



Emerging Technologies

COMP-308

Winter 2018



Introduction to MEAN

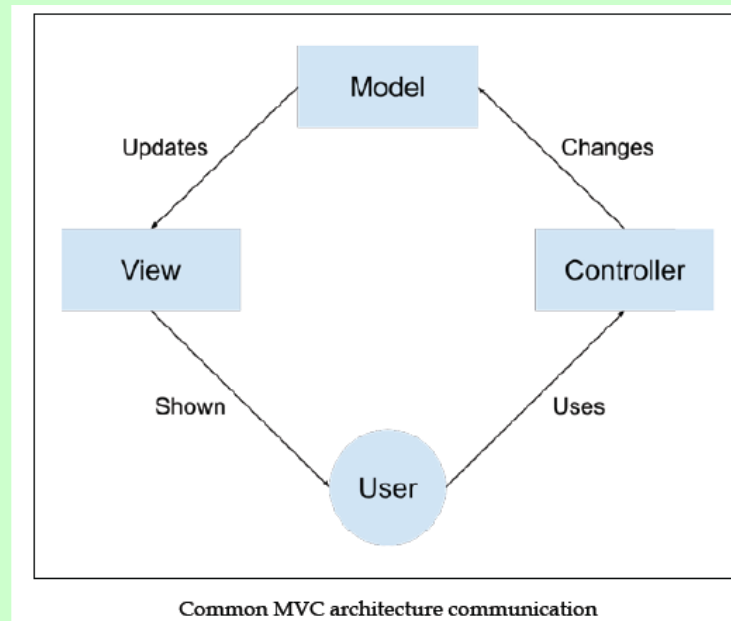
Objectives:

- ☐ Describe MEAN stack architecture.
- ☐ Install and run MongoDB.
- ☐ Install and run Node.js
- ☐ Create simple Node.js apps in Visual Studio 2017



Three-tier web application development

- ❑ The **three-tier architecture** consists of three important layers: **data services**, **business logic**, and **presentation**.
- ❑ In the MVC paradigm, the logic, data, and visualization are separated into three types of objects:
 - **View** - handles UI
 - **Model** - handles the data manipulation
 - **Controller** - controls the model and view



Common MVC architecture communication



Web technology stacks

☐ **LAMP** stack:

- Linux
- Apache
- MySQL
- PHP/Python/Perl

☐ **.NET** stack:

- .NET
- IIS
- ASP.NET
- Web API and WCF
- MS-SQL Server

☐ Other frameworks and tools

☐ Each layer uses a **different knowledge base!**



MEAN Stack

- ❑ MEAN is an abbreviation for **MongoDB**, **Express**, **AngularJS**, and **Node.js**.
- ❑ Uses **only JavaScript - driven solutions** to implement the different parts of a web application.
- ❑ Has the following advantages:
 - A **single language** is used throughout the application
 - All the parts of the application can **support and often enforce** the use of the **MVC architecture**
 - Serialization and deserialization of data structures is no longer needed because **data marshaling is done using JSON objects**.



MEAN Stack

- ❑ **MongoDB** is a scalable NoSQL database that used a JSON-like data model with dynamic schemas.
- ❑ **Express** is a **lightweight node.js web application framework**, providing a robust set of features for building single and multi-page, and hybrid web application.
- ❑ **Node.js** is a **server side JavaScript execution environment** built on Google Chrome's V8 JavaScript runtime - helps in building highly scalable and concurrent applications rapidly.
- ❑ **AngularJS** is a JavaScript framework developed by Google - a complete solution for rapid **front end development**.



LAMP versus MEAN

- ❑ Linux → Node.js (platform)
- ❑ Apache → Express.js (web server)
- ❑ MySQL → MongoDB (persistence layer)
- ❑ PHP or Python or Perl → Angular.js (User Interface)





Advantages



❑ High Performance


- Increasing number of requests
- Reducing response time
- Non-blocking I/O - allows web server to handle more concurrent requests without requiring additional hardware or configuration
- Cross-platform (Windows, Linux, MacOS)
- One single programming language for the entire project



Installing MongoDB

← → ↻ [Secure](#) | <https://www.mongodb.com/download-center#production> ☆  




[DOCS](#) [LEARN](#) [WHAT'S MONGODB?](#) [LOGIN](#)   [Free Sandbox !\[\]\(68c803856f5d0e2869157394e52652f1_img.jpg\)](#) [Download !\[\]\(e46bd8df68d5eebf9040942667ff08d4_img.jpg\)](#)

 [FOR GIANT IDEAS](#) [SOLUTIONS](#) [CLOUD](#) [CUSTOMERS](#) [RESOURCES](#) [ABOUT US](#)

[Atlas](#) **[Community Server](#)** [Enterprise Server](#) [Ops Manager](#) [Compass](#) [Connector for BI](#)

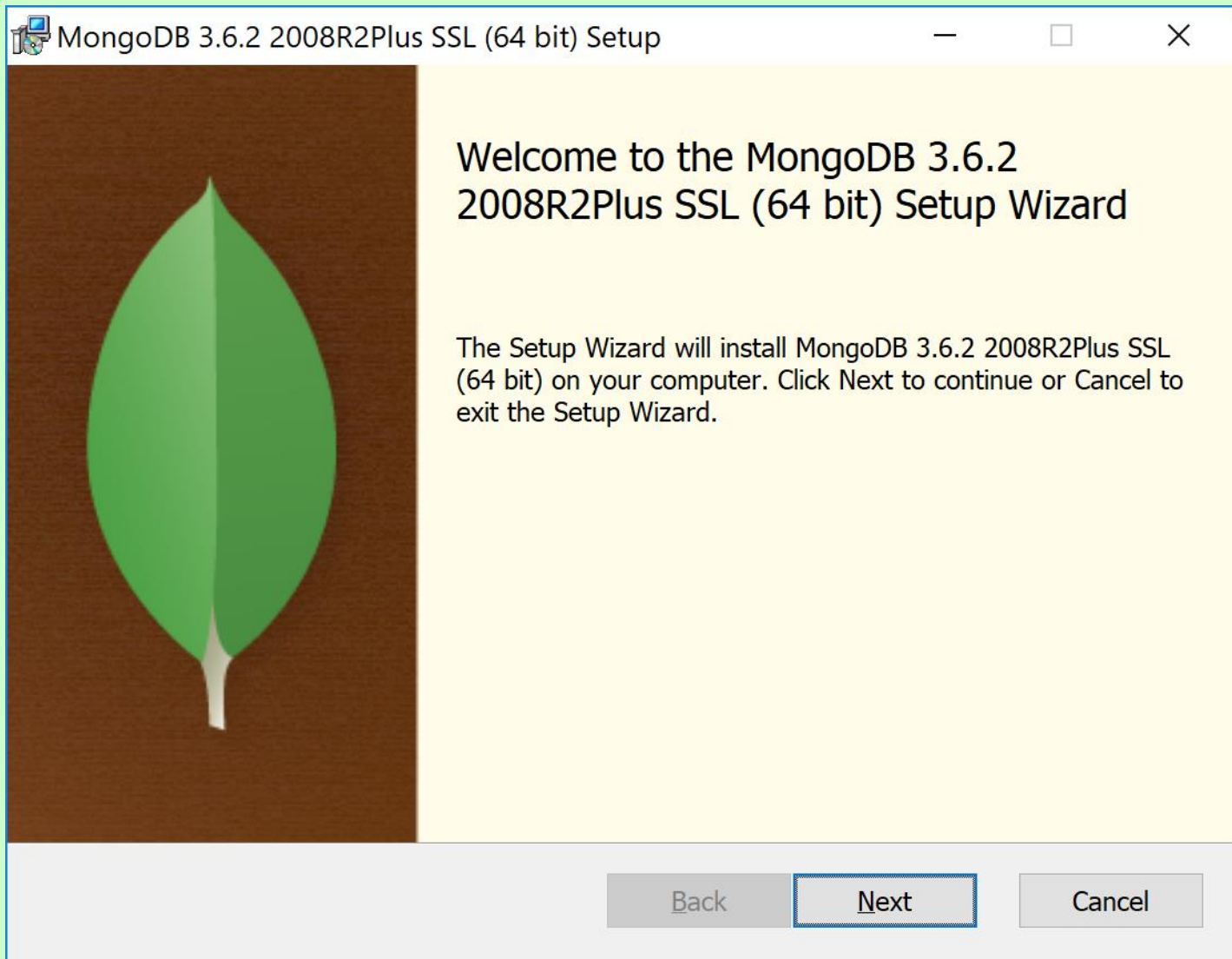
[Current Release](#) | [Previous Releases](#) | [Development Releases](#)

