Exercise: Computer Systems and Software – npm, Node.js

Problems for exercises and homework for the "Software Technologies" course @ Software University.

1. What is it and why you need it

The NPM – Node Package Manager is the default package manager for Node.js. **Node.js** is an open-source, cross-platform JavaScript runtime environment that allows developers to build server-side applications using JavaScript. It can also be useful for manual testers when working on a web application that is built using Node.js. Running the application locally and test it in a development environment will require Node.js installed.

2. How to install

0. Prerequisites

Here's what you need to go ahead with Node.js and NPM.

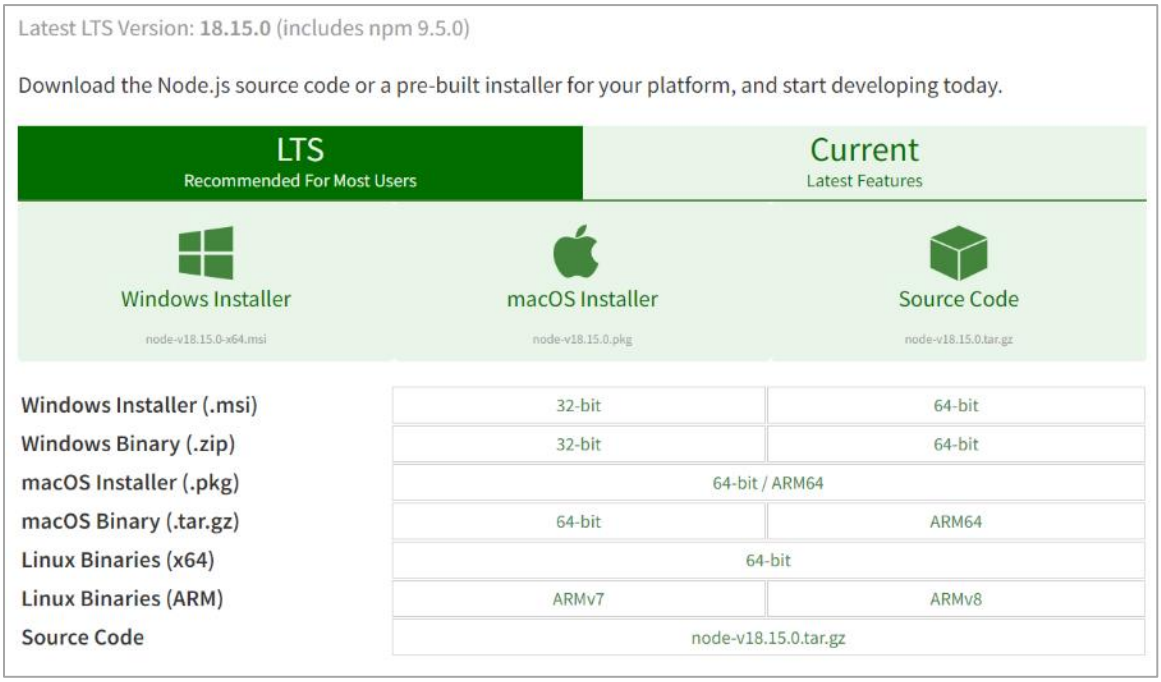
Hardware Requirements

RAM 4GB

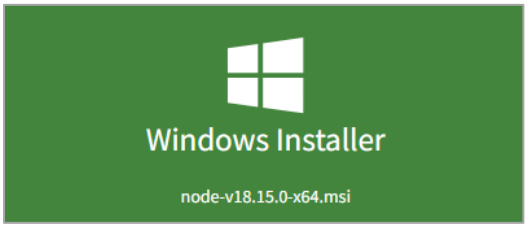
CPU Intel Core i3TM i3 HQ CPU @2.50 GHz

1. Installing via official website

Navigate to https://nodejs.org/en/download

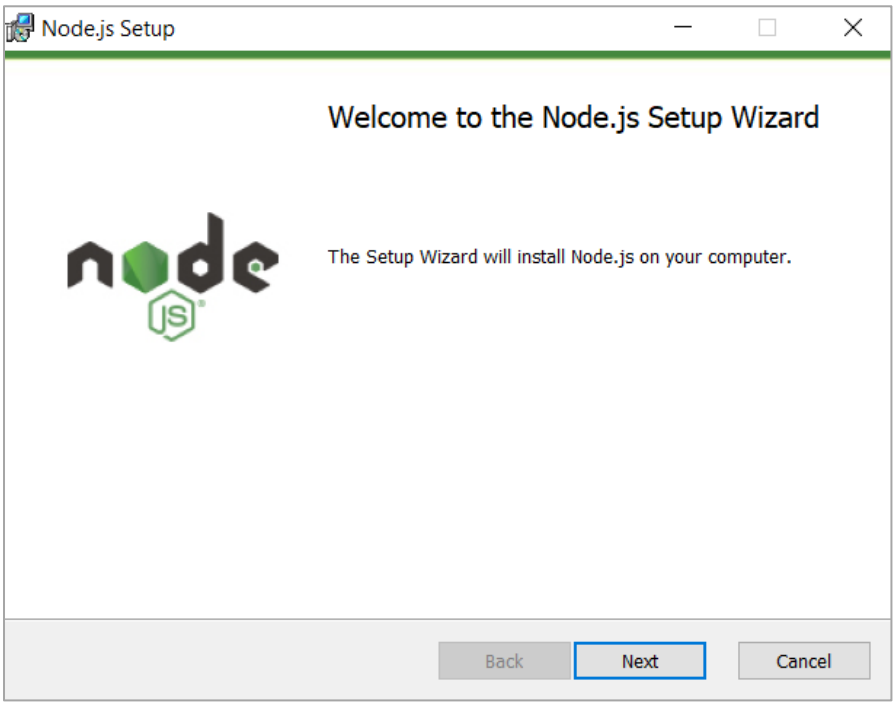


2. Choose the **appropriate installer**. Here, we are choosing the 64-bit version of the Node.js installer.

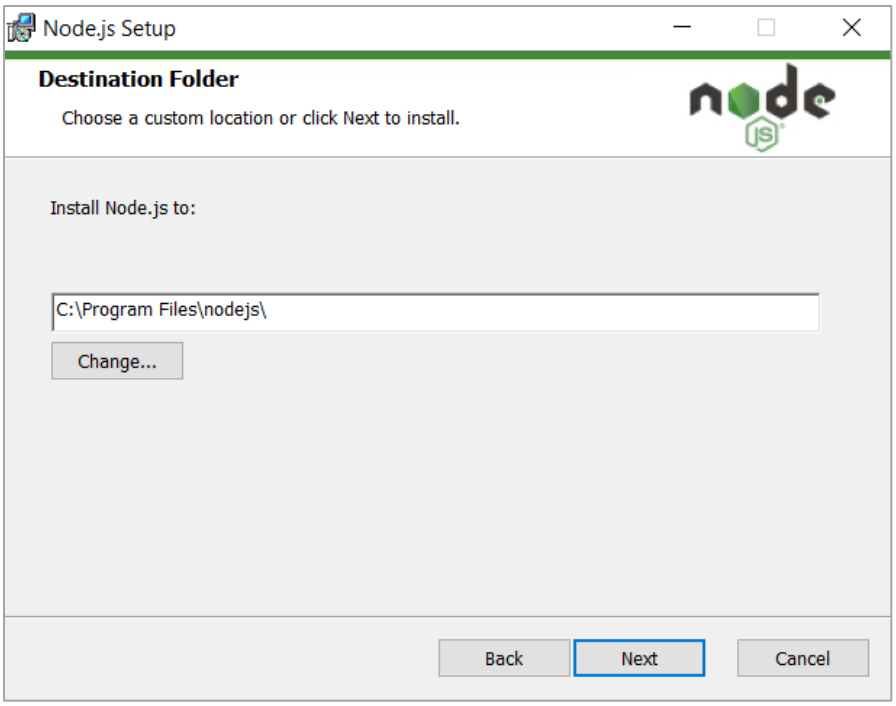


The LTS (Long-term Support) version is highly recommended for you. Now .msi file will be downloaded to your browser.

3. Double click on .msi binary files to initiate the installation process. You will get a welcome message on your screen and click the "Next" button. The installation process will start.



