos\NodeJS>npm run compile

> iamharsh-1stnodejs@0.0.0 compile C:\Users\harkuma\source\repos\NodeJS
> babel src --out-dir .compiled --source-maps --watch

src\helloWorld.js -> .compiled\helloWorld.js
src\index.js -> .compiled\index.js
```

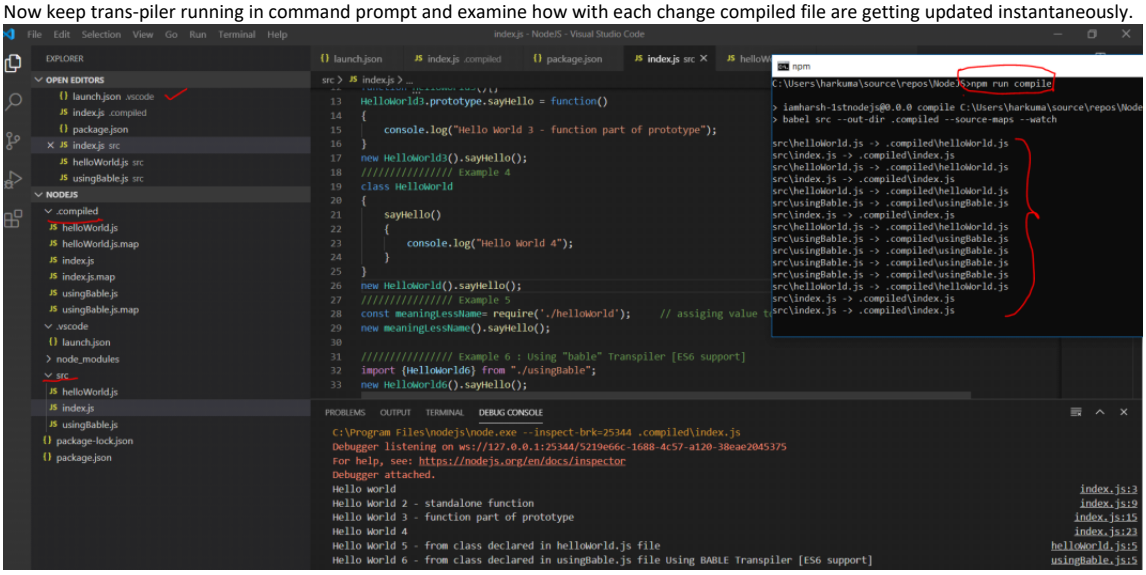
  - To check trans-piled files check files under ./compiled folder
  - Now if you will run this Code from Editor it will throw an error, because we need to tell VS Code in this case how to actually run our code, by making below entry in launch.json file

```
"program": "${workspaceFolder}\\src\\index.js",
"outFiles": ["${workspaceFolder}/.compiled/**/*.js"]
```

launch.json

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "node",
            "request": "launch",
            "name": "Launch Program",
            "skipFiles": [
                "<node_internals>/**"
            ],
            "program": "${workspaceFolder}\\src\\index.js",
            "outFiles": ["${workspaceFolder}/.compiled/**/*.js"]
        }
    ]
}
```

- Now keep trans-piler running in command prompt and examine how with each change compiled file are getting updated instantaneously.



> **Asynchronous programming**

==== Introducing Express ==========

> **Introducing Express**
> **Mimicking ASP.NET Core with Express**

==== Unit Testing ==========

> **Installing Jasmine to test Node.js**
> **Mocking test classes with Jasmine**

==== Continuous Integration ==========

> **Introducing Gulp**
> **Adding support for async/await in Babel**
> **Creating a CI pipeline in Visual Studio Team Services**
> **Deploying our web app to Azure**

==========
==========
==========
==========
==========

==========