Current Stable Release (3.6.2)
01/10/2018: [Release Notes](#) | [Changelog](#)
Download Source: [tgz](#) | [zip](#)

 Windows  Linux  OSX

Version:

Installation Package:
[↓ DOWNLOAD \(msi\)](#)





Installing MongoDB

```
Command Prompt

C:\Program Files\MongoDB\Server\3.6\bin>dir
Volume in drive C is BOOTCAMP
Volume Serial Number is B27F-0A43

Directory of C:\Program Files\MongoDB\Server\3.6\bin

01/21/2018  12:12 AM    <DIR>          .
01/21/2018  12:12 AM    <DIR>          ..
01/10/2018  07:16 PM         6,937,280  bsondump.exe
01/10/2018  07:40 PM          1,041  InstallCompass.ps1
12/19/2016  06:30 PM        2,000,384  libeay32.dll
01/10/2018  07:28 PM       14,100,992  mongo.exe
01/10/2018  07:41 PM       30,839,296  mongod.exe
01/10/2018  07:42 PM      327,274,496  mongod.pdb
01/10/2018  07:18 PM        9,060,710  mongodump.exe
01/10/2018  07:17 PM        7,204,660  mongoexport.exe
01/10/2018  07:17 PM        7,118,475  mongofiles.exe
01/10/2018  07:17 PM        7,298,763  mongoimport.exe
01/10/2018  07:41 PM       26,096,640  mongoperf.exe
01/10/2018  07:18 PM       10,407,232  mongorestore.exe
01/10/2018  07:33 PM       16,472,064  mongos.exe
01/10/2018  07:33 PM      174,788,608  mongos.pdb
01/10/2018  07:16 PM        7,268,696  mongostat.exe
01/10/2018  07:19 PM        7,072,066  mongotop.exe
12/19/2016  06:30 PM        325,120  ssleay32.dll
          17 File(s)        654,266,523 bytes
           2 Dir(s)      35,371,851,776 bytes free

C:\Program Files\MongoDB\Server\3.6\bin>
```



Running MongoDB

- ❑ **Create MongoDB default folder** to store its files - on Windows, the default location is **C:\data\db**
- ❑ Open the command prompt and execute the following command:
"C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe", or run **mongod** from the **bin** directory:
- ❑ The main **MongoDB service** will start listening to the default 27017 port:

```
C:\WINDOWS\system32\cmd.exe
C:\Users\inika\Desktop>"C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe"
2018-01-21T08:21:07.551-0800 I CONTROL [initandlisten] MongoDB starting : pid=11272 port=27017 dbpath=C:\data\db\ 64-bit host=PR_MA2308-INIKA
2018-01-21T08:21:07.551-0800 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] db version v3.6.2
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] git version: 489d177dbd0f0420a8ca04d39fd78d0a2c539420
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips 22 Sep 2016
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] allocator: tcmalloc
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] modules: none
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] build environment:
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten]   distmod: 2008plus-ssl
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten]   distarch: x86_64
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten]   target_arch: x86_64
2018-01-21T08:21:07.553-0800 I CONTROL [initandlisten] options: {}
2018-01-21T08:21:07.555-0800 I - [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' storage engine, so setting the active
storage engine to 'wiredTiger'.
2018-01-21T08:21:07.555-0800 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=7618M,session_max=20000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log
=(wait=0),verbose=(recovery_progress),
2018-01-21T08:21:07.709-0800 I STORAGE [initandlisten] WiredTiger message [1516551667:709212][11272:140735762150144], txn-recover: Main recovery loop: star
ting at 2/4608
2018-01-21T08:21:07.850-0800 I STORAGE [initandlisten] WiredTiger message [1516551667:850361][11272:140735762150144], txn-recover: Recovering log 2 through
3
2018-01-21T08:21:07.926-0800 I STORAGE [initandlisten] WiredTiger message [1516551667:926439][11272:140735762150144], txn-recover: Recovering log 3 through
3
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-01-21T08:21:08.064-0800 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2018-01-21T08:21:08.066-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.068-0800 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2018-01-21T08:21:08.070-0800 I CONTROL [initandlisten] **      Remote systems will be unable to connect to this server.
2018-01-21T08:21:08.072-0800 I CONTROL [initandlisten] **      Start the server with --bind_ip <address> to specify which IP
2018-01-21T08:21:08.074-0800 I CONTROL [initandlisten] **      addresses it should serve responses from, or with --bind_ip_all to
2018-01-21T08:21:08.099-0800 I CONTROL [initandlisten] **      bind to all interfaces. If this behavior is desired, start the
2018-01-21T08:21:08.101-0800 I CONTROL [initandlisten] **      server with --bind_ip 127.0.0.1 to disable this warning.
2018-01-21T08:21:08.104-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.106-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.108-0800 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is configured to be greater than 40% of the total
memory. This can lead to increased memory pressure and poor performance.
2018-01-21T08:21:08.131-0800 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-cache
2018-01-21T08:21:08.134-0800 I CONTROL [initandlisten]
2018-01-21T11:21:08.401-0500 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:\data\db\diagnostic.data'
2018-01-21T11:21:08.413-0500 I NETWORK [initandlisten] waiting for connections on port 27017
```



Running MongoDB as a service

- ❑ Specify a path for the MongoDB log and configuration files.
 - Start by creating a folder for these files by running the following command in your command prompt:

> md C:\mongodb\log

- ❑ Then, you'll be able to create a configuration file using the --logpath command-line flag, so in the command prompt, issue the following command:

> echo logpath=C:\mongodb\log\mongo.log > C:\mongodb\mongod.cfg

- ❑ When you have your configuration file in place, open a new command prompt window with administrative privileges by right-clicking on the command prompt icon and clicking on **Run as administrator**.
- ❑ In the new command prompt window, install the MongoDB service by running the following command:
- ❑ **> sc.exe create MongoDB binPath= "\"C:\mongodb\bin\mongod.exe\" - service --config=\"C:\mongodb\mongod.cfg\" Display Name= \"MongoDB 2.6\" start=\"auto\"**

- ❑ If the service was successfully created, you will get the following log message:

[SC] CreateService SUCCESS

- ❑ Run it from command prompt:

net start MongoDB



Using the MongoDB shell

- ❑ MongoDB shell allows you to interact with your server instance using the command line.
- ❑ To start the shell, navigate to the MongoDB bin folder and run the mongo service as follows:

```
C:\WINDOWS\system32\cmd.exe

C:\Users\inika\Desktop>"C:\Program Files\MongoDB\Server\3.6\bin\mongo.exe"
MongoDB shell version v3.6.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.2
Server has startup warnings:
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-01-21T08:21:08.064-0800 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2018-01-21T08:21:08.066-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.068-0800 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2018-01-21T08:21:08.070-0800 I CONTROL [initandlisten] **      Remote systems will be unable to connect to this server.
2018-01-21T08:21:08.072-0800 I CONTROL [initandlisten] **      Start the server with --bind_ip <address> to specify which IP
2018-01-21T08:21:08.074-0800 I CONTROL [initandlisten] **      addresses it should serve responses from, or with --bind_ip_all to
2018-01-21T08:21:08.099-0800 I CONTROL [initandlisten] **      bind to all interfaces. If this behavior is desired, start the
2018-01-21T08:21:08.101-0800 I CONTROL [initandlisten] **      server with --bind_ip 127.0.0.1 to disable this warning.
2018-01-21T08:21:08.104-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.106-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.108-0800 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is configured to be greater t
han 40% of the total memory. This can lead to increased memory pressure and poor performance.
2018-01-21T08:21:08.131-0800 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-cache
2018-01-21T08:21:08.134-0800 I CONTROL [initandlisten]
```



Testing the database

❑ To test your database, run the following command:

> **db.courses.insert({title: "COMP-308 Emerging Technologies"}).**

❑ To retrieve the course object, execute the following command:

> **db.courses.find()**

```
C:\WINDOWS\system32\cmd.exe
C:\Users\inika\Desktop>"C:\Program Files\MongoDB\Server\3.6\bin\mongo.exe"
MongoDB shell version v3.6.2
connecting to: mongod://127.0.0.1:27017
MongoDB server version: 3.6.2
Server has startup warnings:
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.061-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-01-21T08:21:08.064-0800 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2018-01-21T08:21:08.066-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.068-0800 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2018-01-21T08:21:08.070-0800 I CONTROL [initandlisten] **      Remote systems will be unable to connect to this server.
2018-01-21T08:21:08.072-0800 I CONTROL [initandlisten] **      Start the server with --bind_ip <address> to specify which IP
2018-01-21T08:21:08.074-0800 I CONTROL [initandlisten] **      addresses it should serve responses from, or with --bind_ip_all to
2018-01-21T08:21:08.099-0800 I CONTROL [initandlisten] **      bind to all interfaces. If this behavior is desired, start the
2018-01-21T08:21:08.101-0800 I CONTROL [initandlisten] **      server with --bind_ip 127.0.0.1 to disable this warning.
2018-01-21T08:21:08.104-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.106-0800 I CONTROL [initandlisten]
2018-01-21T08:21:08.108-0800 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is configured to be greater t
han 40% of the total memory. This can lead to increased memory pressure and poor performance.
2018-01-21T08:21:08.131-0800 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-cache
2018-01-21T08:21:08.134-0800 I CONTROL [initandlisten]
> db.courses.insert({title: "COMP-308 Emerging Technologies"})
WriteResult({ "nInserted" : 1 })
> db.courses.find()
{ "_id" : ObjectId("5a64c3113aa699c004df951d"), "title" : "COMP-308 Emerging Technologies" }
>
```



Installing Node.js


Download | Node.js x


Secure | <https://nodejs.org/en/download/current/>


Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users

Current
Latest Features


Windows Installer
node-v9.4.0-x64.msi


macOS Installer
node-v9.4.0.pkg

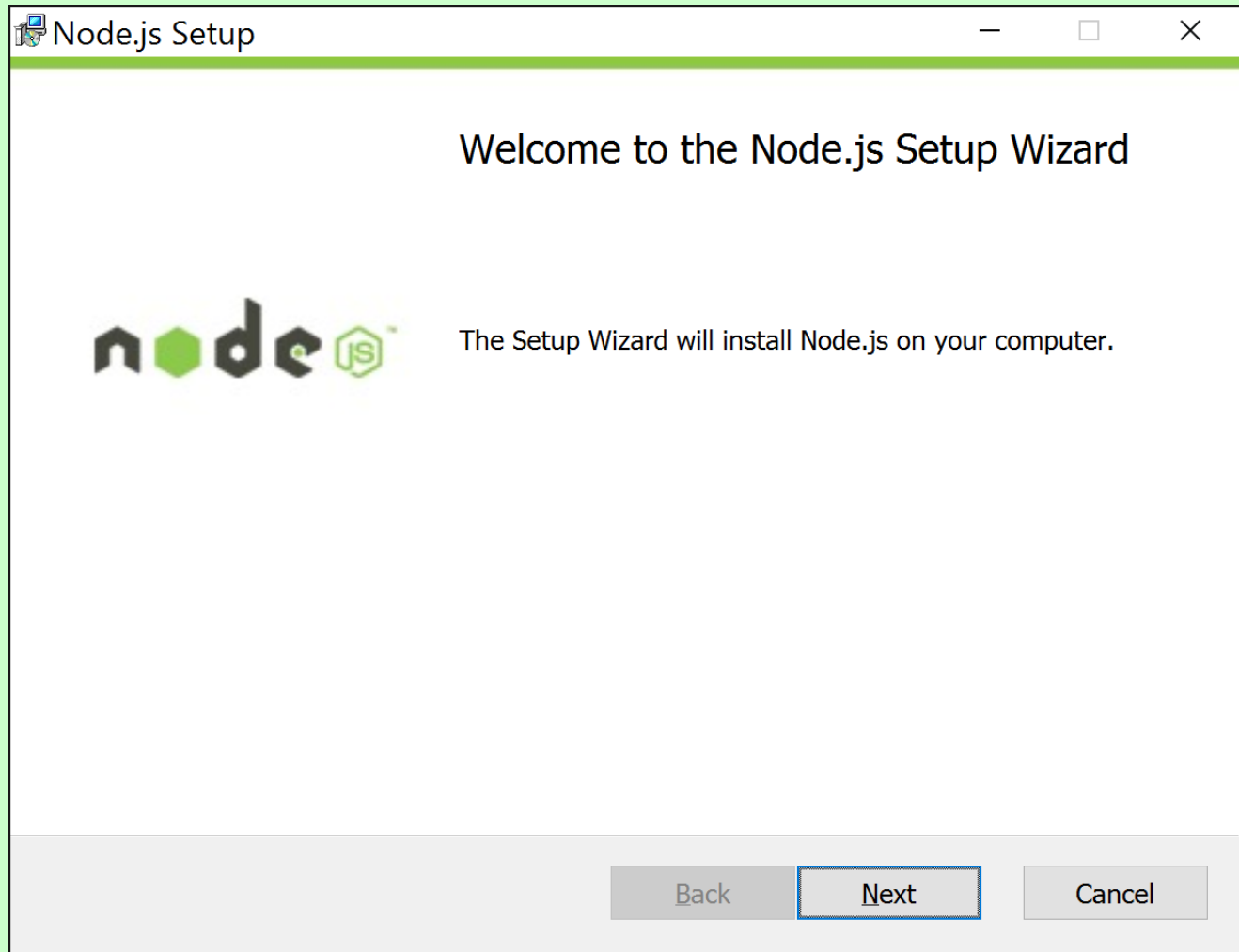

Source Code
node-v9.4.0.tar.gz

Windows Installer (.msi)	32-bit	64-bit	
Windows Binary (.zip)	32-bit	64-bit	
macOS Installer (.pkg)	64-bit		
macOS Binaries (.tar.gz)	64-bit		
Linux Binaries (x64)	64-bit		
Linux Binaries (ARM)	ARMv6	ARMv7	ARMv8
Source Code	node-v9.4.0.tar.gz		