4. Accept the "Licence Agreement" and **choose the desired path** where you want to install Node.js.



5. The following features will be **installed by default:**

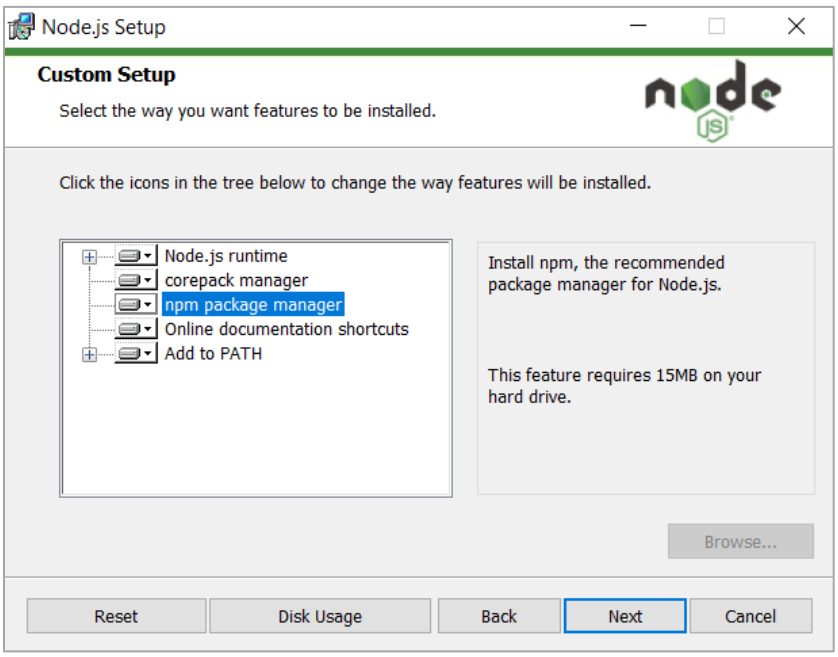
Node.js runtime

Corepack manager

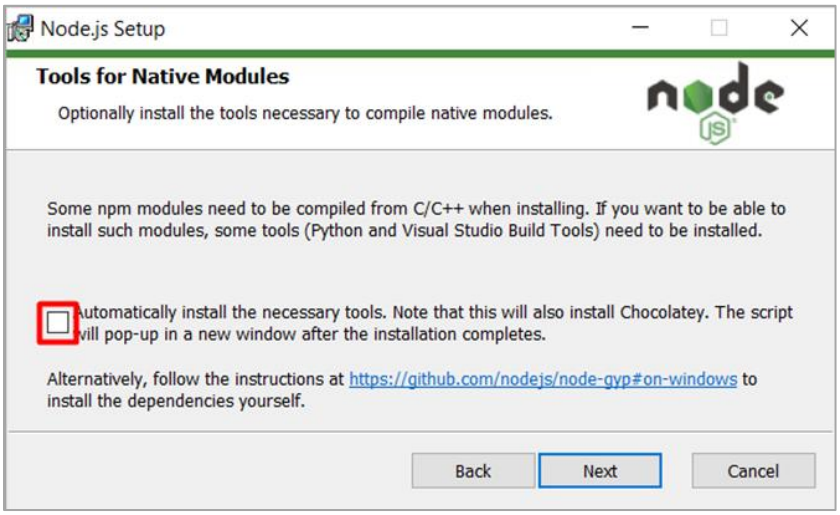
Npm package manager

Online documentation shortcuts

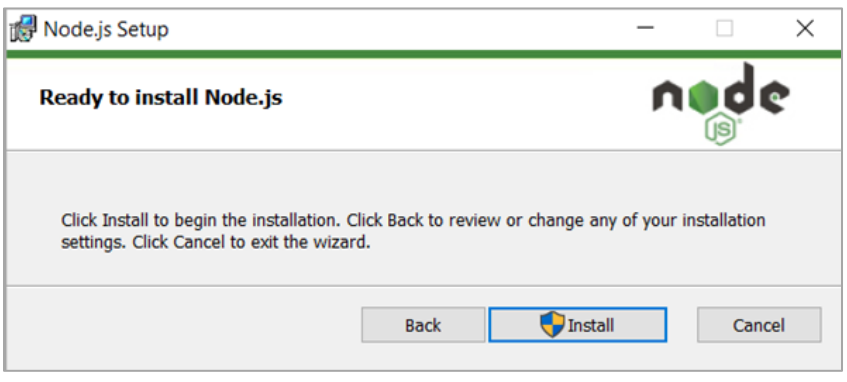
Add to Path



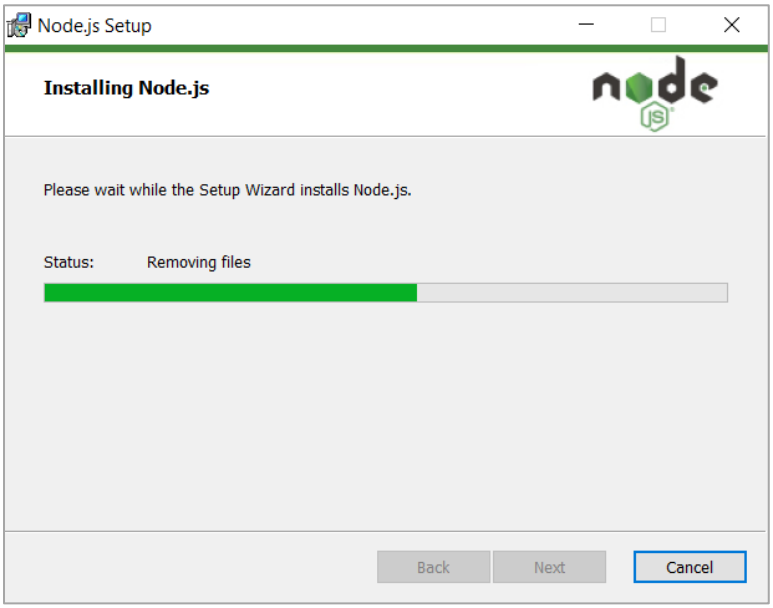
6. Do NOT install the "Tools for native modules". You won't need them for now. Make sure the pointed option is unchecked.



