NPM Node js

#Node JS Notes

Node Package Manager

What is NPM ?

- NPM Means Node Package Manager.
- The **NPM** program is installed on your computer when you install **Node.js**.
- **NPM** install packages in two modes local and global.

Version Check

- npm -v
- npm -version

What is use of NPM ?

- **npm** makes it easy **for JavaScript** developers to share and reuse code, and it makes it easy to update the code that you're sharing.
- Just like php has **composer for** downloading decencies which are just code in files.
- To Find Various NPM Packages visit www.npmjs.com

Command

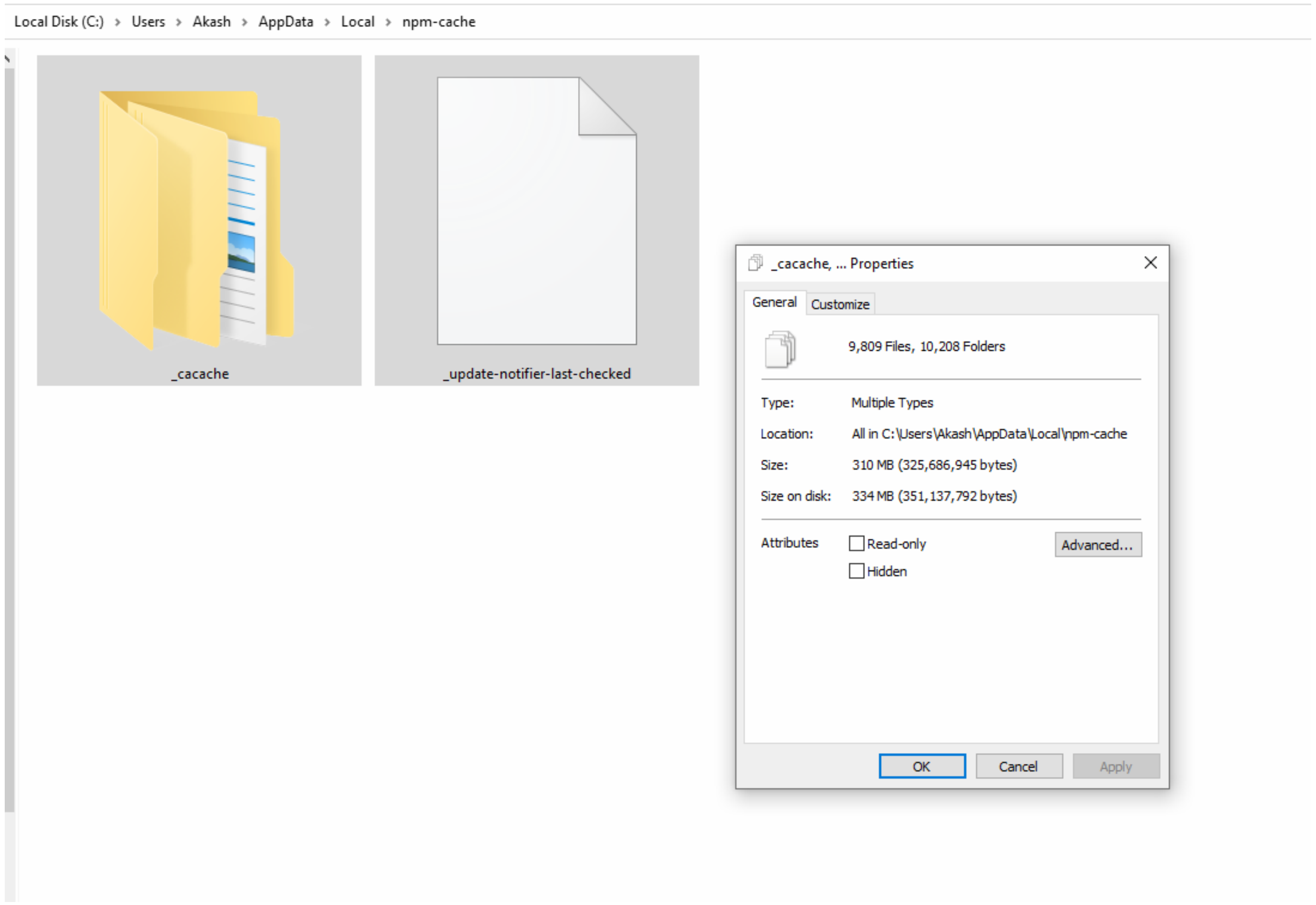
- How to Install/Upgrade NPM
  - npm install -g npm
  - npm install -g npm@7.5.0

Local vs Global

- Local – folder based module
  - In the local mode, **NPM** installs packages in the **node_module** directory of the current working directory which location is owned by current user.
  - **D:\ProjectName\node_module**

- Global – System Based Modules
  - Global packages installed in the directory where the node is installed and the location is owned by the root user
  - **C:\Users\AkashPadhiyar\AppData\Roaming\npm**
  - **Cache : C:\Users\AkashPadhiyar\AppData\Local\npm-cache**

Example


Local Disk (C:) › Users › Akash › AppData › Local › npm-cache

_cacache

_update-notifier-last-checked

_cacache, ... Properties

General | Customize

9,809 Files, 10,208 Folders

Type: Multiple Types

Location: All in C:\Users\Akash\AppData\Local\npm-cache

Size: 310 MB (325,686,945 bytes)

Size on disk: 334 MB (351,137,792 bytes)

Attributes [ ] Read-only [ ] Hidden

What is Module ?

- Each js file is a module
- Variables outside the file cannot visible except their are global or exported
- Use **module.exports** to write modules
- Use require('fs') to import fs module.

What is Package

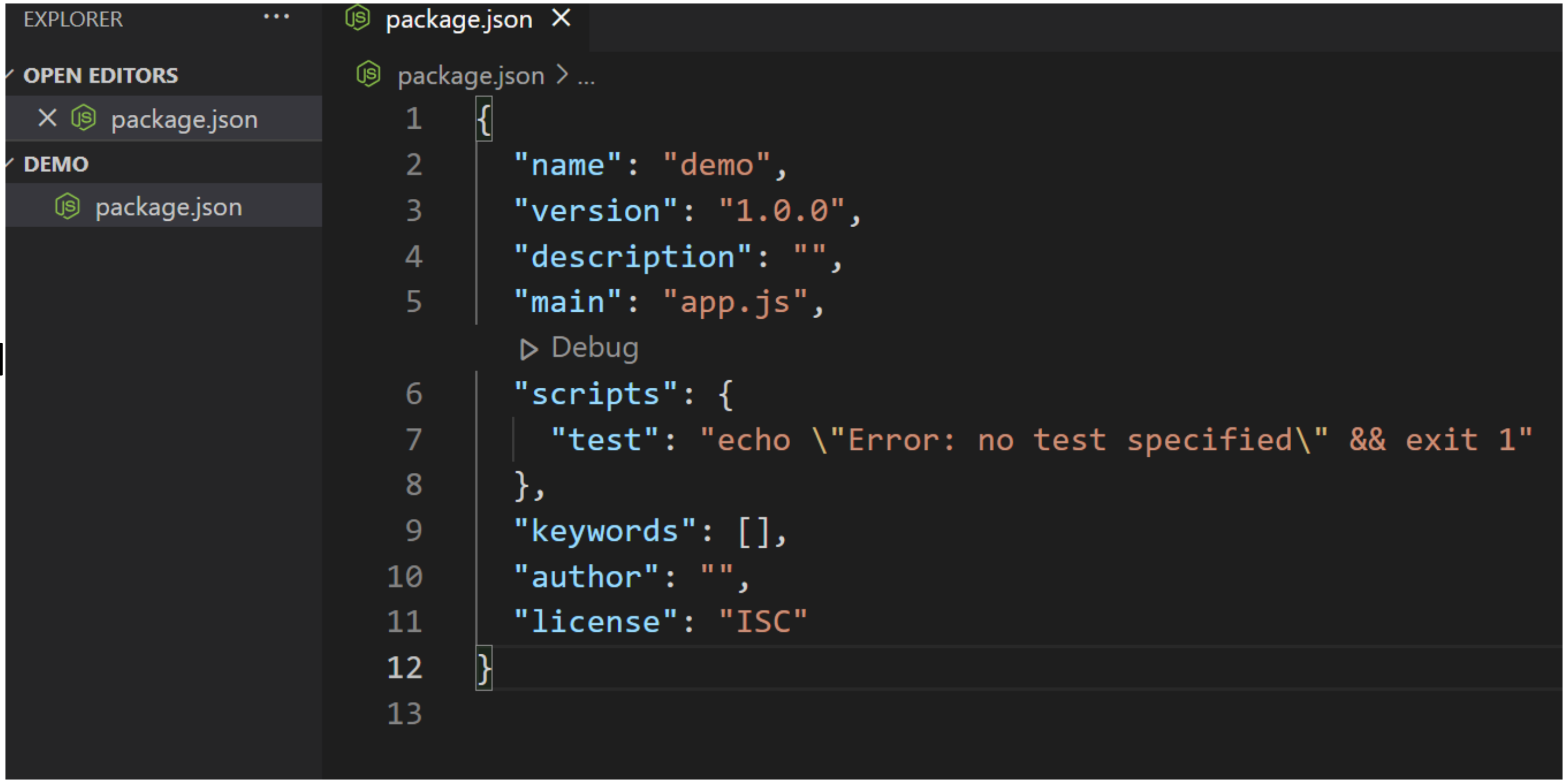
- Package is a group of a modules
- Package metadata is defined in package.json
- package.json in the root of a package dir

What is package. json?

- package.json is a plain JSON text file which manages all the packaged which you installed in your node application.
- Every Node.js applications should have this file at the root directory to describe the application metadata.
- A simple package.json file looks like below

Package.json

```
{
   "name" : “akashpadhiyar",
   "version" : "1.0.0"'
   "repository": {
      "type" : "git",
      "url" : "github_repository_url
   },
   "dependencies": {
      "async": "0.8.0",
      "express": "4.2.x"
   }
}
```



```
{
  "name": "demo",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

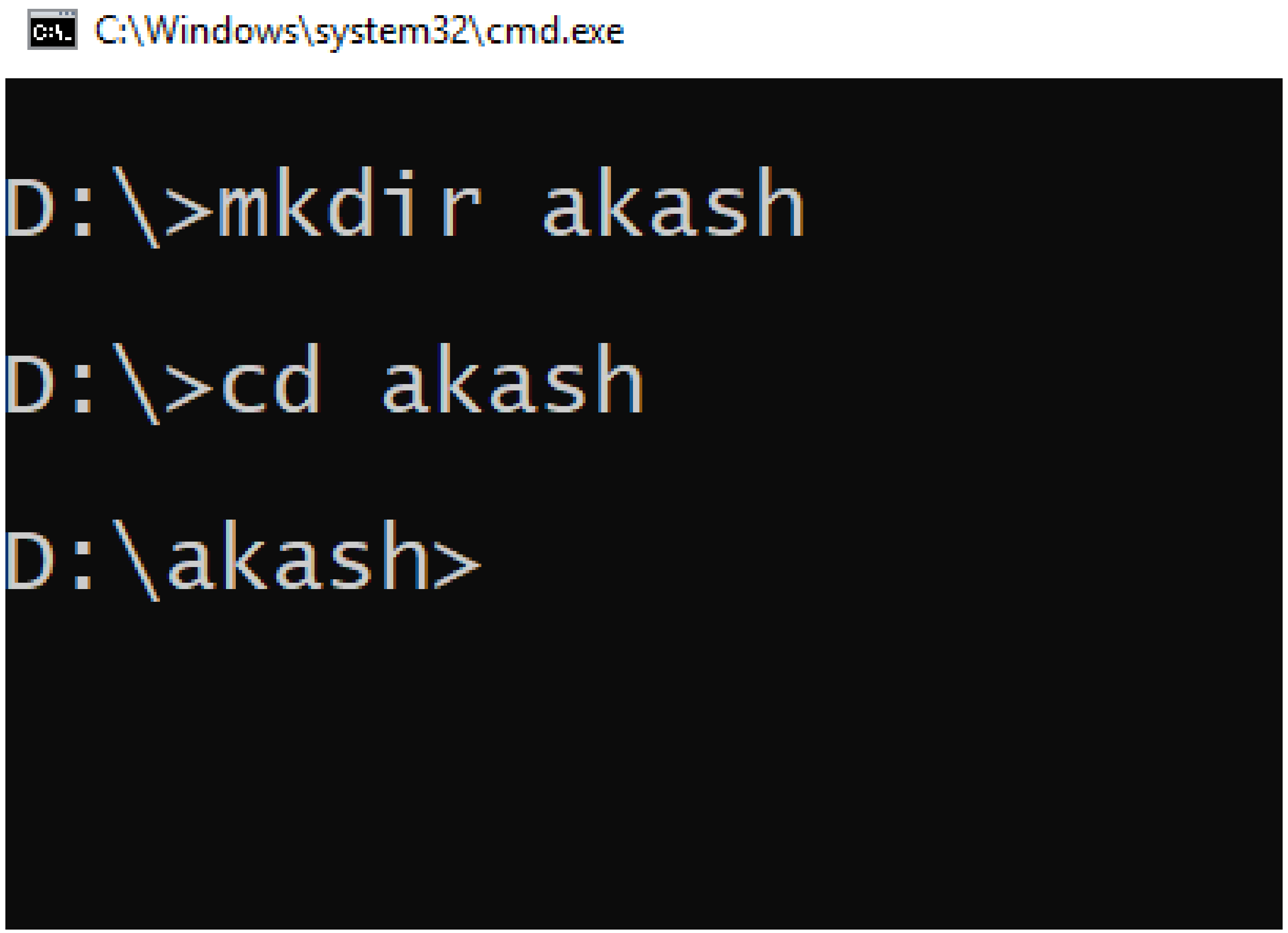
Who uses package.json file?

- NPM (Node Package Manager) uses this package.json file information about Node JS Application information or Node JS Package details.
- package.json file contains a number of different directives or elements.
- It uses these directives to tell NPM "How to handle the module or package".

How to Create package.json file?

- Create one folder.
- Navigate to Folder.
- Open Command prompt
- Type below command
  - npm init
- Provide required information

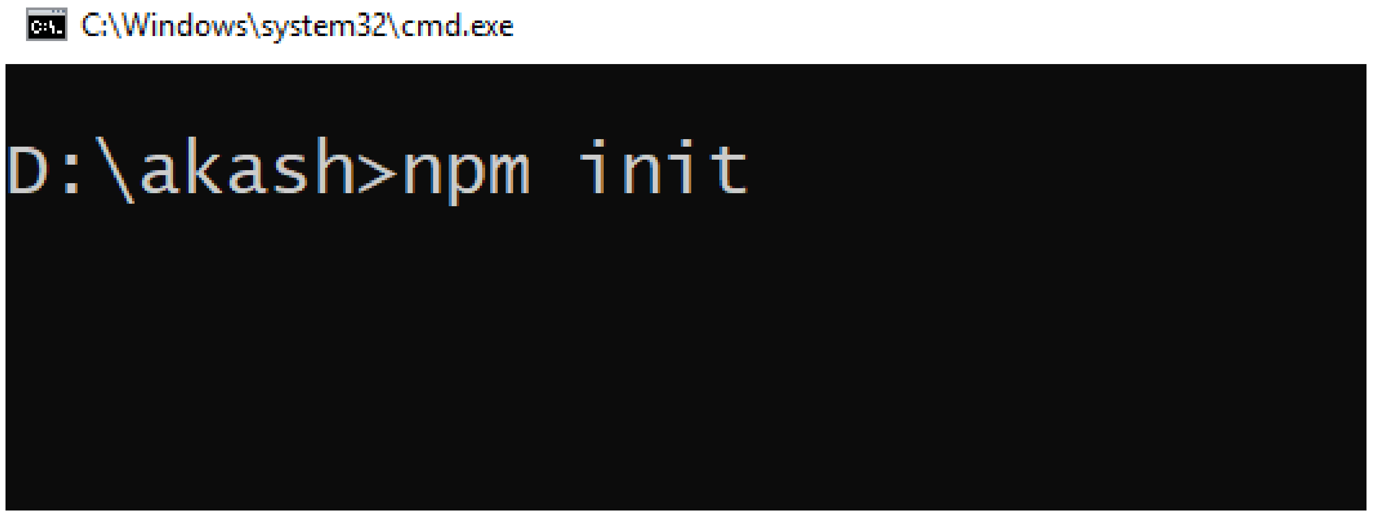
Create Folder


C:\Windows\system32\cmd.exe