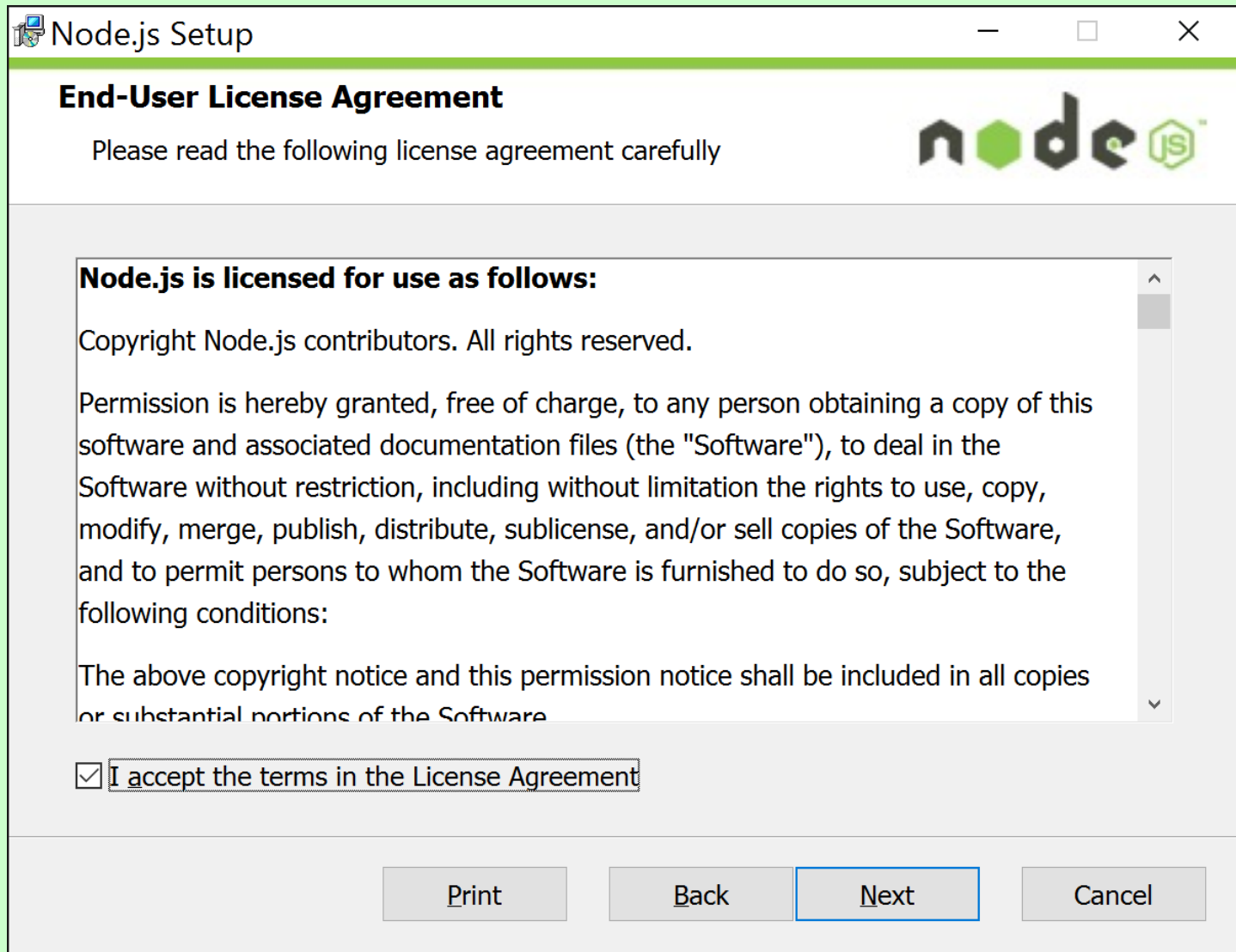
Installing Node.js

- ❑ Install 64bit node-v4.2.4-x64.msi file.