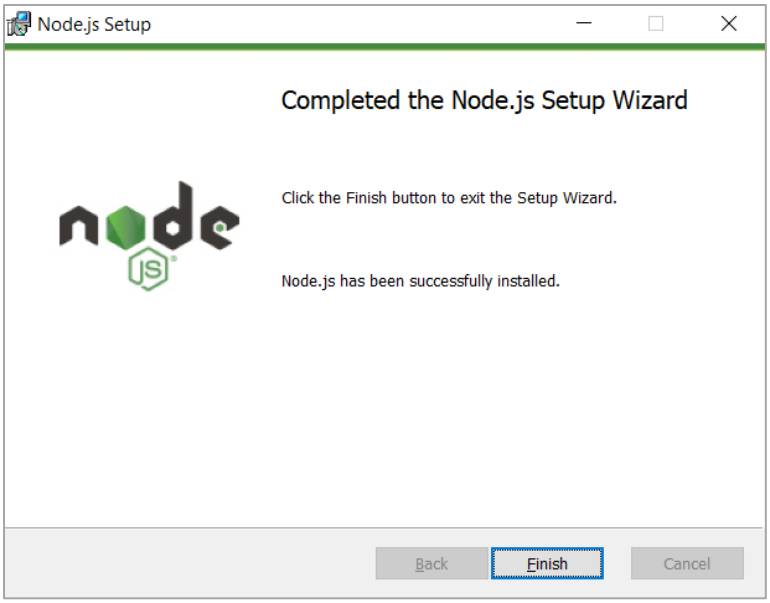
7. Click Install



8. When **prompted by Windows, give permission to make changes**. Installation now should begin.

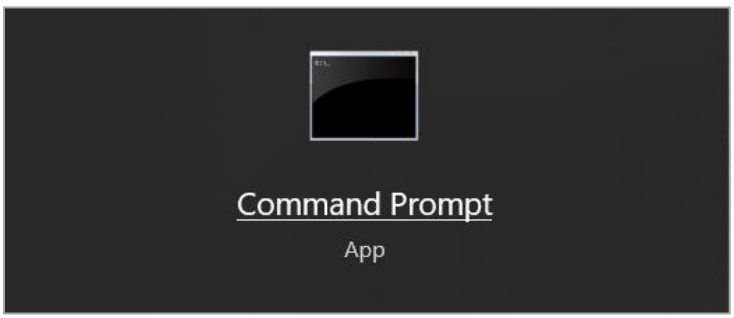


9. Click Finish to exit the setup.



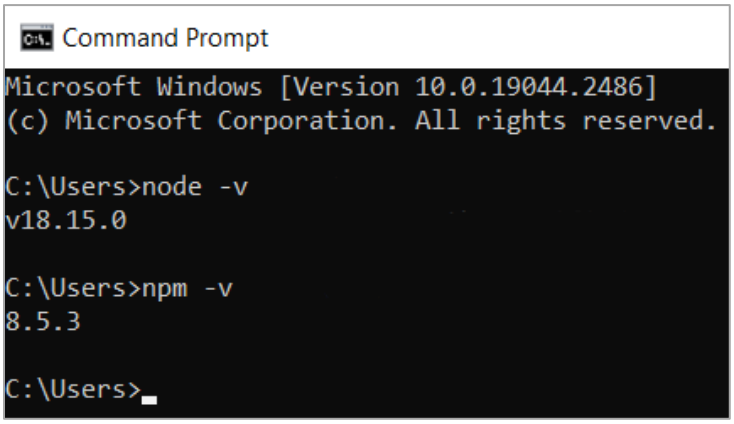
3. Check Node.js and NPM Version

To check whether you have installed everything correctly or not, let's verify it with "Command Prompt".



To confirm Node installation, type **node -v** command.

To confirm NPM installation, type **npm -v** command.

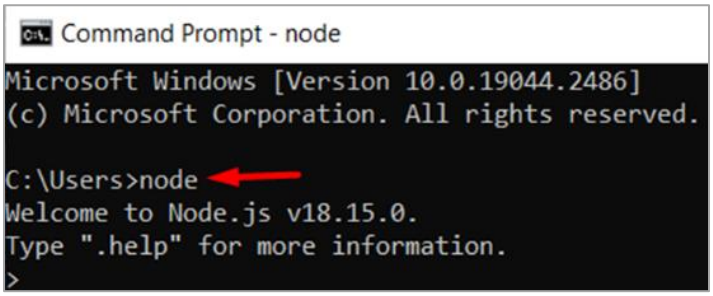


You don't need to worry if you see different version numbers as Node and NPM are updated frequently.

4. Simple expressions

Node.js comes with virtual environment called REPL (aka Node shell). REPL stands for Read-Eval-Print-Loop. It is a quick and easy way to test simple Node.js/JavaScript code.

To **launch the REPL (Node shell)**, open command prompt (in Windows) or terminal (in Mac or UNIX/Linux) and type **node** as shown below. **It will change the prompt to > in Windows and MAC.**