```
D:\>mkdir akash

D:\>cd akash

D:\akash>
```

Npm init

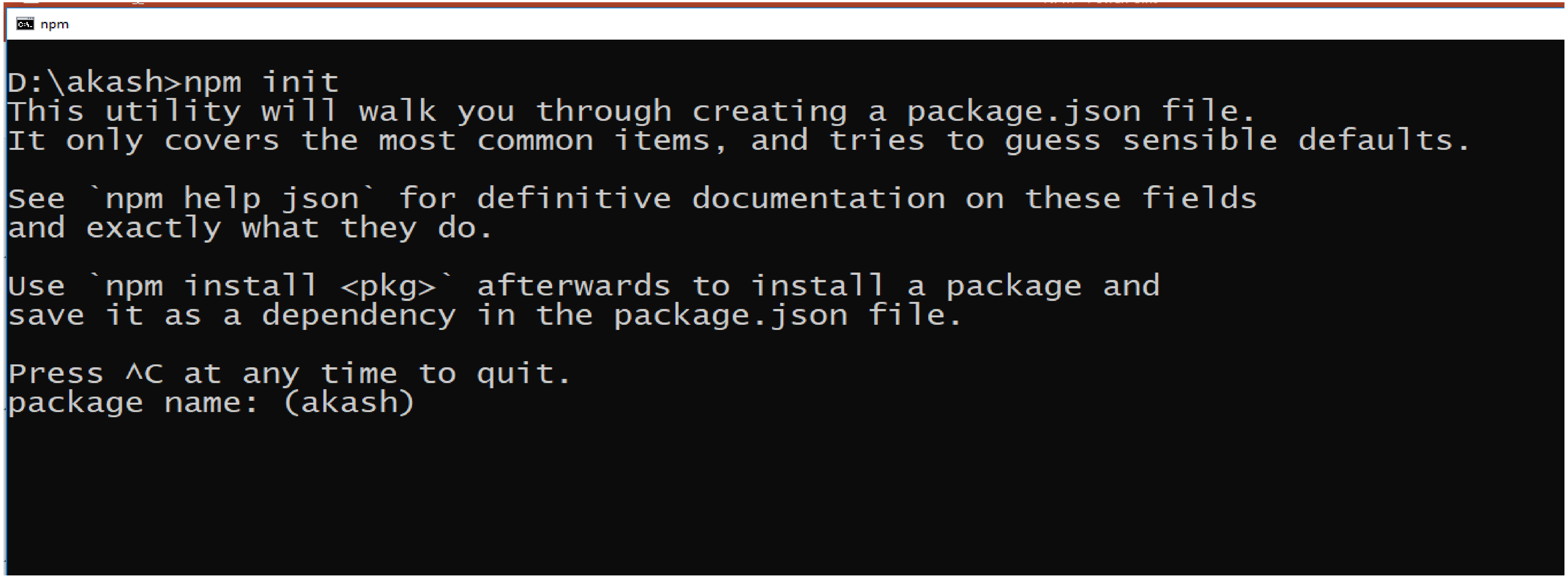
Npm init command will create package.json file.



C:\Windows\system32\cmd.exe

```
D:\akash>npm init
```

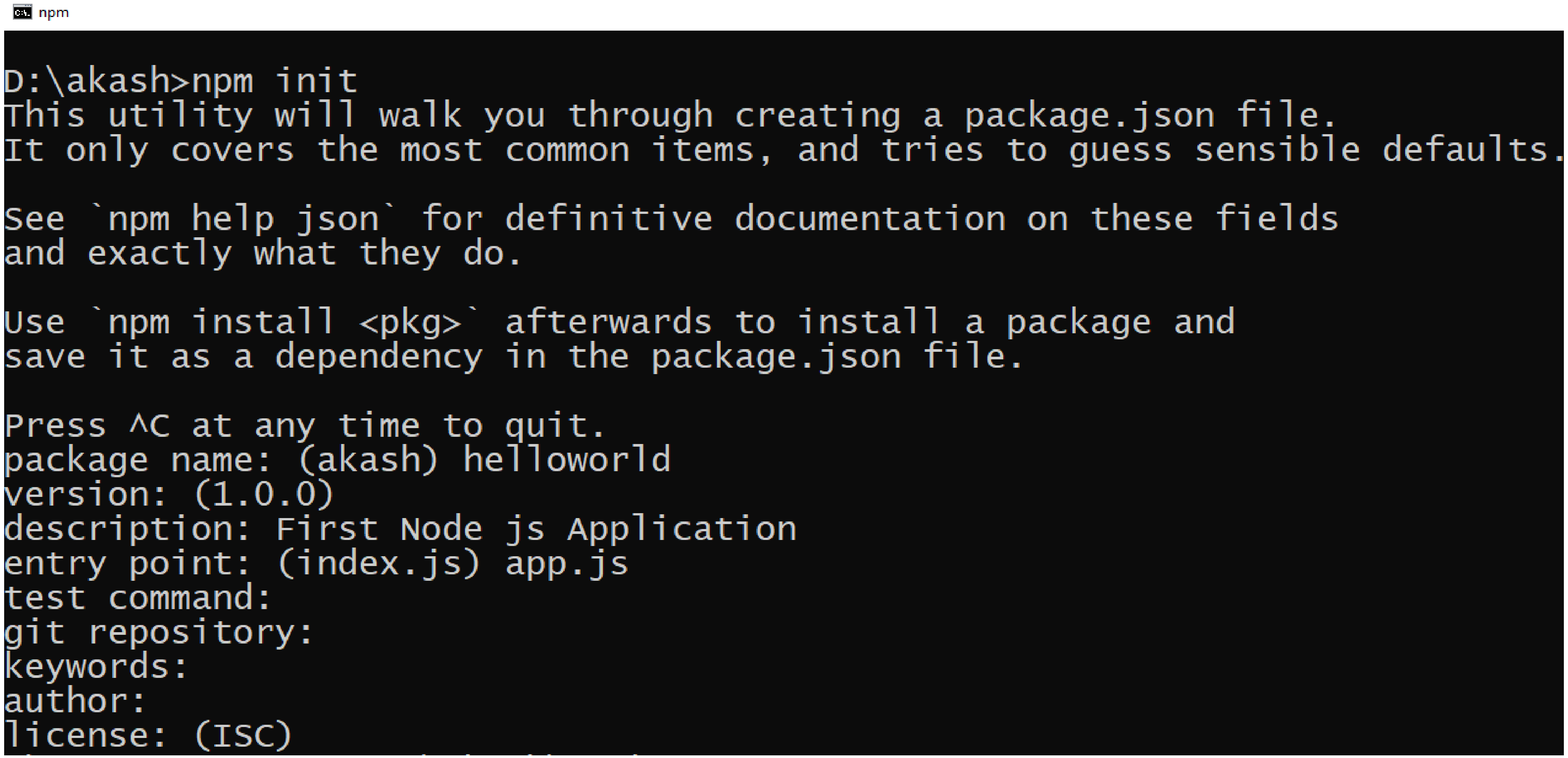
Provide Package Name



npm

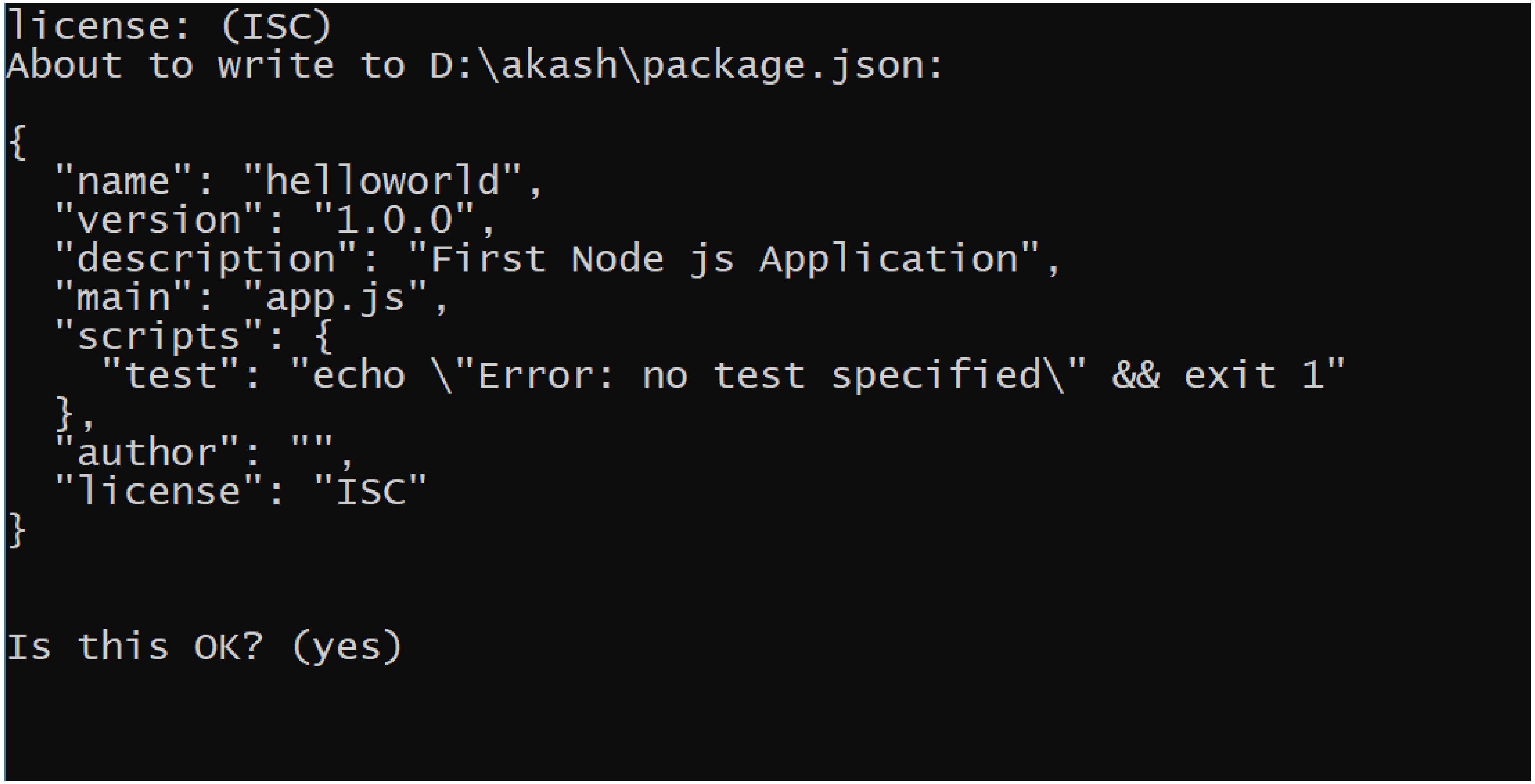
```
D:\akash>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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: (akash)
```

Provide Basic information



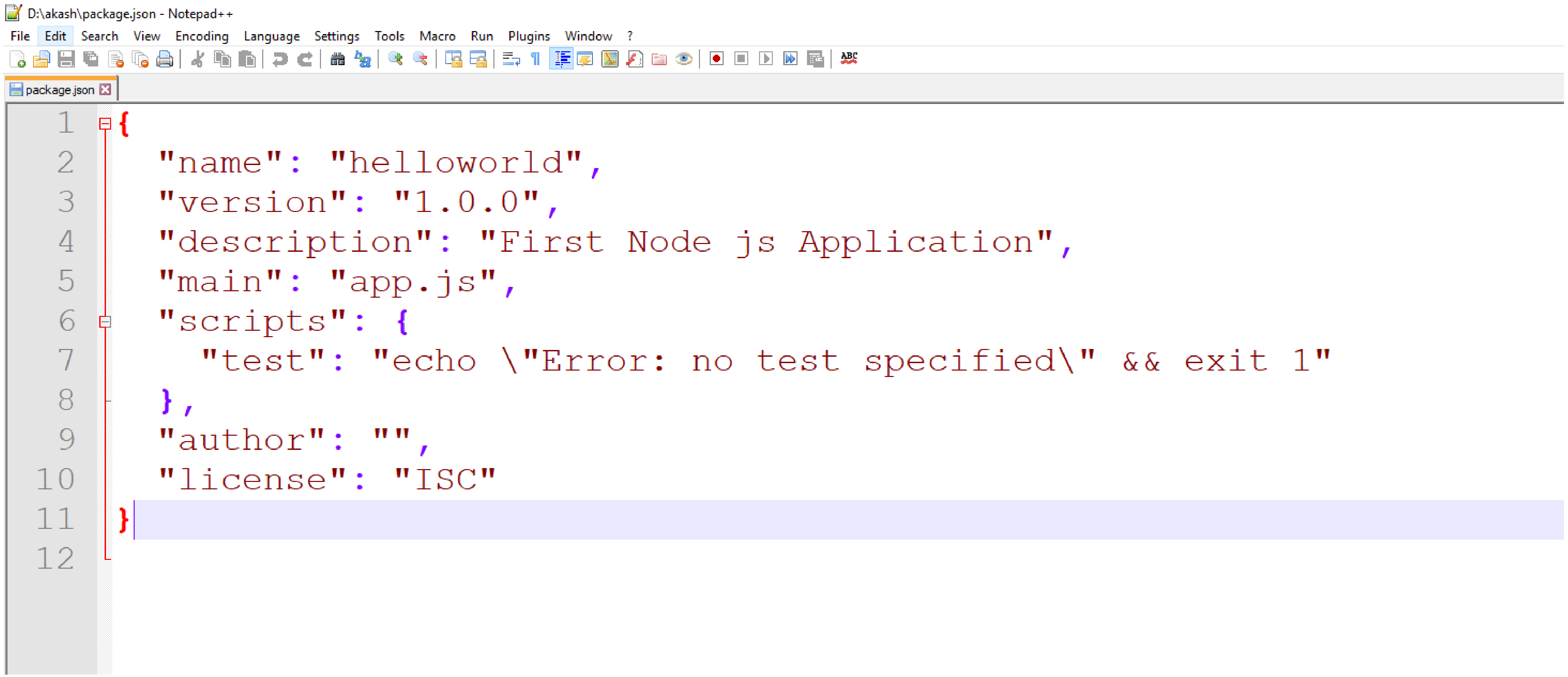
```
D:\akash>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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: (akash) helloworld
version: (1.0.0)
description: First Node js Application
entry point: (index.js) app.js
test command:
git repository:
keywords:
author:
license: (ISC)
```

Review your Details