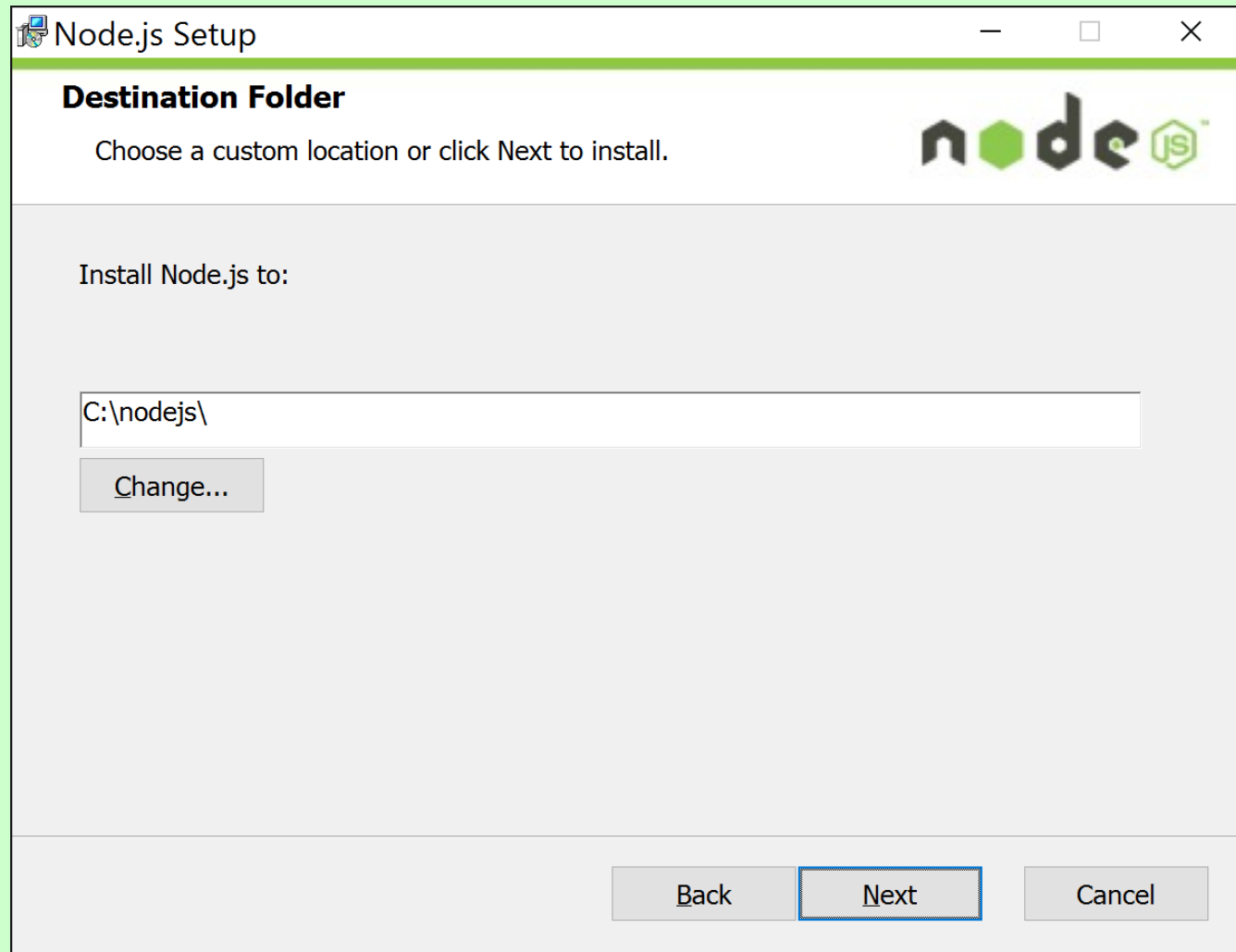


Installing Node.js



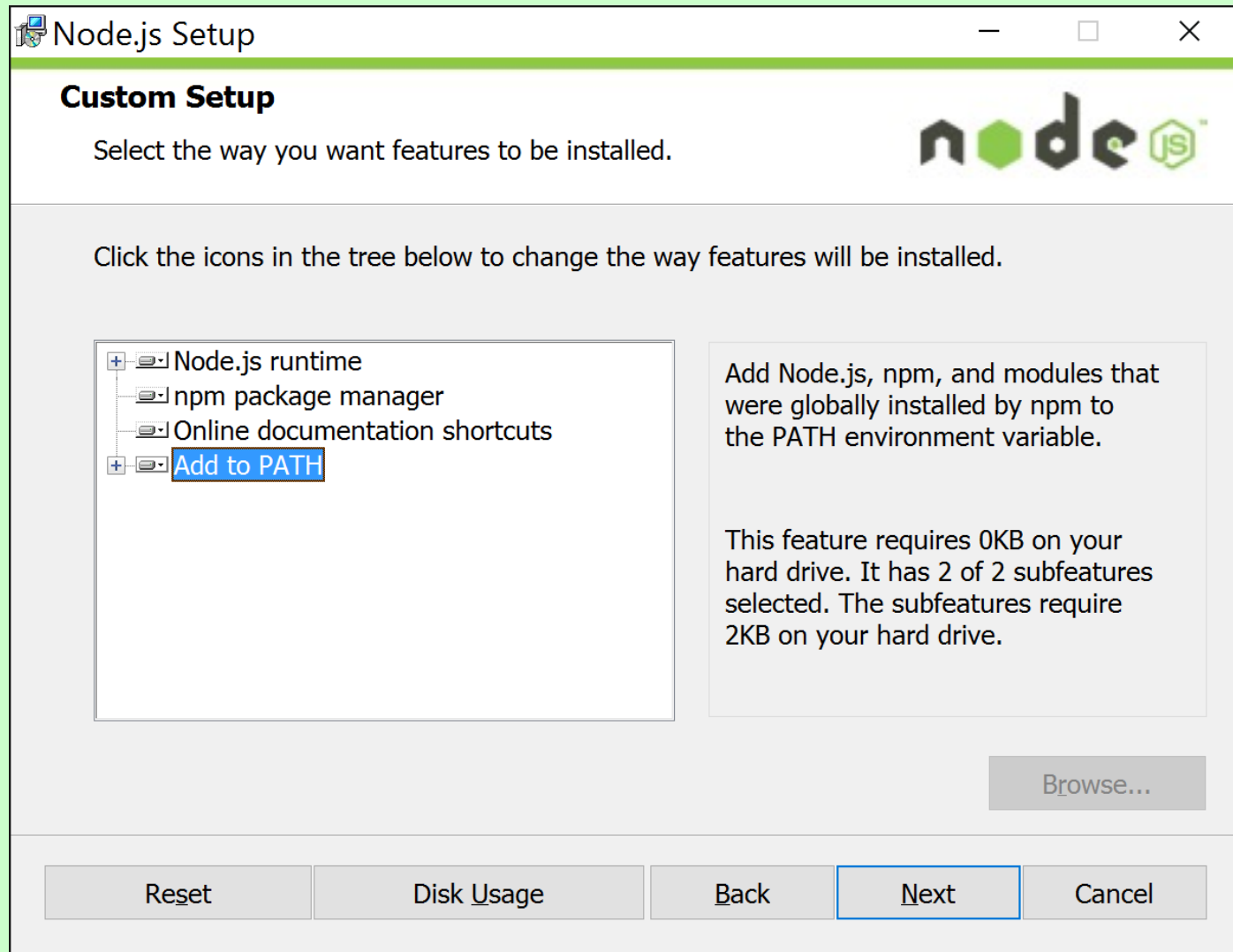


Installing Node.js



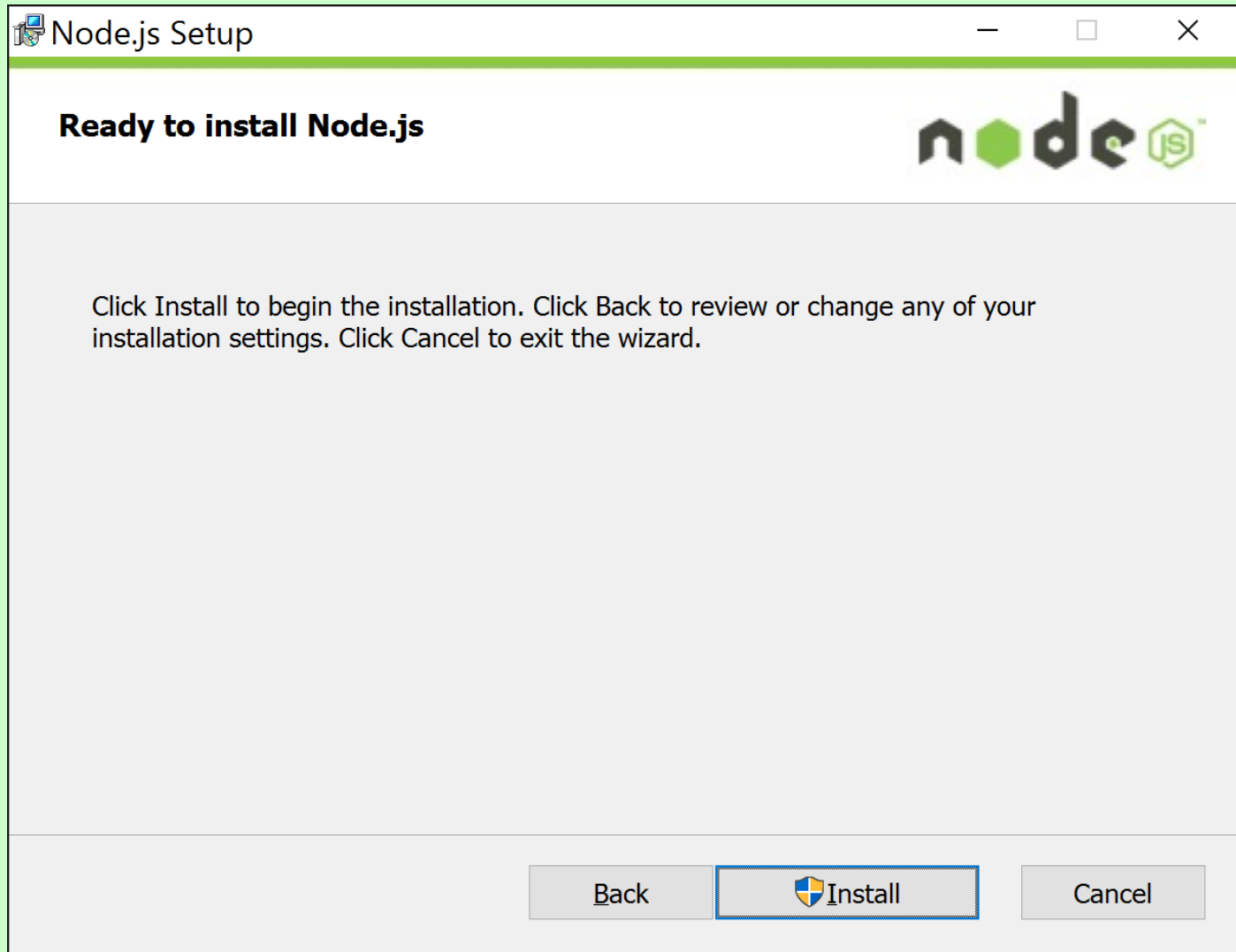


Installing Node.js



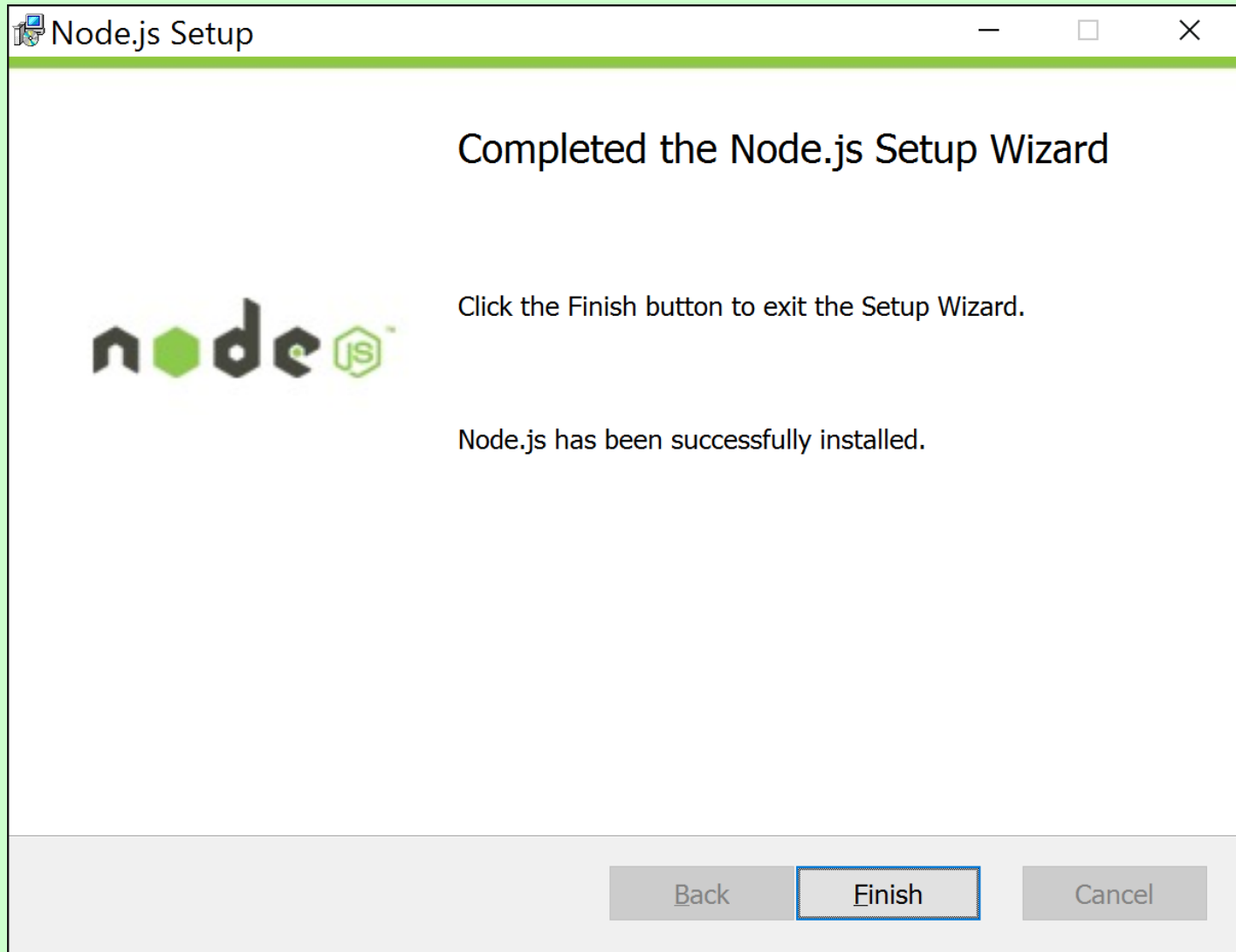


Installing Node.js





Installing Node.js





Running Node.js

- ❑ Just run node from command prompt:

```
Node.js command prompt - node
C:\nodejs>node
> console.log('Node.js is up and running!');
Node.js is up and running!
undefined
> █
```



Running Node.js

- ❑ Create a JavaScript file `application.js` with this line:
`console.log('Node.js is up and running!');`

```
Node.js command prompt
C:\nodejs>node application.js
Node.js is up and running!
C:\nodejs>
```




Node.js prompt commands

Command Prompt - node

```
C:\nodejs>node
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor      Enter editor mode
.exit       Exit the repl
.help       Print this help message
.load       Load JS from a file into the REPL session
.save       Save all evaluated commands in this REPL session to a file
>
```



Node Package Manager

- ❑ NPM is the best way to install, update, and remove Node.js modules
- ❑ NPM has the following main features:
 - A registry of packages to browse, download, and install third-party modules
 - A CLI tool to manage local and global packages
- ❑ NPM is installed during the Node.js installation process.



Using NPM

- ❑ NPM has two installation modes: **local** and **global**.
- ❑ The **default local mode** is used more often and installs the third-party packages in a local **node_modules** folder placed inside your application folder.
- ❑ It has no effect system-wise, and is used to install the packages your application needs, without polluting your system with unnecessary global files.
- ❑ The global mode is used to install packages you want Node.js **to use globally**.
- ❑ Will demonstrate the use of NPM by installing Express



Using NPM

- ❑ The global mode will usually install the packages in the
C:\Users\%USERNAME%\AppData\Roaming\npm\node_modules folder, **making it available to any Node.js application running on the system.**