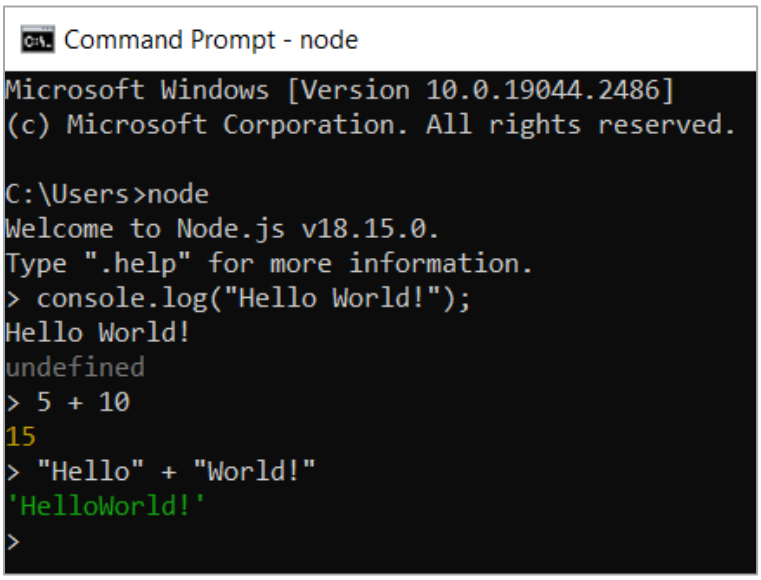


You can now test pretty much any Node.js/JavaScript expression in REPL.

5 + 10 will display 15 in new line**.**

You can print something, for example **console.log("Hello World!").**

The + operator also concatenates strings as in browser's JavaScript. So **"Hello" + "World!"** becomes 'HelloWorld!'**.**



You can also **define variables** and **perform some operation** on them.

```
> var x=10, y=20;
undefined
> x + y
30
>
```

If you need to write multi line JavaScript expression or function then **just press Enter whenever you want to write something in the next line** as a continuation of your code. The REPL terminal will display three dots (...), it means you can continue on next line. Write .**break to get out of continuity mode**.

For example, you can define a function and execute it.

```
> function multiply(x, y)
... {
... return x*y;
... }
undefined
> multiply(10, 20)
200
>
```

To exit from the REPL terminal, **press Ctrl + C twice** or write .exit and press Enter.

```
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
```