```
license: (ISC)
About to write to D:\akash\package.json:

{
  "name": "helloworld",
  "version": "1.0.0",
  "description": "First Node js Application",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)
```

Package json file demo


D:\akash\package.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package.json

```json
{
  "name": "helloworld",
  "version": "1.0.0",
  "description": "First Node js Application",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

Mandatory Directives of Package.json

- name:
  - It is a unique Node JS Package (Module) name or our Node JS Project name.
  - This name should be in lower case letters. It is mandatory directive. Without this directive, we cannot install our Node JS Package by using NPM.

- Version
  - version is the Node JS Package version number.
  - It is a Mandatory directive in package.json file. NPM uses this version number to install or uninstall or update the right package in our NODE JS Environment.

```
{
"name" : “helloworld",
"version" : "1.0.0"
}
```

Optional Directives

- **description**:
  - It is the description of the Node JS Project or module. We need to provide brief and concise description about “What this module does”.

- **dependencies**:
  - Every Node JS Project or Module may dependent on other Node JS or Third Party Modules or our own Custom Modules. We should provide all those dependencies by using this “dependencies” directive.

Demo of Package.json file

```
{
"name" : "sampleapp",
"version" : "1.0.0"'
"dependencies": {
        "mypackage": "1.0.0",
        "async": "0.8.0",
        "express": "4.2.x"
}
}
```

Short Explanation:

- **name:** The name of the application/project.
- **version:** The version of application. The version should follow semantic versioning rules.
- **description:** The description about the application, purpose of the application, technology used like React, MongoDB, etc.
- **main:** This is the entry/starting point of the app. It specifies the main file of the application that triggers when the application starts. Application can be started using **npm start**.
- **scripts:** The scripts which needs to be included in the application to run properly.

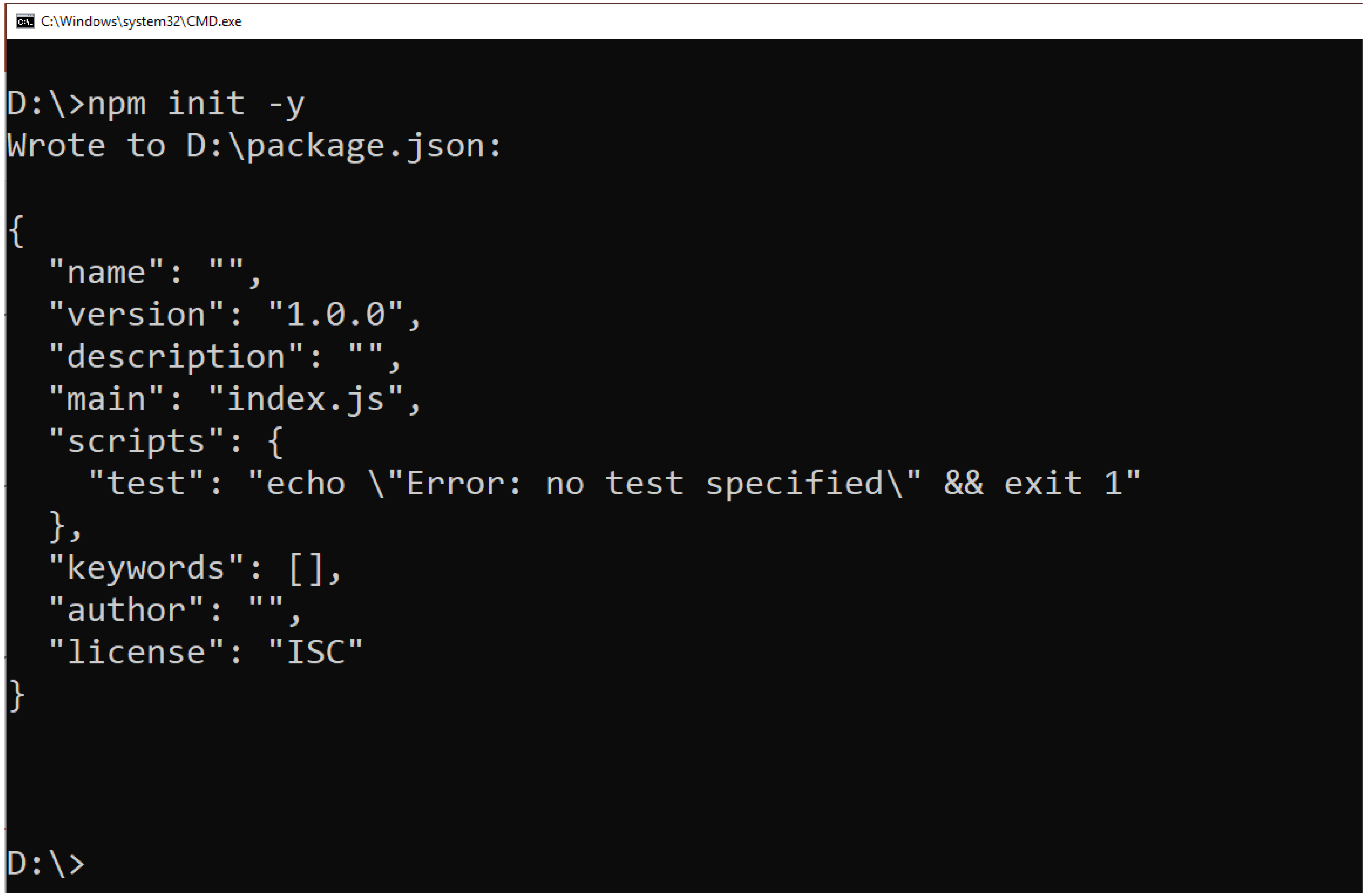
- **keywords:** It specifies the array of strings that characterizes the application.
- **author:** It consist of the information about the author like name, email and other author related information.
- **license:** The license to which the application confirms are mentioned in this key-value pair.
- **dependencies:** The third party package or modules installed using **npm** are specified in this segment.

Easy Way to Create package.json

- Npm init --yes / -y

- It will create package.json file with default values.

npm init -y



```
D:\>npm init -y
Wrote to D:\package.json:

{
  "name": "",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}


D:\>
```

#Global vs. Local Installation.

- By default, **NPM** installs any dependency in the local mode. Here local mode refers to the package installation in the **node_modules** directory inside our working project. Locally deployed packages are accessible via require() method.
- For example, when we installed an express module, it created a **node_modules** directory in the current directory where it installed the express module.

Local Example

```
$ npm install express
```

Now you can use this module in your js file as following

```
Var express = require('express');
```

Global

- Globally installed packages are stored in the system directory. Such dependencies can be used in**CLI** (Command Line Interface) function of any **node.js** but cannot be imported using **require**() in Node application directly.

- Example:
- npm install express -g

NPM Commands

Commands

- npm install <Module Name>
- Install Express :
  - npm install express
- Uninstalling a Module
  - npm uninstall express
- Updating a Module
  - npm update express
- Search a Module
  - npm search express

Global Package

- **Install Package Globally:**
  - npm install -g express
- **Npm Help**
  - npm help

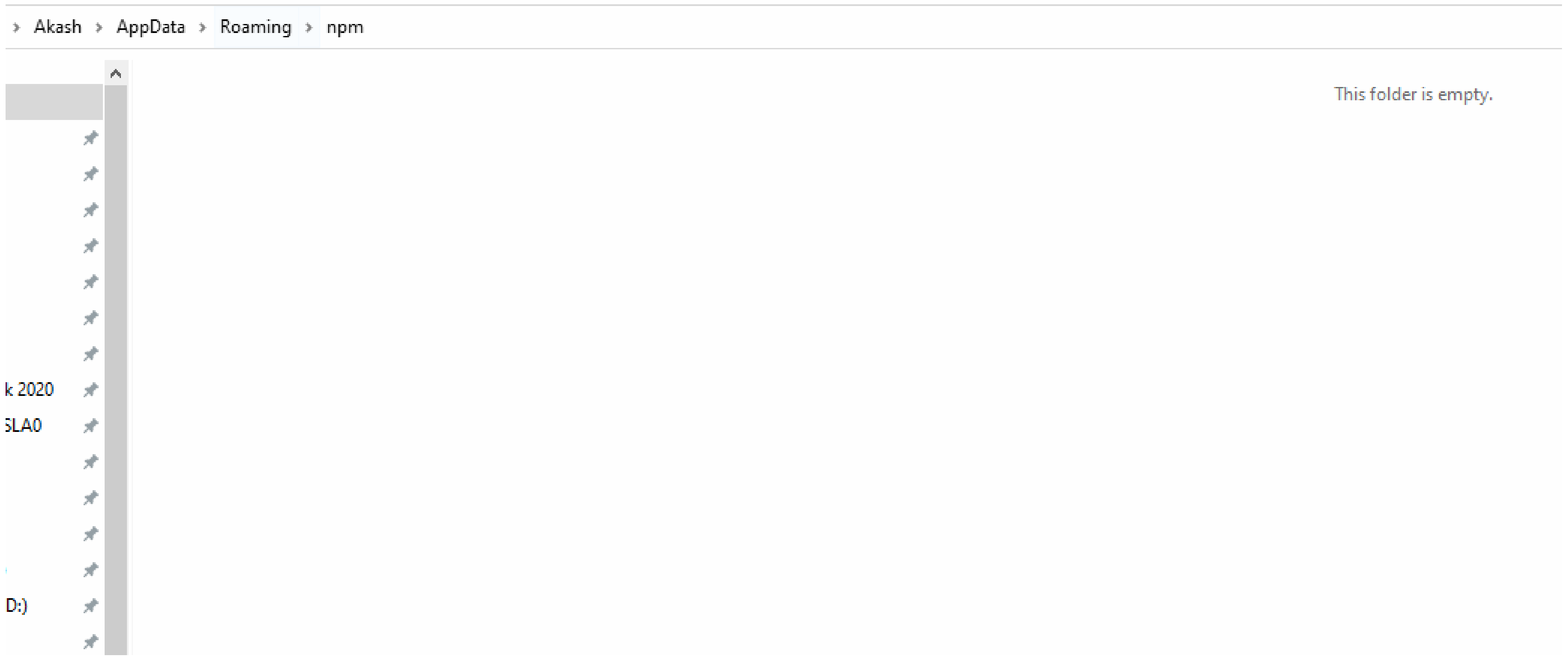
NPM installs global packages into */<User>/local/lib/node_modules* folder.

Global Package Location :

C:\Users\Akash\AppData\Roaming\npm

Global Folder Location

- C:\Users\Akash\AppData\Roaming\npm


› Akash › AppData › Roaming › npm

This folder is empty.