- ❑ To install a package using the `npm` run the following command:

\$ npm install <Package Unique Name>

- ❑ Installing a module globally is similar to its local counterpart, but you'll have to add the `-g` flag as follows:

\$ npm install -g <Package Unique Name>



Using NPM

- ❑ For example, to locally install Express, you'll need to **navigate to your application folder** and issue the following command:

\$ npm install express

- ❑ NPM supports a wide range of semantic versioning, so to install a specific version of a package, you can use the **npm** install command as follows:

\$ npm install <Package Unique Name>@<Package Version>

- ❑ For instance, to install the second major version of the Express package, you'll need to issue the following command:

\$ npm install express@2.x



Using NPM

☐ Removing a package using NPM

- ☐ To remove an installed package, you'll have to navigate to your application folder and run the following command:

\$ npm uninstall < Package Unique Name>

- ☐ NPM will then look for the package and try to remove it from the local node_modules folder.
- ☐ To remove a global package, you'll need to use the -g flag as follows:

\$ npm uninstall -g < Package Unique Name>



Using NPM

☐ Updating a package using NPM

- ☐ To update a package to its latest version, issue the following command:

\$ npm update < Package Unique Name>

- ☐ NPM will download and install the latest version of this package even if it doesn't exist yet.
- ☐ To update a global package, use the following command:

\$ npm update -g < Package Unique Name>



Managing dependencies using the `package.json` file

- ❑ NPM allows you to use a **configuration file** named **`package.json`** in the root folder of your application.
- ❑ In your `package.json` file, you'll be able to **define various metadata properties of your application**, including properties such as the **name**, **version**, and **author** of your application.
- ❑ This is also where you **define your application dependencies**.