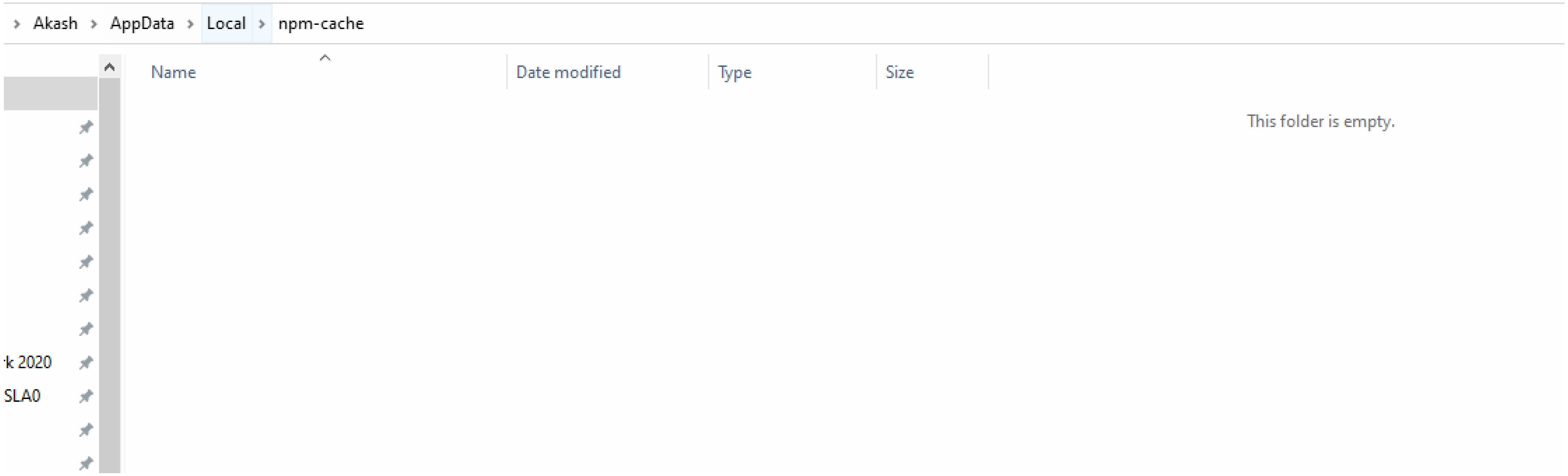
k 2020

5LA0

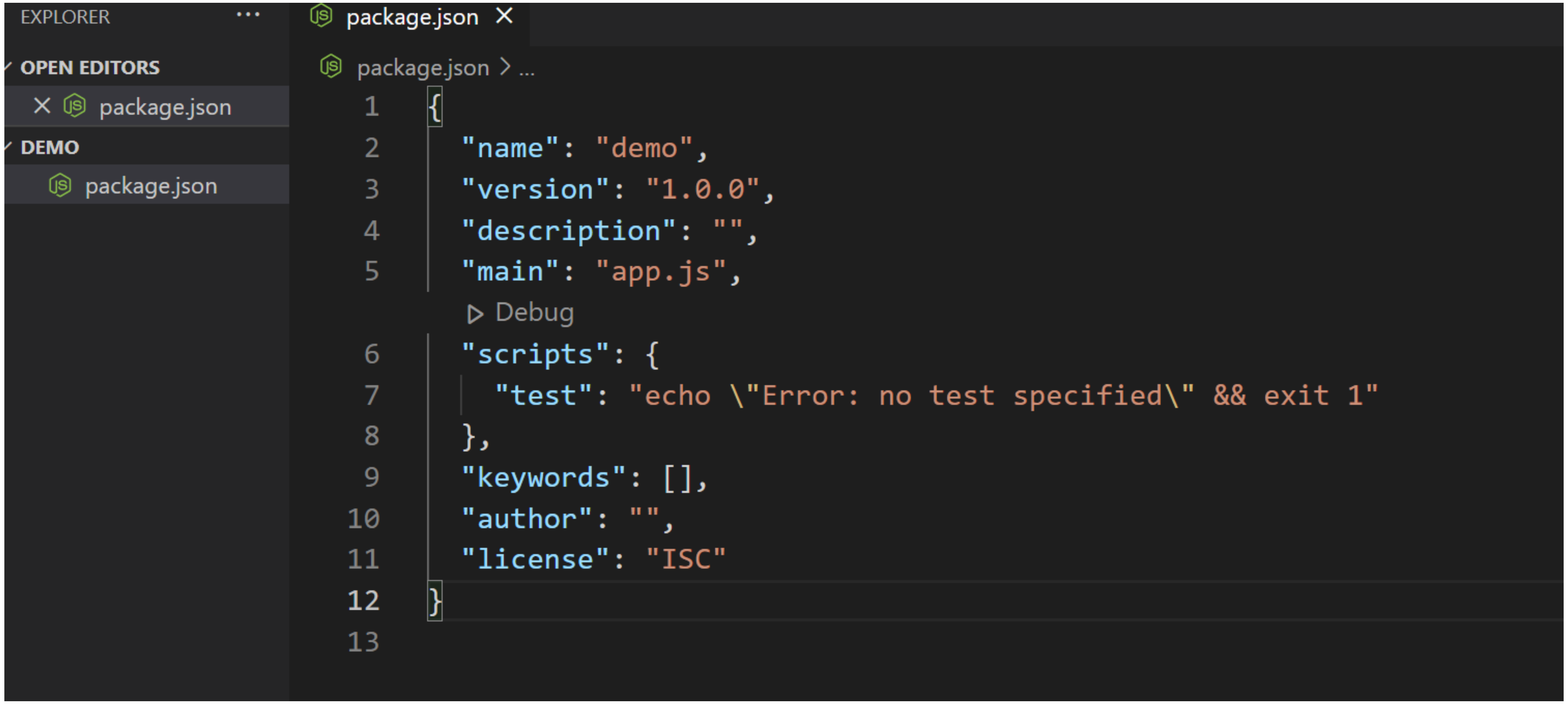
D:)

Global File Location of Cache

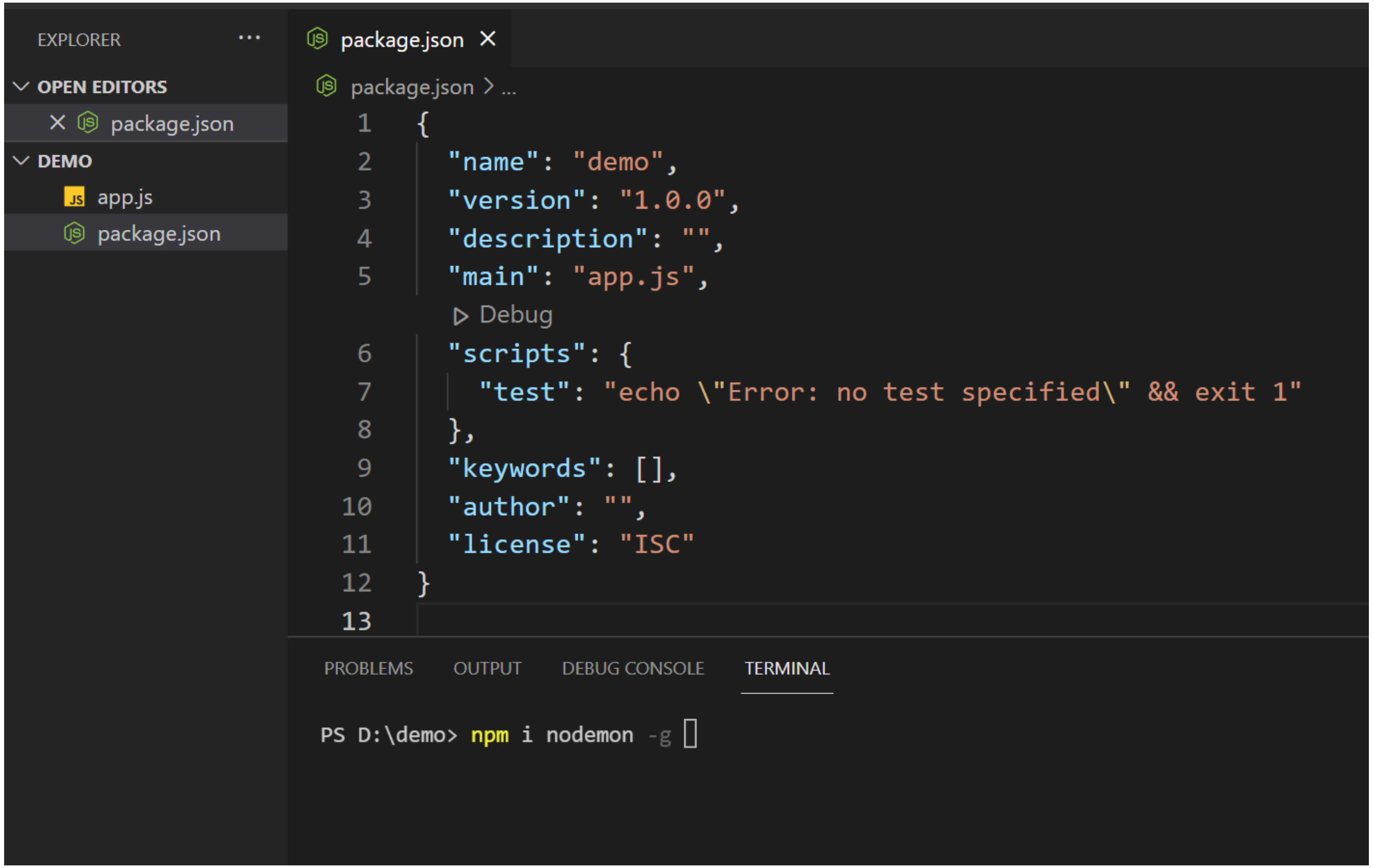
- C:\Users\Akash\AppData\Local\npm-cache



This folder is empty.

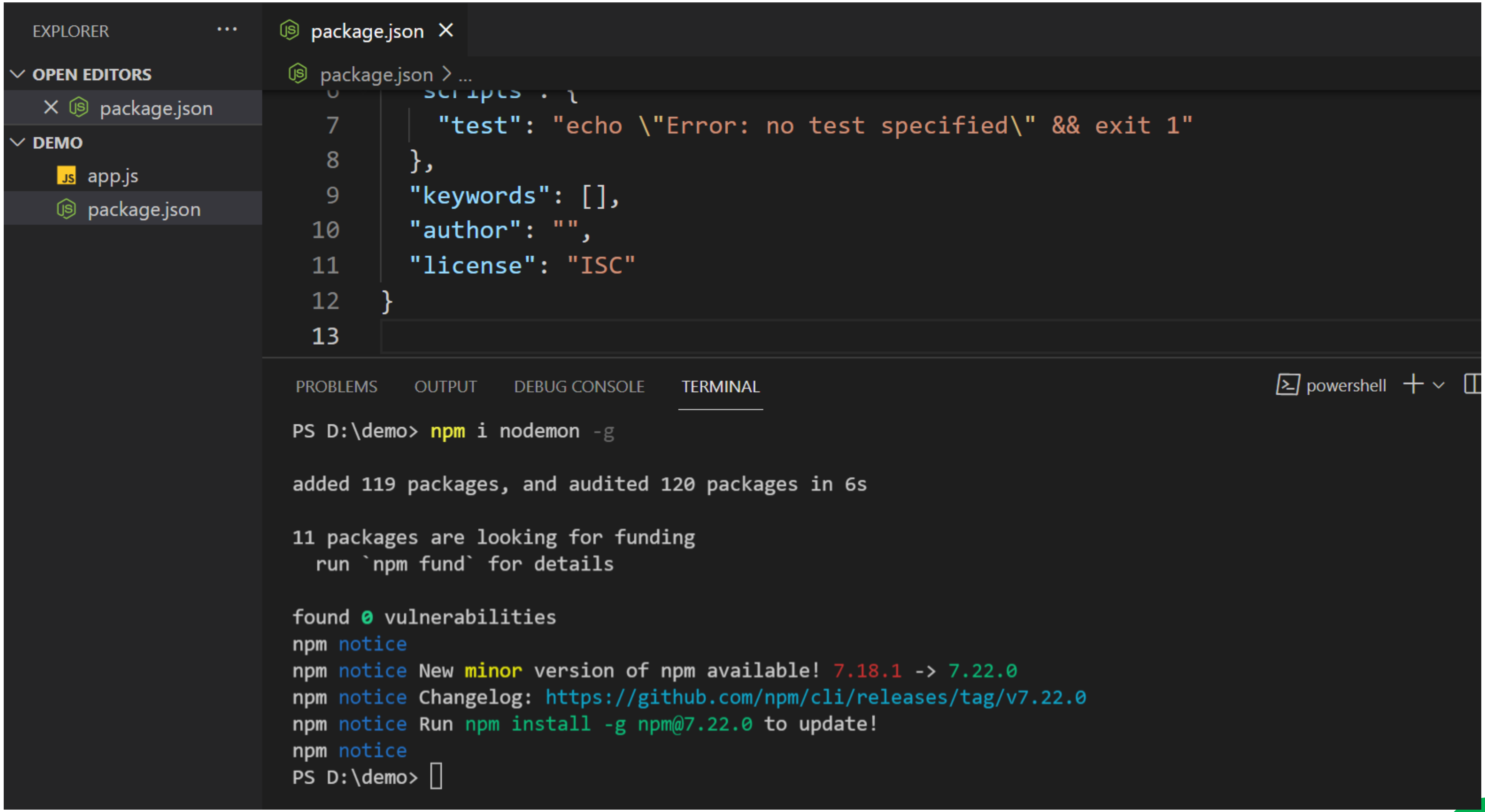


```json
{
  "name": "demo",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```



```json
{
  "name": "demo",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

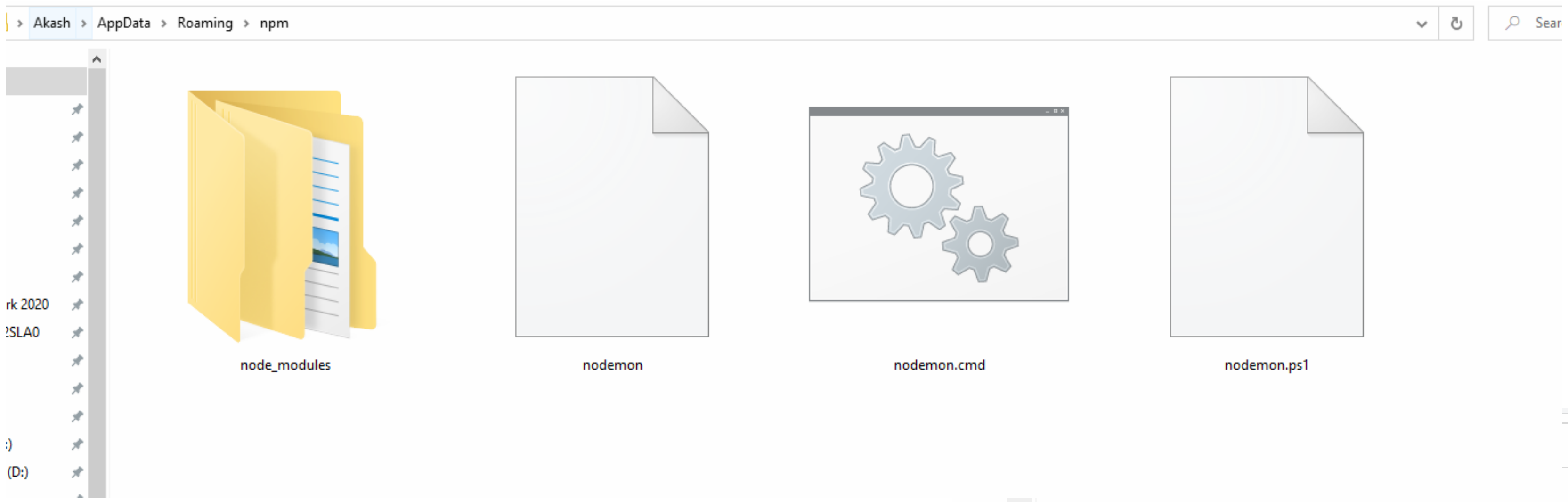
```
PS D:\demo> npm i nodemon -g
```



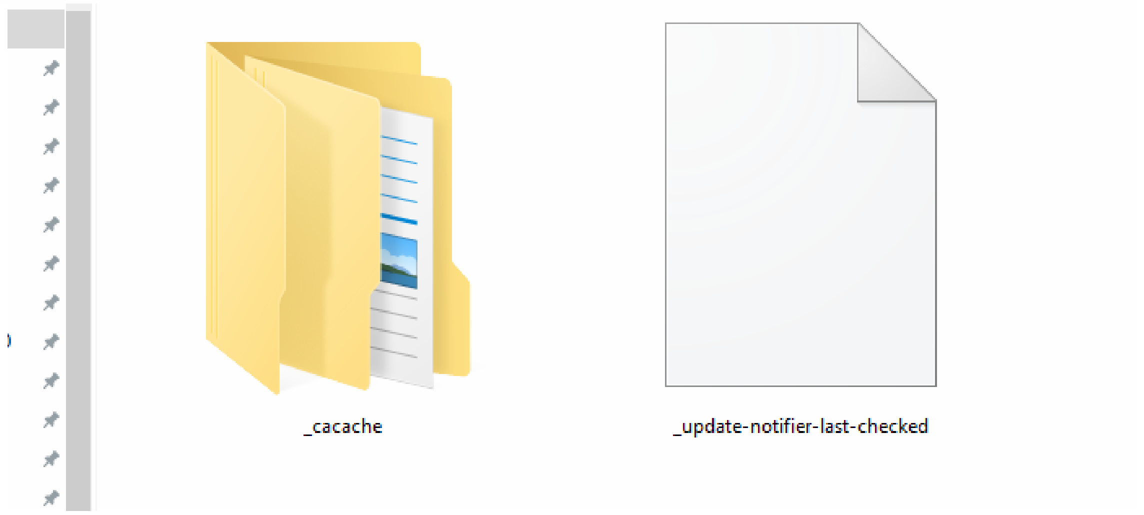
```
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
PS D:\demo> npm i nodemon -g

added 119 packages, and audited 120 packages in 6s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.18.1 -> 7.22.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.22.0
npm notice Run npm install -g npm@7.22.0 to update!
npm notice
PS D:\demo>
```

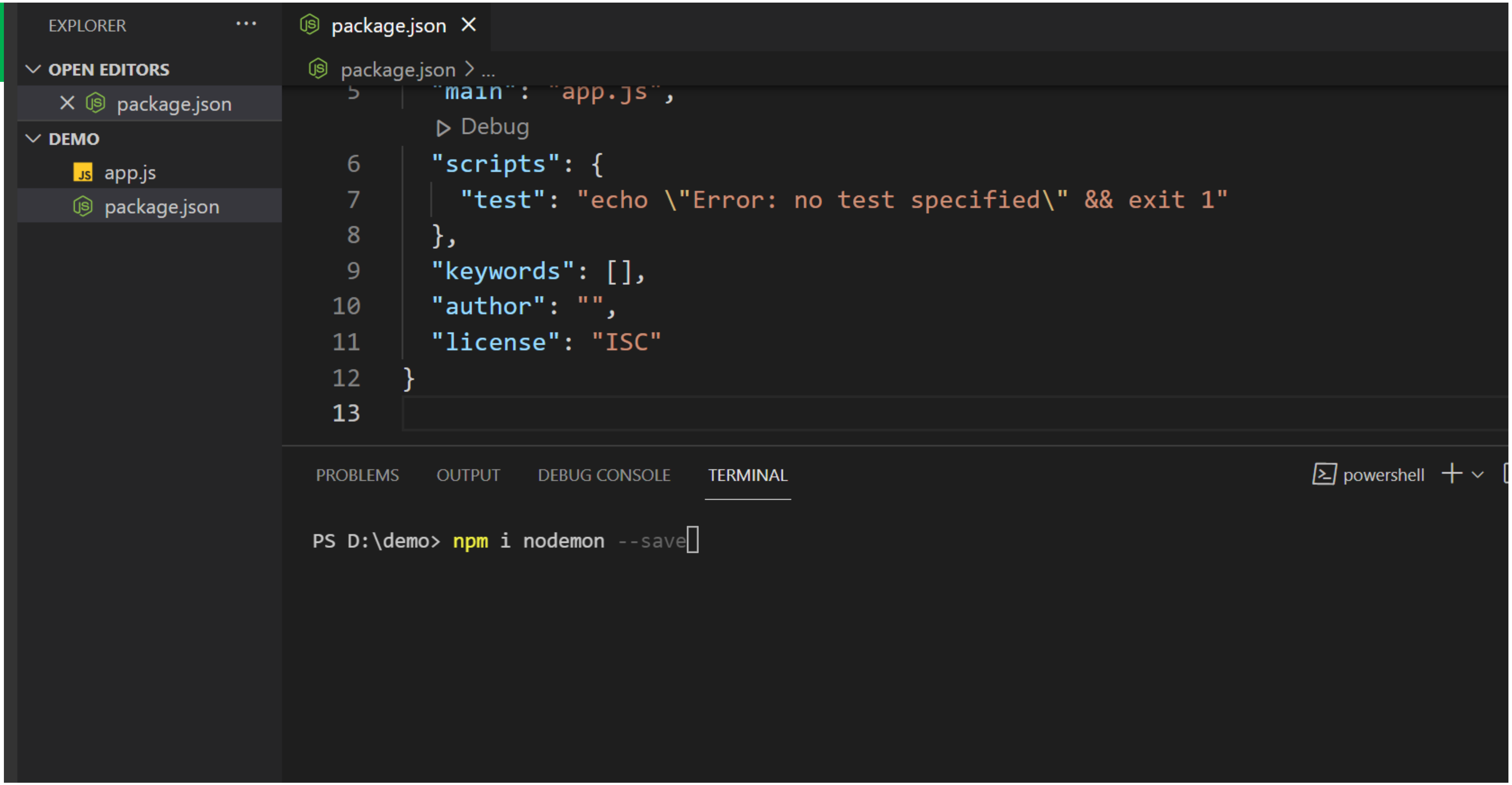
After Downloading Global Package



node_modules nodemon nodemon.cmd nodemon.ps1

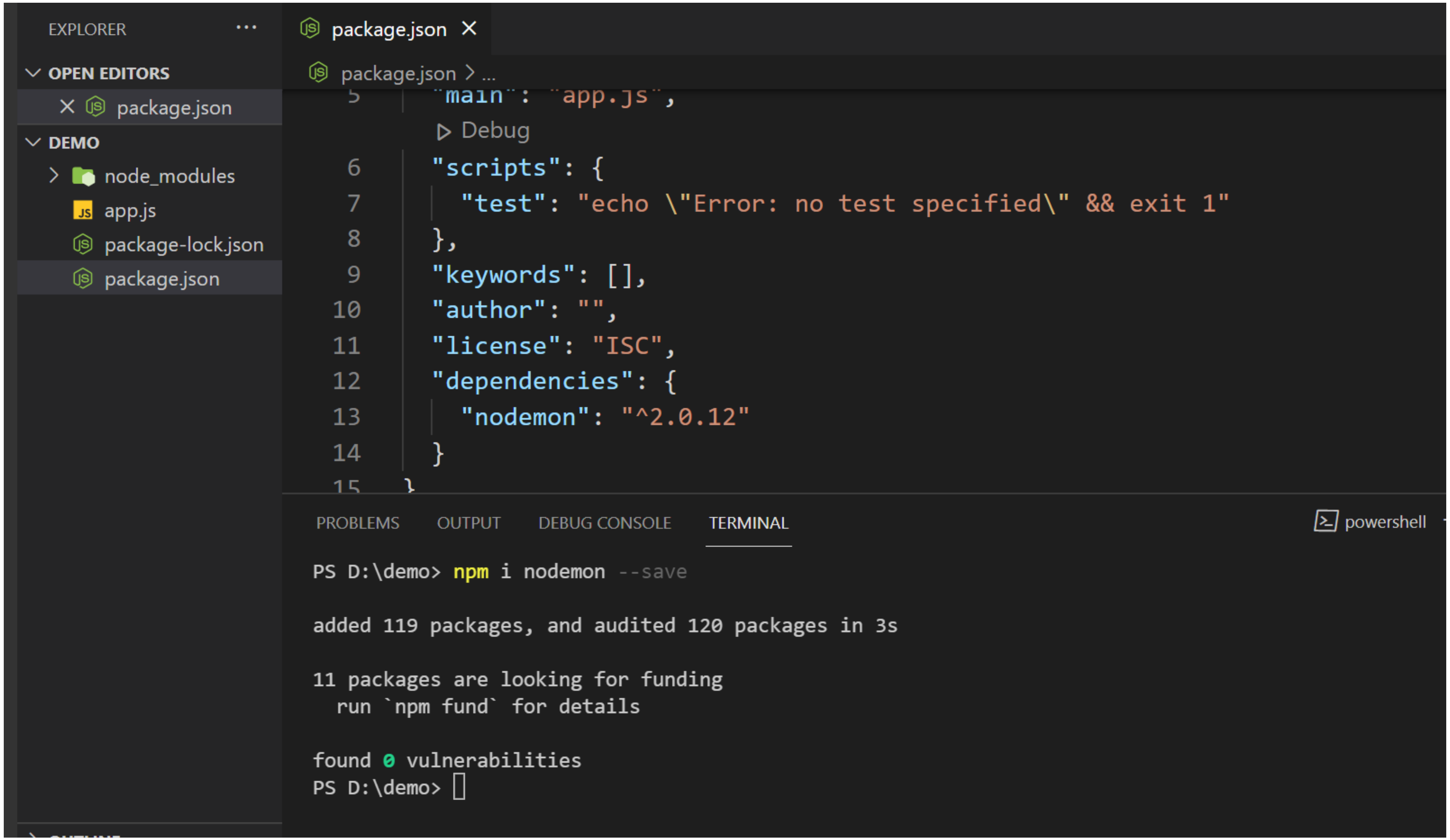


_cacache _update-notifier-last-checked



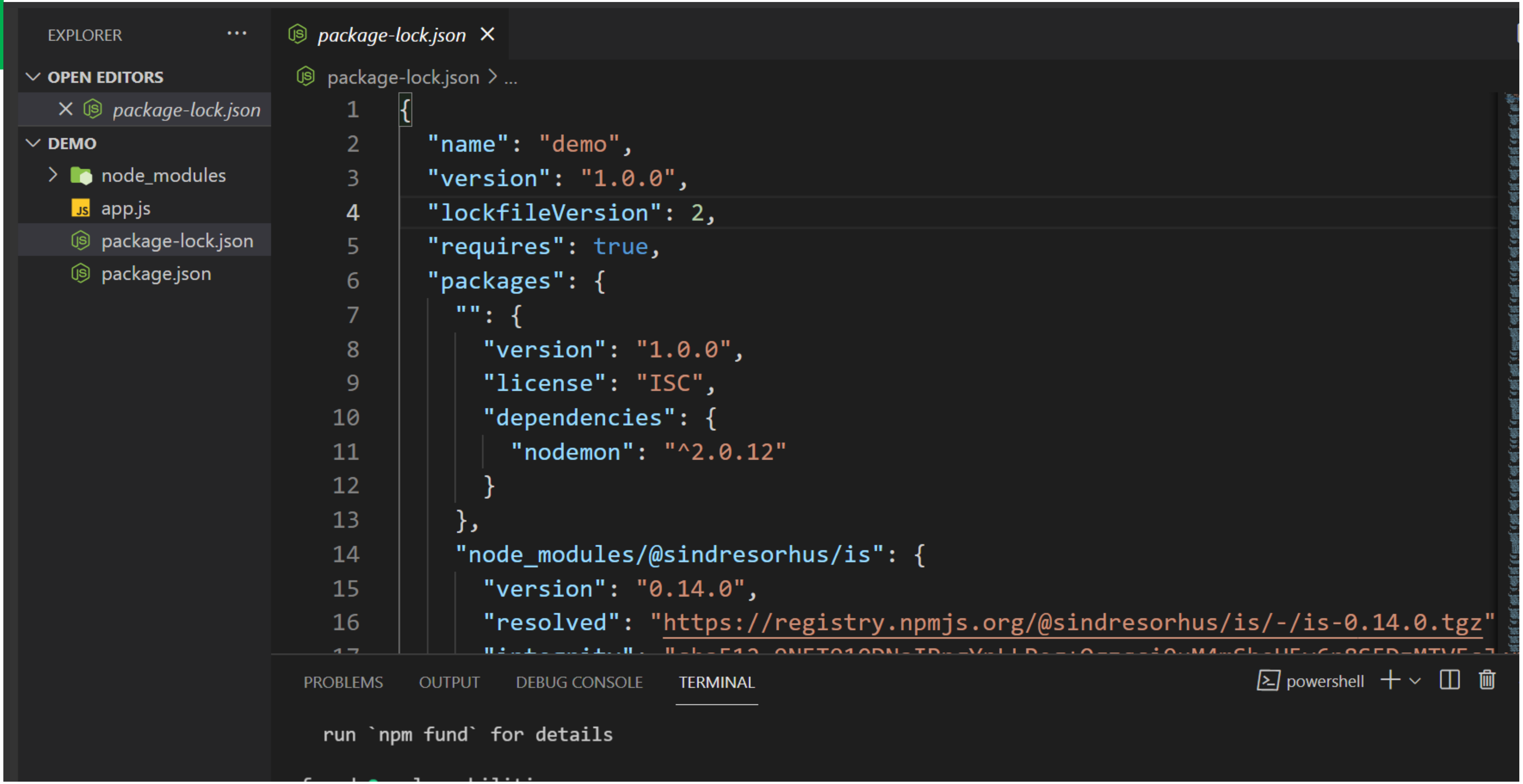
```
  "main": "app.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
PS D:\demo> npm i nodemon --save
```