Managing dependencies using the `package.json` file

- ❑ The **`package.json`** file is basically a JSON file that contains the **different attributes you'll need to describe your application properties**.
- ❑ An application using the latest **Express** and **Grunt** packages will have a **`package.json`** file as follows:

```
{  
  "name" : "MEAN",  
  "version" : "0.0.1",  
  "dependencies" : {  
    "express" : "latest",  
    "grunt" : "latest"  
  }  
}
```



Creating a package.json file

- ❑ While you can manually create a `package.json` file, an easier approach would be to use the **npm init** command.
- ❑ To do so, use your command-line tool and issue the following command:

\$ npm init

- ❑ NPM will ask you a few questions about your application and will automatically **create a new package.json file** for you.
- ❑ A sample process should look similar to the following screenshot:



Creating a package.json file

cmd Command Prompt

```
C:\Classes\COMP308\Examples\Node>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> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (Node) MEAN
Sorry, name can no longer contain capital letters.
name: (Node) mean
version: (1.0.0) 0.0.1
description: my mean app
entry point: (index.js) server.js
test command:
git repository:
keywords: MongoDB, Express, AngularJS, NodeJS
author: ilia nika
license: (ISC) MIT
About to write to C:\Classes\COMP308\Examples\Node\package.json:
{
  "name": "mean",
  "version": "0.0.1",
  "description": "my mean app",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "MongoDB",
    "Express",
    "AngularJS",
    "NodeJS"
  ],
  "author": "ilia nika",
  "license": "MIT"
}

Is this ok? (yes) yes
```



Installing the **package.json** dependencies

- ❑ After creating your **package.json** file, you'll be able to install your application dependencies by navigating to your application's root folder and using the **npm** install command as follows:

\$ npm install

- ❑ NPM will automatically detect your **package.json** file and will install all your application dependencies, placing them under a local **node_modules** folder.
- ❑ An alternative and sometimes better approach to install your dependencies is to use the following **npm** update command:

\$ npm update



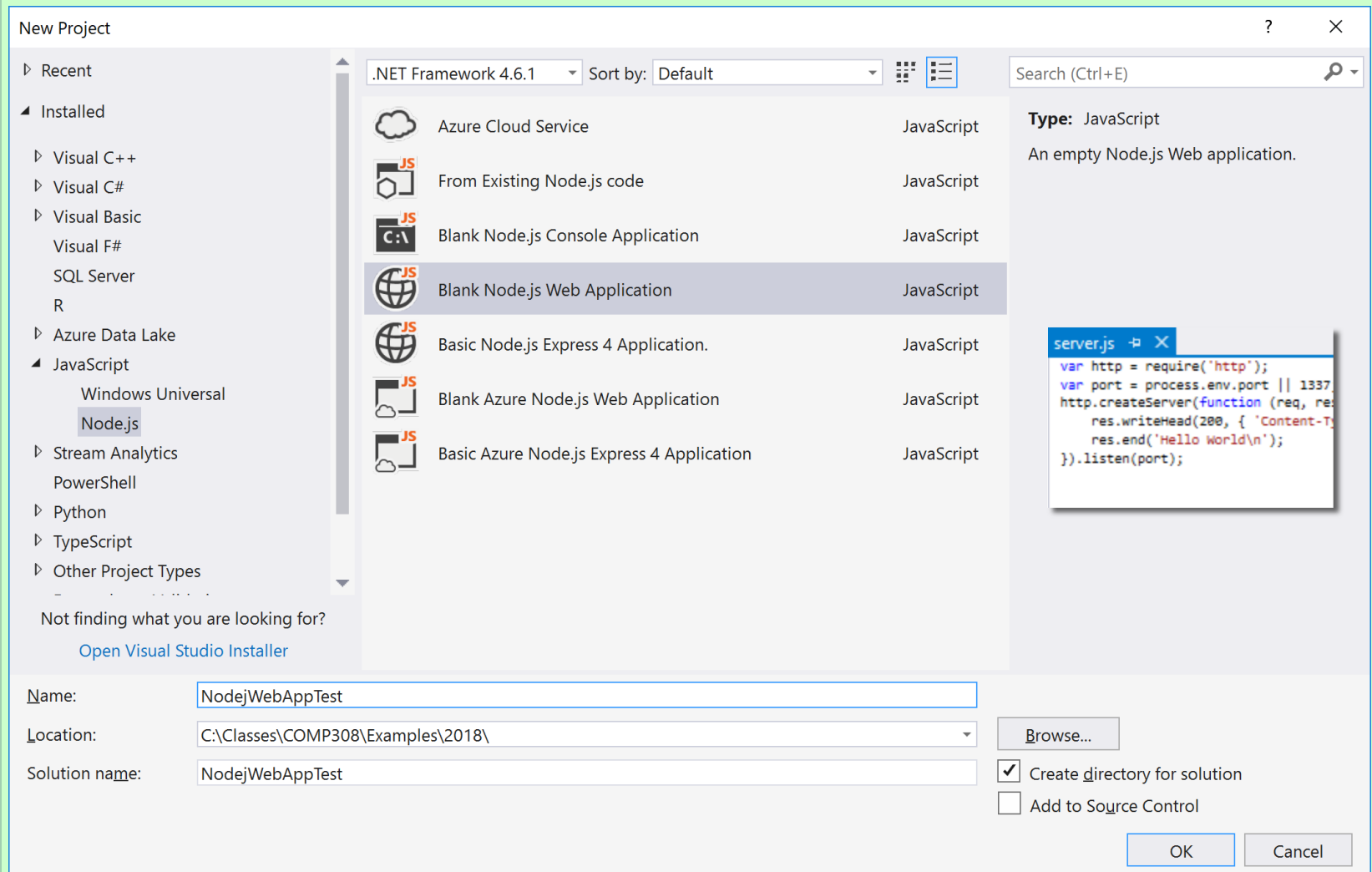
Updating the package.json file

- ❑ Another robust feature of the **npm** install command is the ability to **install a new package and save the package information as a dependency in your package.json file.**
- ❑ This can be accomplished using the **--save** optional flag when installing a specific package.
- ❑ For example, to install the latest version of Express and save it as a dependency, you can issue the following command:

```
$ npm install express --save
```

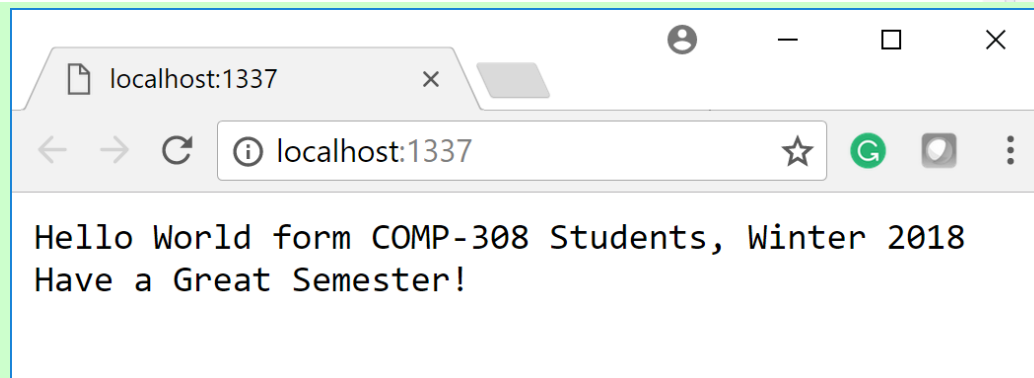
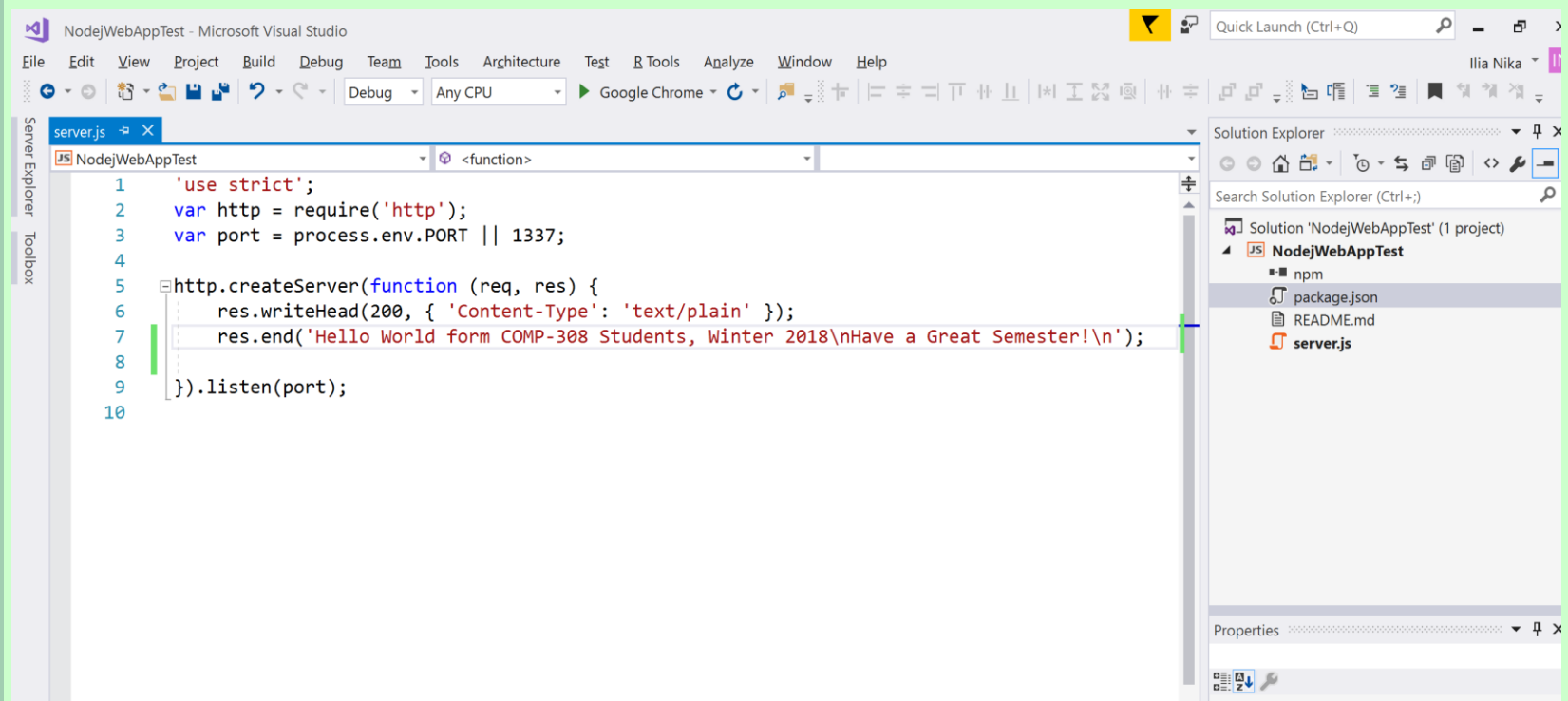