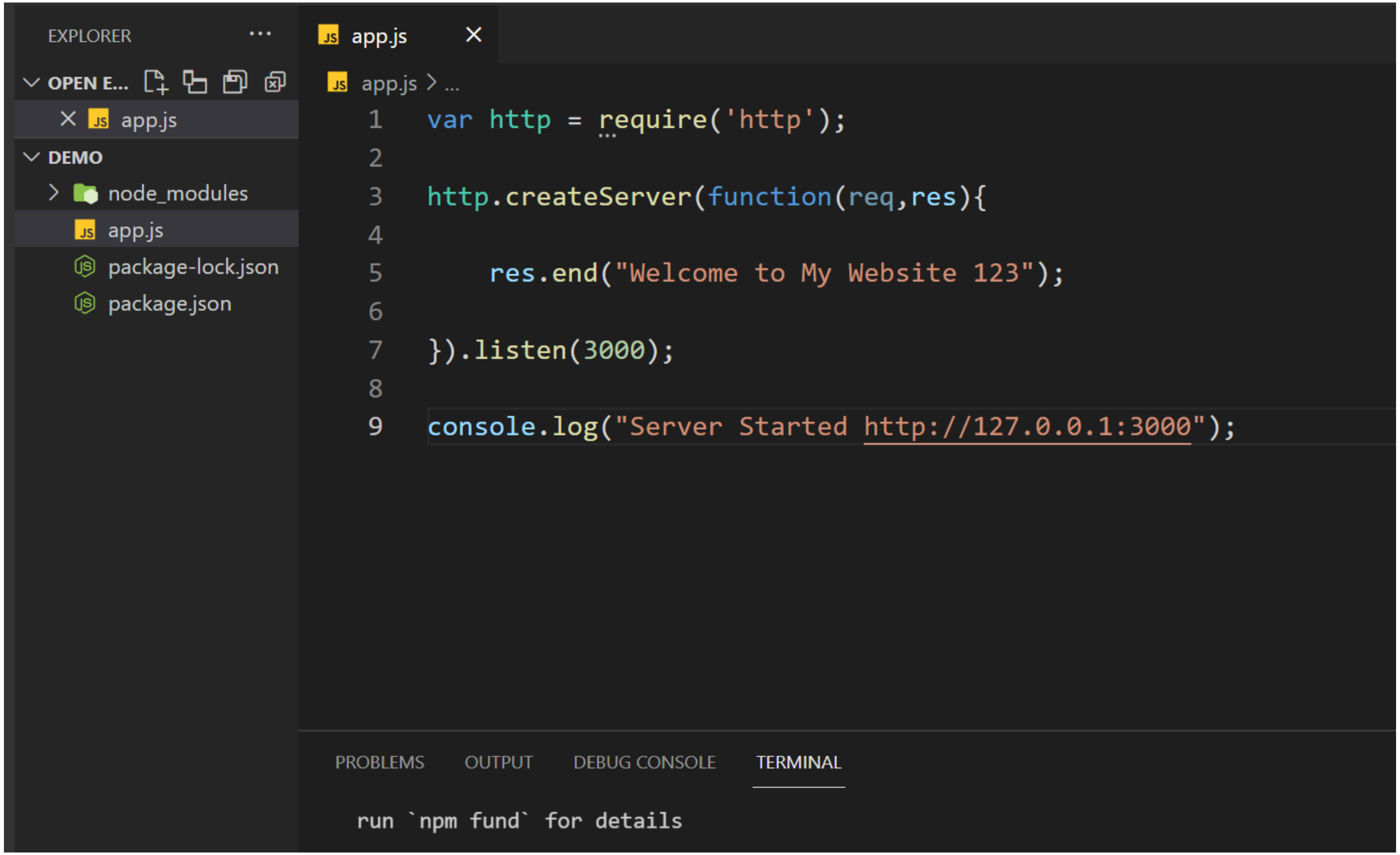
```json
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "nodemon": "^2.0.12"
  }
}
```

```
PS D:\demo> npm i nodemon --save

added 119 packages, and audited 120 packages in 3s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\demo>
```



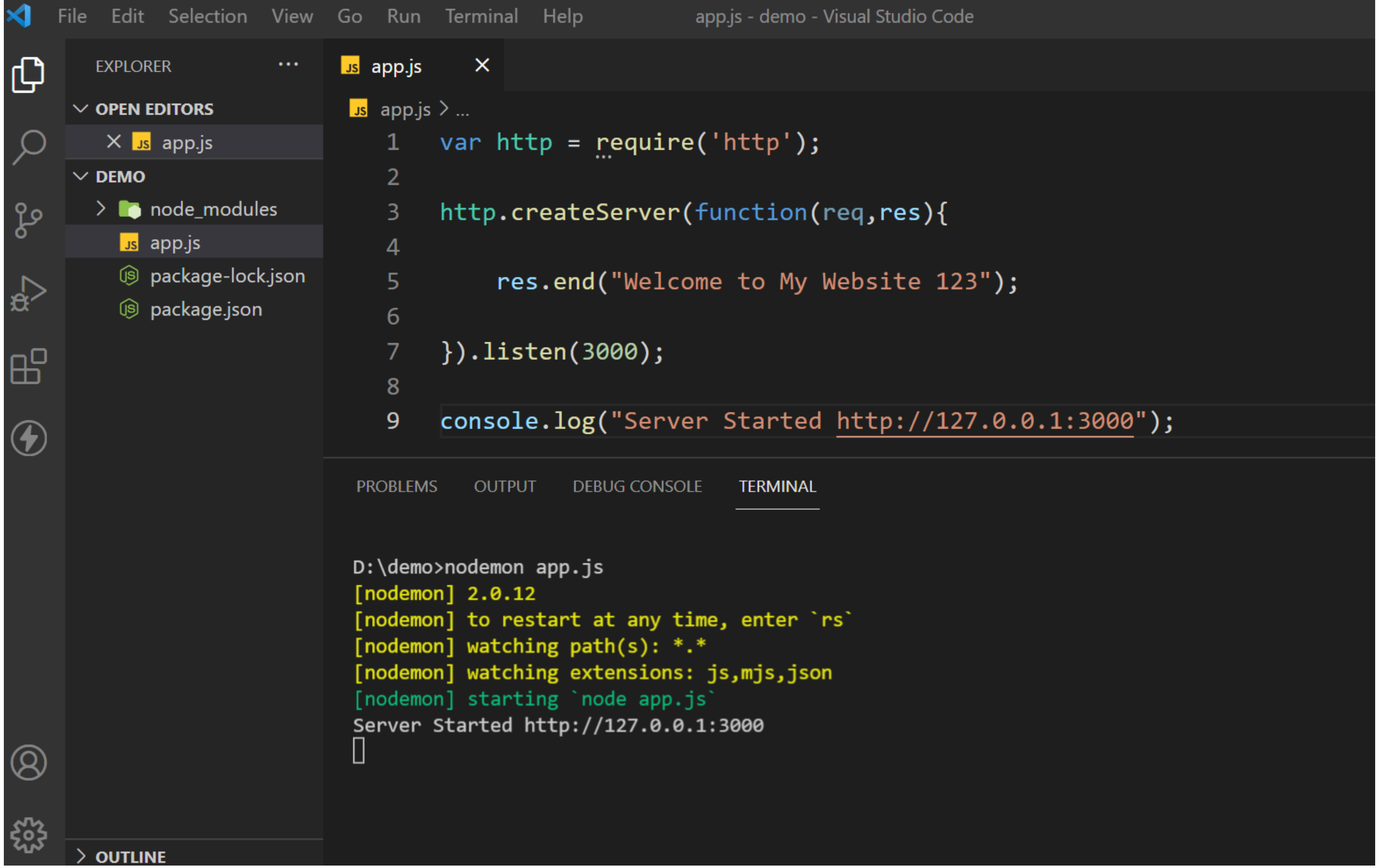
```json
{
  "name": "demo",
  "version": "1.0.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "nodemon": "^2.0.12"
      }
    },
    "node_modules/@sindresorhus/is": {
      "version": "0.14.0",
      "resolved": "https://registry.npmjs.org/@sindresorhus/is/-/is-0.14.0.tgz"
```

```
run `npm fund` for details
```



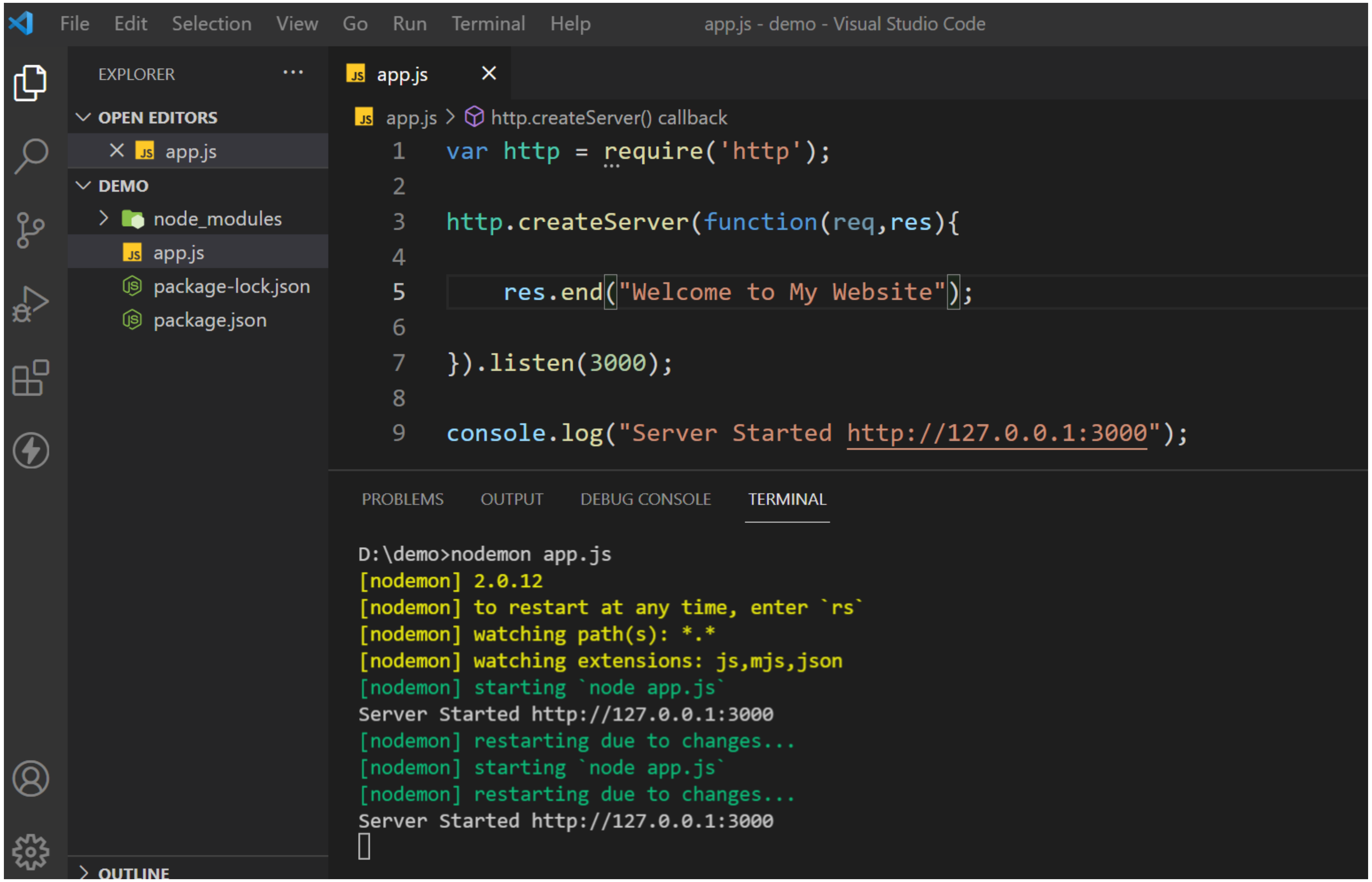
```
var http = require('http');

http.createServer(function(req,res){

    res.end("Welcome to My Website 123");

}).listen(3000);

console.log("Server Started http://127.0.0.1:3000");
```

```
run `npm fund` for details
```



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

http.createServer(function(req,res){

    res.end("Welcome to My Website 123");

}).listen(3000);

console.log("Server Started http://127.0.0.1:3000");
```

```
D:\demo>nodemon app.js
[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Server Started http://127.0.0.1:3000
```



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

http.createServer(function(req,res){

    res.end("Welcome to My Website");

}).listen(3000);

console.log("Server Started http://127.0.0.1:3000");
```

```
D:\demo>nodemon app.js
[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Server Started http://127.0.0.1:3000
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
[nodemon] restarting due to changes...
Server Started http://127.0.0.1:3000
```

- npm -v
- npm update
- npm help
- npm install -h
- npm help update

- npm init
- npm init --yes
- npm init --y

Install Command

- npm install express
- npm update express
- npm install express -g
- npm install express --save
- npm install express --save-dev

Install multiple package

- Npm install express nodemon --save

Uninstall Command

- npm uninstall express
- npm un express
- npm remove express
- npm rm express
- npm uninstall express -g
- npm uninstall express --save
- npm uninstall express --save-dev

Search Command

- npm list
- npm list --depth 1
- npm list --global true --depth 0
- npm list --depth 1

Npm Package Version Check

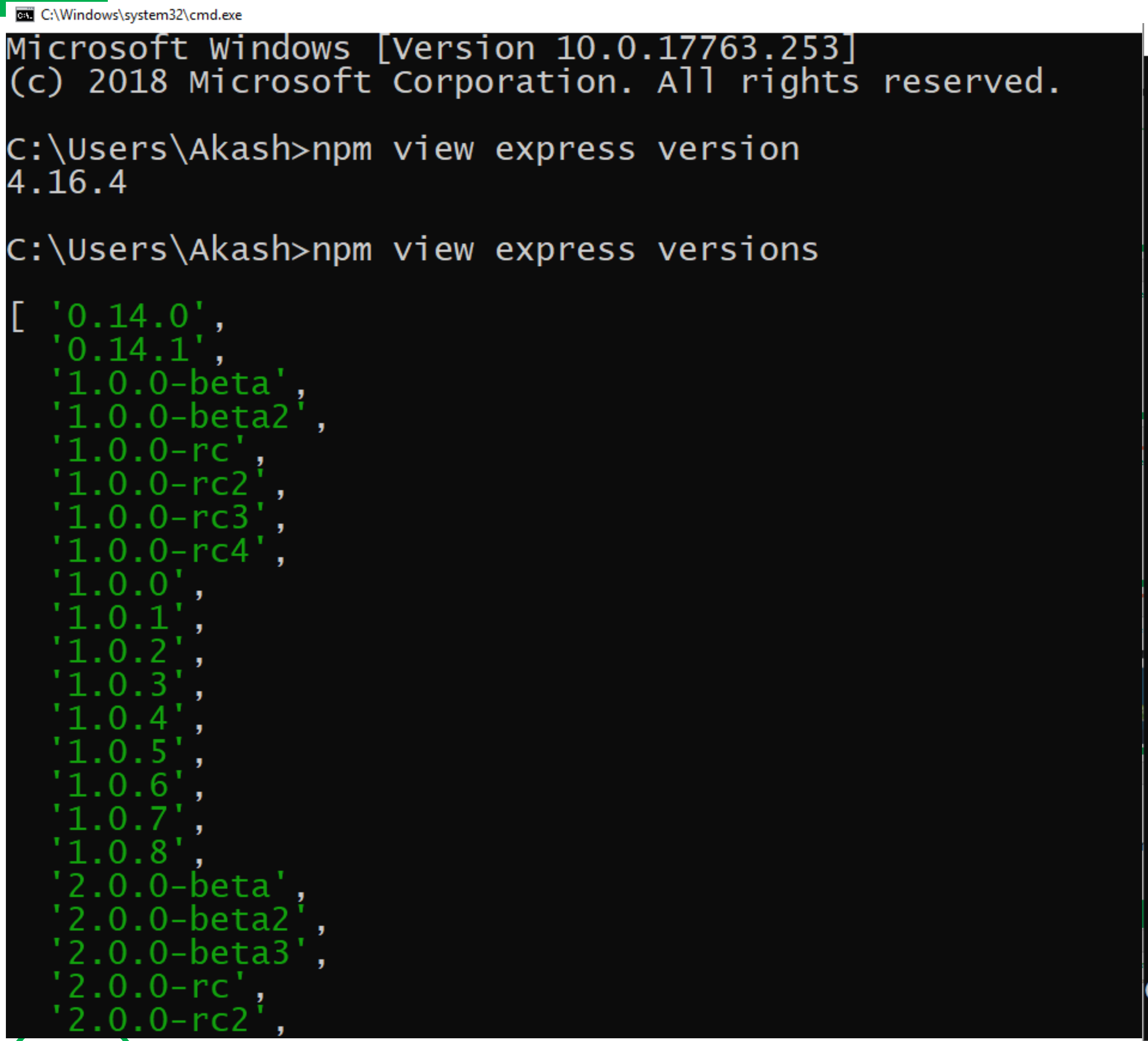
- > npm view express version

Npm Package Version Listing

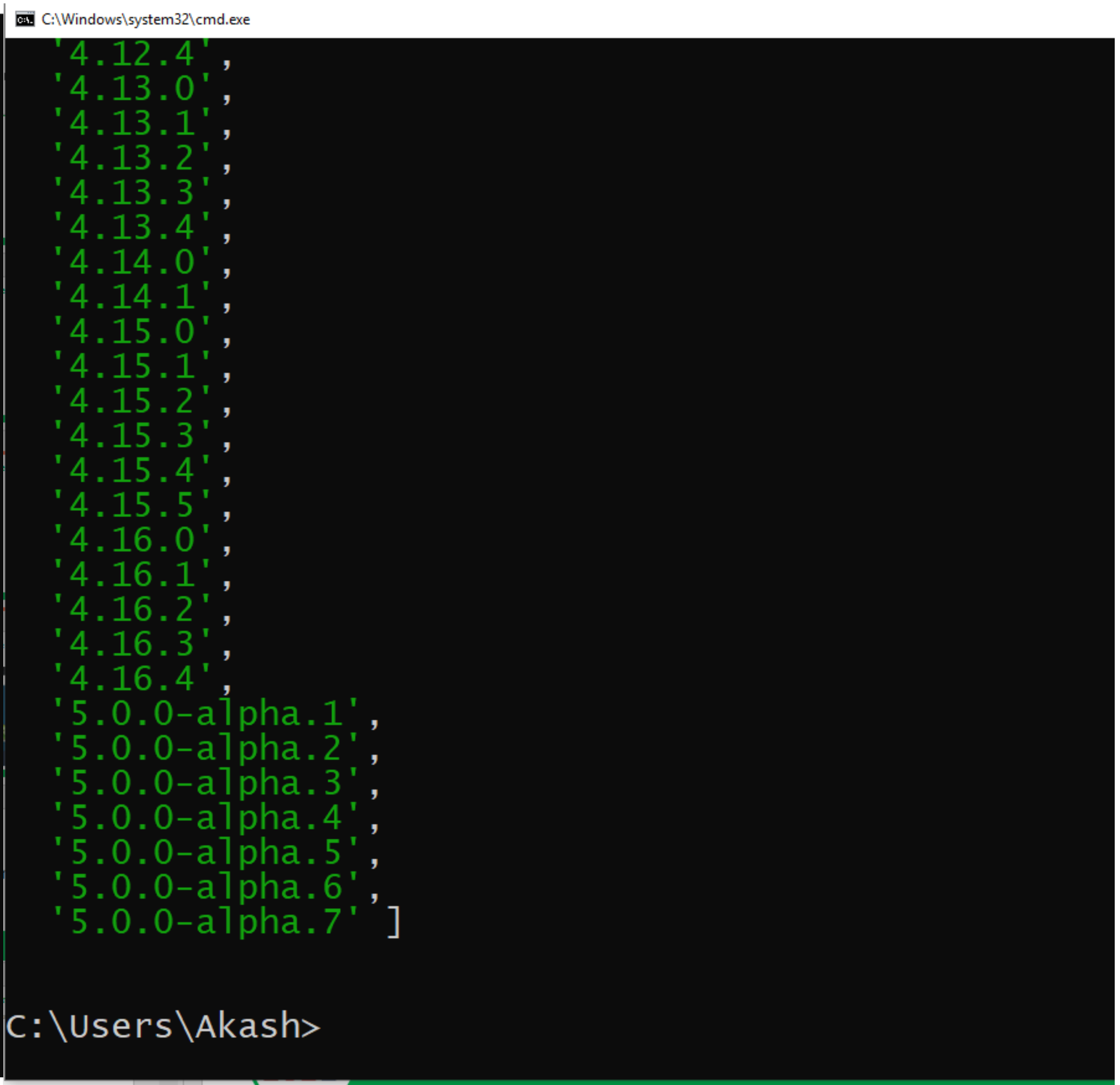
- To View Specific Package version history use below command

```
>   npm view express versions
```

Example



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.253]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Akash>npm view express version
4.16.4

C:\Users\Akash>npm view express versions

[ '0.14.0',
  '0.14.1',
  '1.0.0-beta',
  '1.0.0-beta2',
  '1.0.0-rc',
  '1.0.0-rc2',
  '1.0.0-rc3',
  '1.0.0-rc4',
  '1.0.0',
  '1.0.1',
  '1.0.2',
  '1.0.3',
  '1.0.4',
  '1.0.5',
  '1.0.6',
  '1.0.7',
  '1.0.8',
  '2.0.0-beta',
  '2.0.0-beta2',
  '2.0.0-beta3',
  '2.0.0-rc',
  '2.0.0-rc2',
```



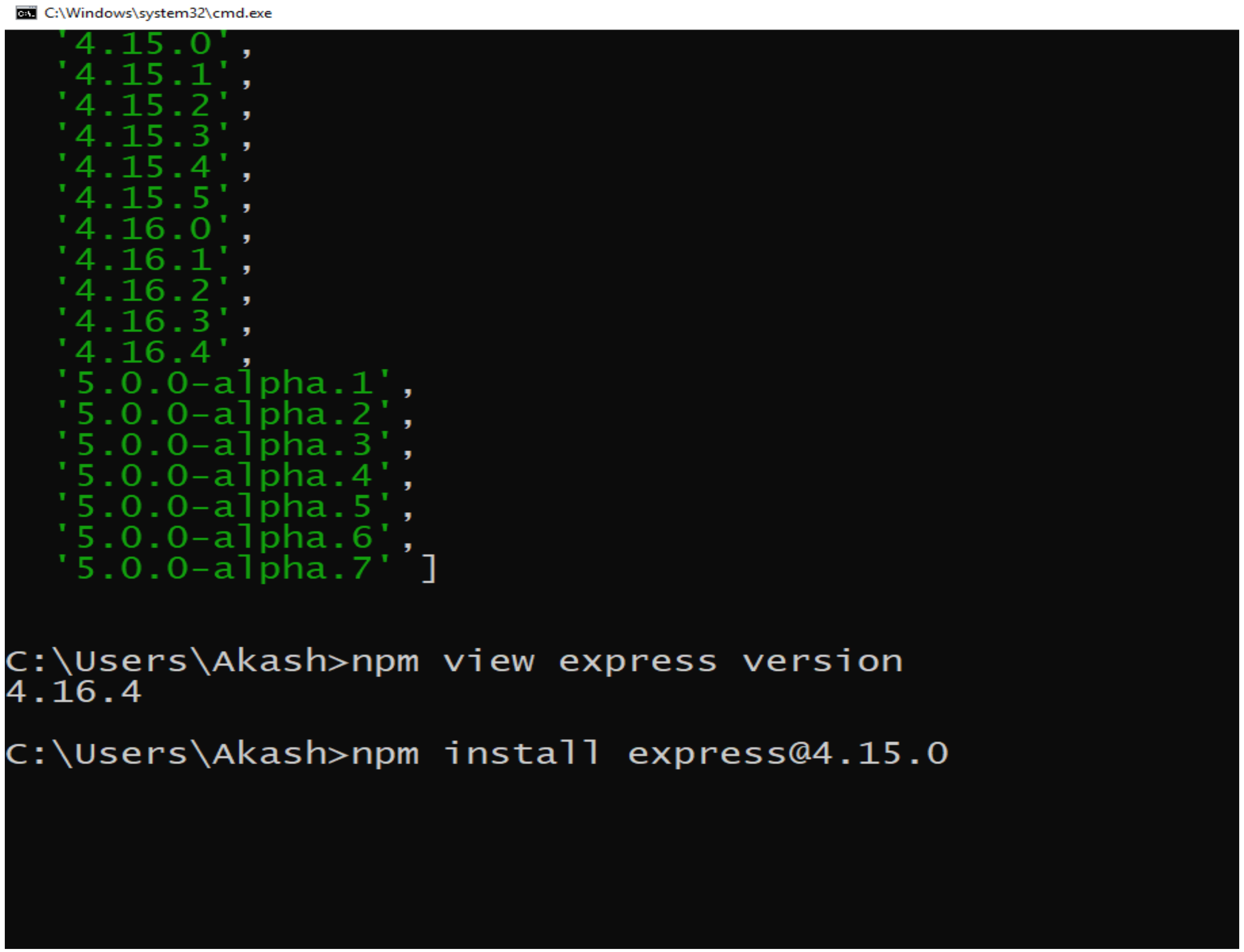
```
C:\Windows\system32\cmd.exe
  '4.12.4',
  '4.13.0',
  '4.13.1',
  '4.13.2',
  '4.13.3',
  '4.13.4',
  '4.14.0',
  '4.14.1',
  '4.15.0',
  '4.15.1',
  '4.15.2',
  '4.15.3',
  '4.15.4',
  '4.15.5',
  '4.16.0',
  '4.16.1',
  '4.16.2',
  '4.16.3',
  '4.16.4',
  '5.0.0-alpha.1',
  '5.0.0-alpha.2',
  '5.0.0-alpha.3',
  '5.0.0-alpha.4',
  '5.0.0-alpha.5',
  '5.0.0-alpha.6',
  '5.0.0-alpha.7' ]


C:\Users\Akash>
```

Install Specific Version of Package

- To Install old version of Package Use Version number.

- > npm install express@4.15.0

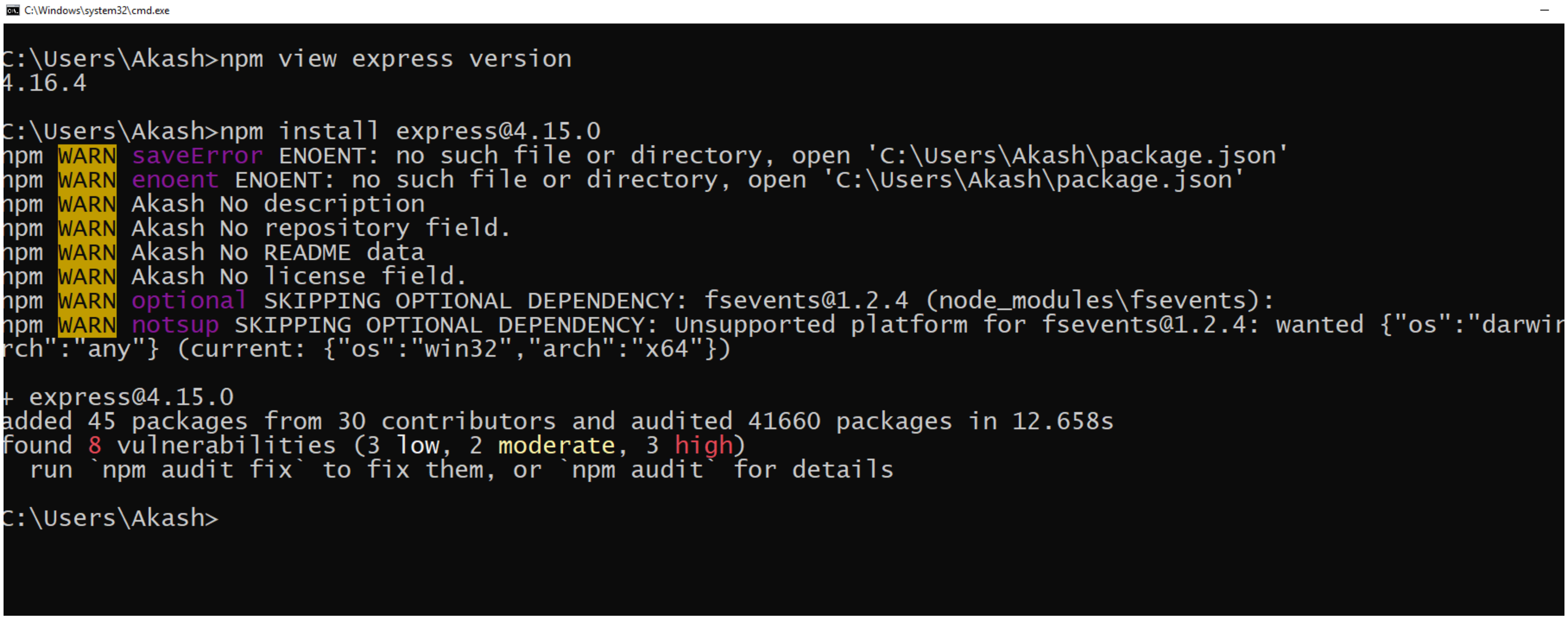
Example



C:\Windows\system32\cmd.exe