Creating Node.js Apps in VS 2017





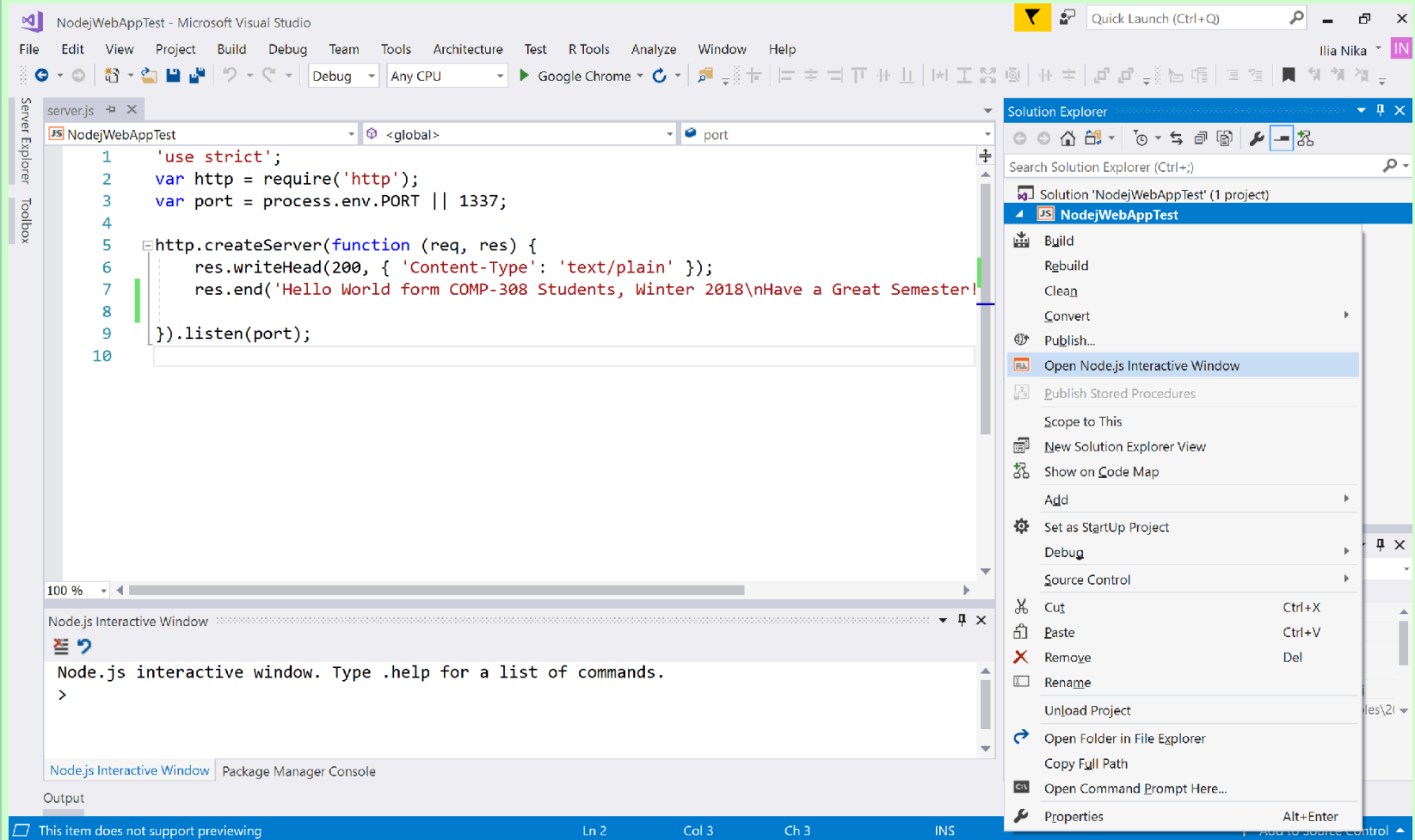
Creating Node.js Apps in VS 2017





Node.js Interactive Window

❑ Right click on the project and select it:





Installing Express

❑ In Node.js Interactive Window type:
`.npm install express`

NodejWebAppTest - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test R Tools Analyze Window Help

Debug Any CPU Google Chrome

package.json server.js

Schema: http://json.schemastore.org/package

```
1 {
2   "name": "nodej-web-app-test",
3   "version": "0.0.0",
4   "description": "NodejWebAppTest",
5   "main": "server.js",
6   "author": {
7     "name": "inika"
8   },
9   "dependencies": {
10     "express": "^4.16.2"
11   }
12 }
13
```

100 %

Node.js Interactive Window

Node.js interactive window. Type .help for a list of commands.

```
> .npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
+ express@4.16.2
added 49 packages in 2.351s
npm WARN nodej-web-app-test@0.0.0 No repository field.
npm WARN nodej-web-app-test@0.0.0 No license field.
install express successfully completed
>
```

Output

Solution Explorer

Search Solution Explorer (Ctrl+):

Solution 'NodejWebAppTest' (1 project)

- JS NodejWebAppTest
 - npm
 - express@4.16.2
 - package.json
 - README.md
 - server.js

Properties



References

- ❑ Textbook
- ❑ <https://nodejs.org/en/>
- ❑ <https://nodejs.org/api/cli.html>
- ❑ <https://nodejs.org/api/repl.html>
- ❑ <http://expressjs.com/>
- ❑ <https://angularjs.org/>
- ❑ <https://www.mongodb.org/>
- ❑ <https://www.visualstudio.com/vs/node-js/>