```
    '4.15.0',
    '4.15.1',
    '4.15.2',
    '4.15.3',
    '4.15.4',
    '4.15.5',
    '4.16.0',
    '4.16.1',
    '4.16.2',
    '4.16.3',
    '4.16.4',
    '5.0.0-alpha.1',
    '5.0.0-alpha.2',
    '5.0.0-alpha.3',
    '5.0.0-alpha.4',
    '5.0.0-alpha.5',
    '5.0.0-alpha.6',
    '5.0.0-alpha.7' ]


C:\Users\Akash>npm view express version
4.16.4

C:\Users\Akash>npm install express@4.15.0
```

Npm Audit

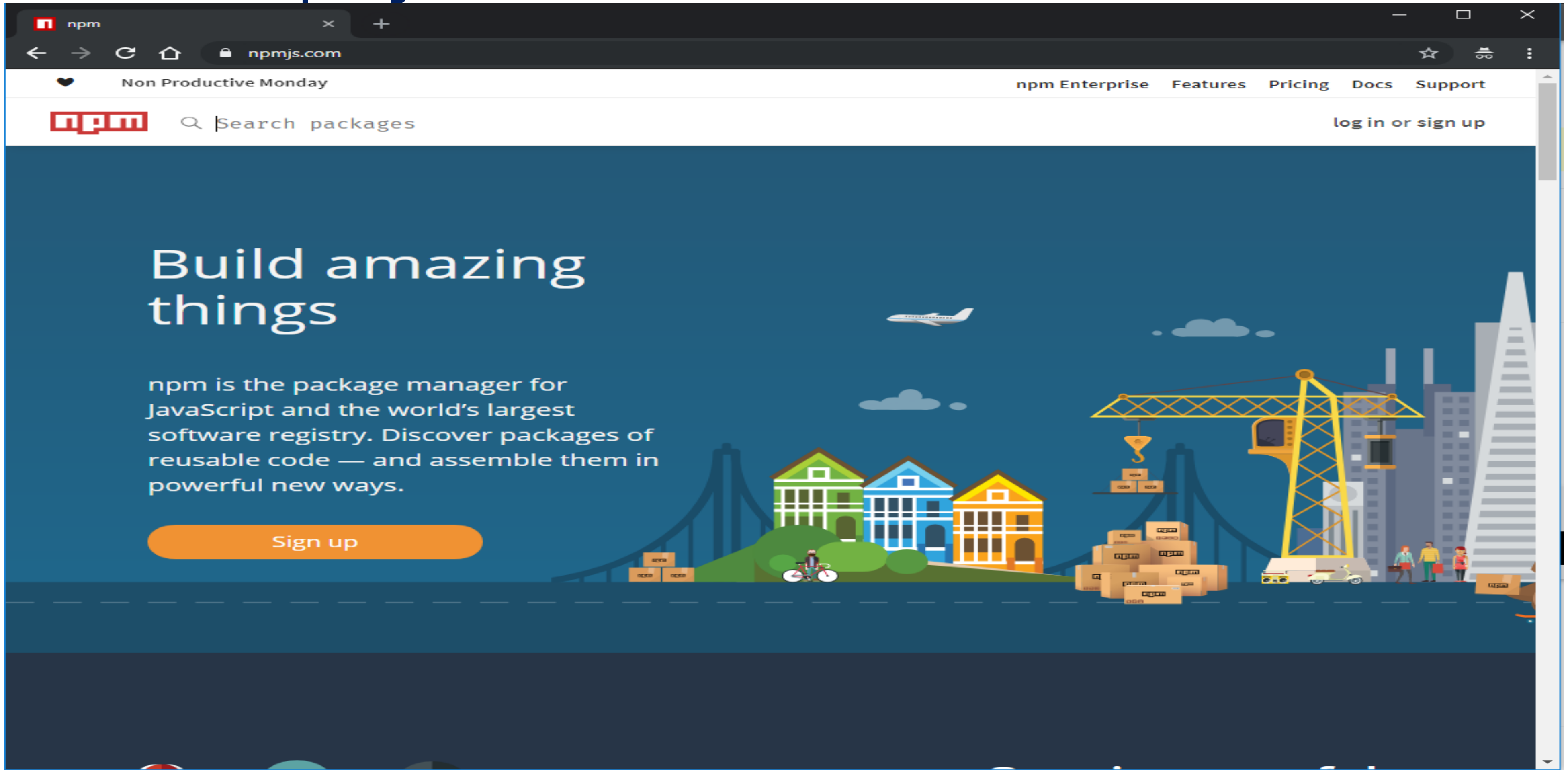


C:\Windows\system32\cmd.exe

```
:\Users\Akash>npm view express version
.16.4

:\Users\Akash>npm install express@4.15.0
pm WARN saveError ENOENT: no such file or directory, open 'C:\Users\Akash\package.json'
pm WARN enoent ENOENT: no such file or directory, open 'C:\Users\Akash\package.json'
pm WARN Akash No description
pm WARN Akash No repository field.
pm WARN Akash No README data
pm WARN Akash No license field.
pm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fsevents):
pm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwi
rch":"any"} (current: {"os":"win32","arch":"x64"})

 express@4.15.0
dded 45 packages from 30 contributors and audited 41660 packages in 12.658s
ound 8 vulnerabilities (3 low, 2 moderate, 3 high)
  run `npm audit fix` to fix them, or `npm audit` for details

:\Users\Akash>
```

NPM Package Search Engine.

- https://www.npmjs.com



Build amazing things

npm is the package manager for JavaScript and the world's largest software registry. Discover packages of reusable code — and assemble them in powerful new ways.

Nodemon Npm

Nodemon NPM

- **“nodemon”** module helps to automatically restart your Server once you made the changes in Source code.
- https://nodemon.io/
- Install Command
- npm install -g nodemon

How to use Nodemon:

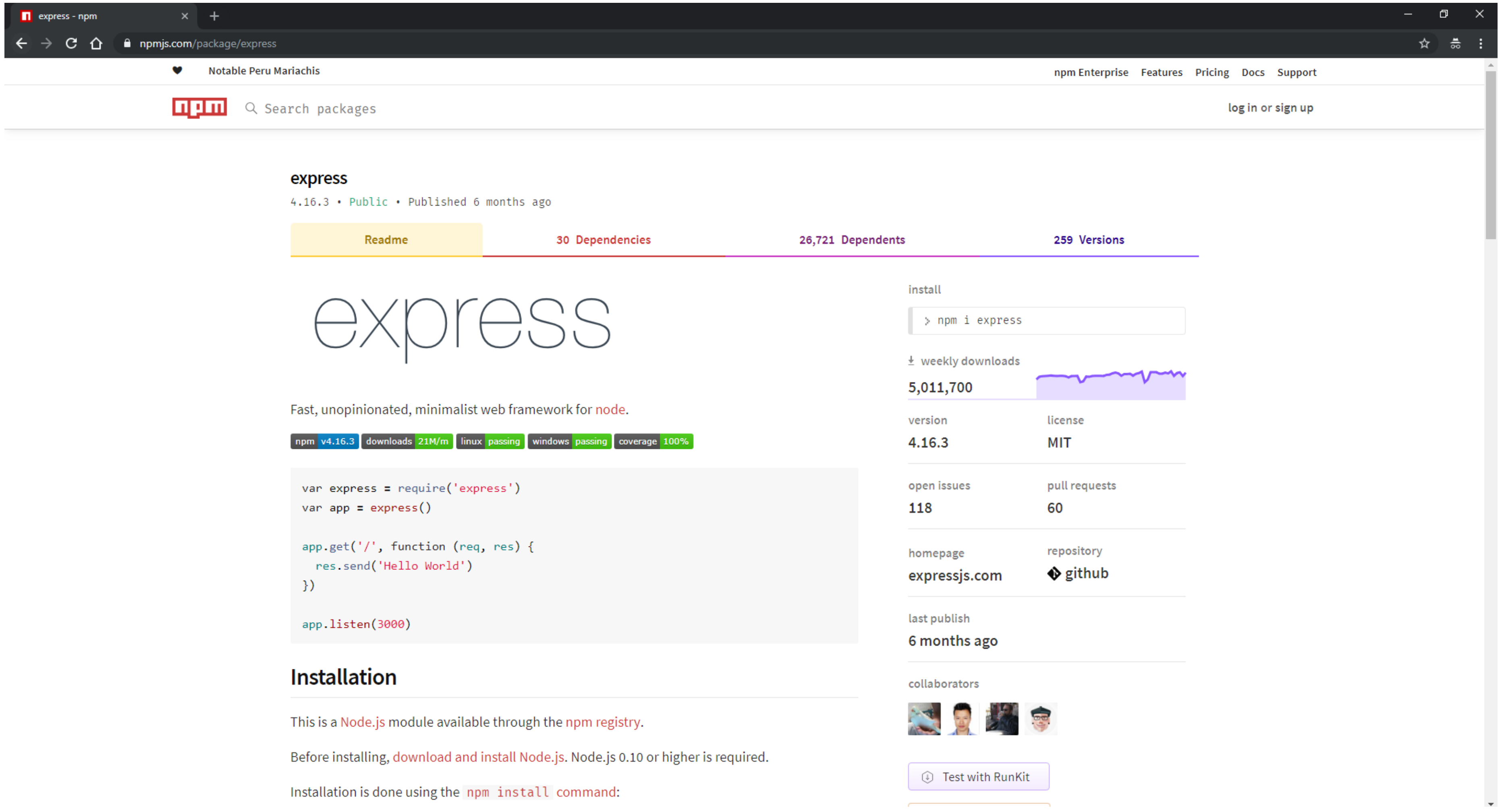
- To restart Node Server program we do this by using
  - node Server_file.js

  but now you have to this using
  - nodemon Server_file.js

- Nodemon will deal with starting it on Specific port and handling code updates. default the time is 1 second but if you want to change

  - nodemon --delay 10 server.js

Express NPM



express

4.16.3 • Public • Published 6 months ago

Fast, unopinionated, minimalist web framework for node.

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

Installation

This is a Node.js module available through the npm registry.

Before installing, download and install Node.js. Node.js 0.10 or higher is required.

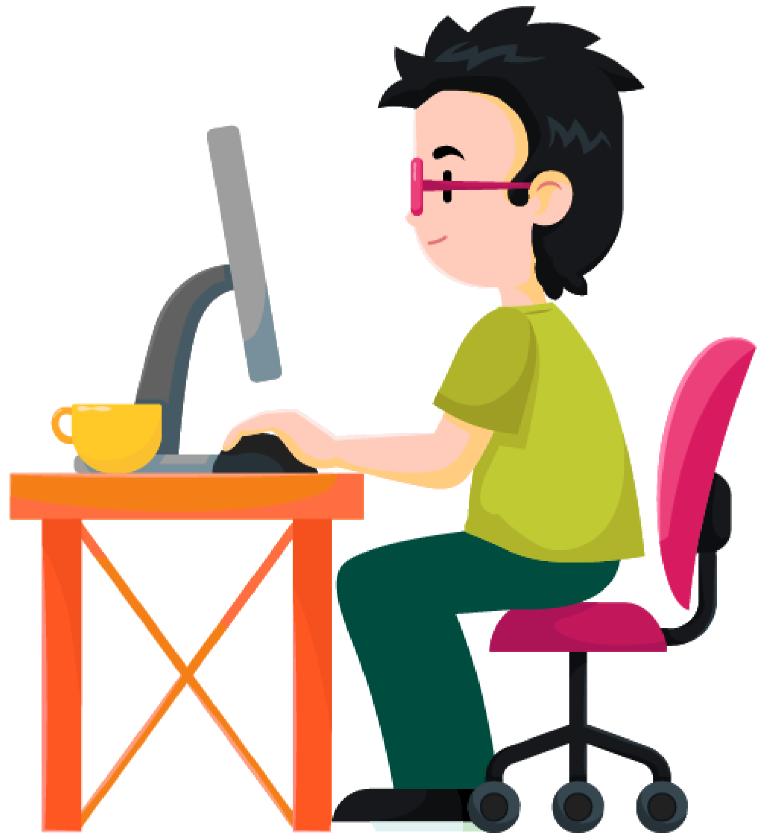
Installation is done using the `npm install` command:

install

```
> npm i express
```

weekly downloads	5,011,700
version	4.16.3
license	MIT
open issues	118
pull requests	60
homepage	expressjs.com
repository	github
last publish	6 months ago

Get Exclusive Video Tutorials



www.aptutorials.com
https://www.youtube.com/user/Akashtips

Get More Details

www.akashsir.com

If You Liked It !

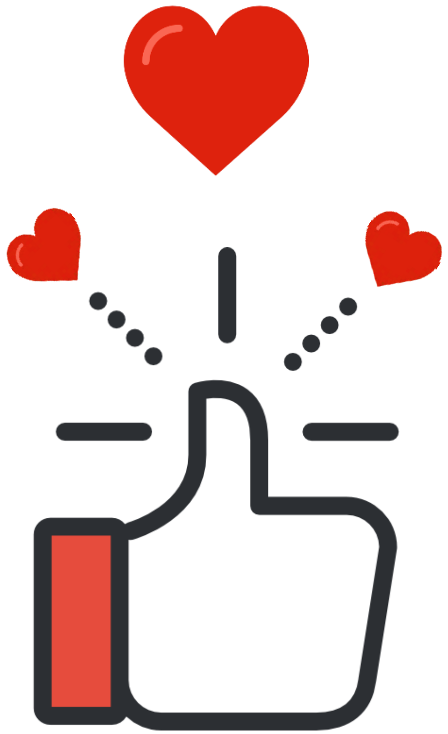
Rating Us Now

Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET

Sulekha

https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad



Connect With Me

Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechnolabs.com
www.akashpadhiyar.com
www.aptutorials.com

# Social Info

Akash.padhiyar
Akashpadhiyar
Akash_padhiyar
+91 99786-21654

#Akashpadhiyar
#aptutorials