

# Full Stack Web Development

## WS2019 - BSA5 Ausgewählte Kapitel

Alija Sasic

[sasic@technikum-wien.at](mailto:sasic@technikum-wien.at)

Smart Homes and Assistive Systems

October 14, 2019

# Outline

1

Node.js

# Section

## 1 Node.js

- 1.1 History
- 1.2 Installation
- 1.3 Applications
- 1.4 Basics
- 1.5 Server

# Node.js - Ryan Dahl [2, 3]

- Born and raised in San Diego, California.
- Studied mathematics at the UC San Diego.
- Dropped out of the [PhD](#) program and bought one-way ticket to South America, where he found a job as web developer.
- Started the work on [Node.js](#) in 2009.
- Stepped away from active development on Node.js in 2012.
- Started the work on [Deno](#) in 2018.



Figure 1: Ryan Dahl [1]



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js

- JavaScript runtime
- Cross-platform
- Single-threaded
- Asynchronous and event driven
- Fast (Google's JavaScript Engine [V8](#))



Figure 2: Node.js Logo [4]



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History I [2]

- 2009: In January, Ryan Dahl starts work on a JavaScript runtime named Node.js based on Google's V8.
- 2009: In September, Mark Schuelter starts work on a Node Package Manager, further known as `npm`.
- 2009: In November, Ryan Dahl presents Node.js at the JSConf EU in Berlin [5].
- 2011: v0.6.0 is released adding support for Windows OSs.
- 2011: v0.6.3 ships `npm` with installation.



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History I [2]

- 2009: In January, Ryan Dahl starts work on a JavaScript runtime named Node.js based on Google's V8.
- 2009: In September, Mark Schuelter starts work on a Node Package Manager, further known as [npm](#).
- 2009: In November, Ryan Dahl presents Node.js at the JSConf EU in Berlin [5].
- 2011: v0.6.0 is released adding support for Windows OSs.
- 2011: v0.6.3 ships [npm](#) with installation.



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History I [2]

- 2009: In January, Ryan Dahl starts work on a JavaScript runtime named Node.js based on Google's V8.
- 2009: In September, Mark Schuelter starts work on a Node Package Manager, further known as [npm](#).
- 2009: In November, Ryan Dahl presents Node.js at the JSConf EU in Berlin [5].
- 2011: v0.6.0 is released adding support for Windows OSs.
- 2011: v0.6.3 ships [npm](#) with installation.



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History I [2]

- 2009: In January, Ryan Dahl starts work on a JavaScript runtime named Node.js based on Google's V8.
- 2009: In September, Mark Schuelter starts work on a Node Package Manager, further known as [npm](#).
- 2009: In November, Ryan Dahl presents Node.js at the JSConf EU in Berlin [5].
- 2011: v0.6.0 is released adding support for Windows OSs.
- 2011: v0.6.3 ships [npm](#) with installation.



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History I [2]

- 2009: In January, Ryan Dahl starts work on a JavaScript runtime named Node.js based on Google's V8.
- 2009: In September, Mark Schuelter starts work on a Node Package Manager, further known as [npm](#).
- 2009: In November, Ryan Dahl presents Node.js at the JSConf EU in Berlin [5].
- 2011: v0.6.0 is released adding support for Windows [OSs](#).
- 2011: v0.6.3 ships [npm](#) with installation.



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

# Node.js - History II [2]

- 2014: Community fork of Node.js, called io.js.
- 2015: In February, the Node.js Foundation is established, supervising future development of io.js.
- 2015: In June, both projects, Node.js and io.js, join the Node.js Foundation. Future development is guided by a *Technical Steering Committee* (TSC).
- 2015: In September, a joined version (v4.0.0) is released. Regular release cycles. Even version numbers are *Long Term Stable* (LTS) releases.



Alija Sasic  
Web Development

## Node.js

- History
- Installation
- Applications
- Basics
- Server

# Node.js - History II [2]

- 2014: Community fork of Node.js, called io.js.
- 2015: In February, the **Node.js Foundation** is established, supervising future development of io.js.
- 2015: In June, both projects, Node.js and io.js, join the Node.js Foundation. Future development is guided by a *Technical Steering Committee (TSC)*.
- 2015: In September, a joined version (v4.0.0) is released. Regular release cycles. Even version numbers are *Long Term Stable (LTS)* releases.



Alija Sasic  
Web Development

## Node.js

- History
- Installation
- Applications
- Basics
- Server

# Node.js - History II [2]

- 2014: Community fork of Node.js, called io.js.
- 2015: In February, the [Node.js Foundation](#) is established, supervising future development of io.js.
- 2015: In June, both projects, Node.js and io.js, join the Node.js Foundation. Future development is guided by a *Technical Steering Committee (TSC)*.
- 2015: In September, a joined version (v4.0.0) is released. Regular release cycles. Even version numbers are *Long Term Stable (LTS)* releases.



Alija Sasic  
Web Development

## Node.js

- History
- Installation
- Applications
- Basics
- Server

# Node.js - History II [2]

- 2014: Community fork of Node.js, called io.js.
- 2015: In February, the [Node.js Foundation](#) is established, supervising future development of io.js.
- 2015: In June, both projects, Node.js and io.js, join the Node.js Foundation. Future development is guided by a *Technical Steering Committee (TSC)*.
- 2015: In September, a joined version (v4.0.0) is released. Regular release cycles. Even version numbers are *Long Term Stable (LTS)* releases.



Alija Sasic  
Web Development

## Node.js

- History
- Installation
- Applications
- Basics
- Server

# Node.js - Versions

Since the establishment of the Node.js Foundation, Node.js is released regularly.

Release	Codename	Initial Release	End of Life
4.x LTS	Argon	08.09.2015	April 2018
5.x		29.10.2015	June 2016
6.x LTS	Boron	26.04.2016	April 2019
7.x		25.10.2016	June 2017
8.x LTS	Carbon	30.05.2017	December 2019
9.x		01.10.2017	June 2018
10.x LTS	Dubnium	24.04.2018	April 2021
11.x		23.10.2018	June 2019
12.x LTS		23.04.2019	April 2022

Table 1: Node.js release schedule.



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Installation



The simplest way to install Node.js is install it with a package manager or to download the current version from the [Node.js download page](#).

Node.js provides installers, prebuild binaries for different operating systems and architectures, together with other options like Docker container (cf. Figure 3).

- Alija Sabic  
.....  
[Web Development](#)
- [Node.js](#)
- [History](#)
- Installation**
- [Linux, Windows, macOS](#)
- [Custom Build](#)
- [Node Version Manager](#)
- [Applications](#)
- [Basics](#)
- [Server](#)

# Node.js - Installation

LTS Recommended For Most Users	Current Latest Features																
 <b>Windows Installer</b> node-v10.16.3-x86.msi	 <b>macOS Installer</b> node-v10.16.3.pkg																
<a href="#">Windows Installer (.msi)</a> <a href="#">Windows Binary (.zip)</a> <a href="#">macOS Installer (.pkg)</a> <a href="#">macOS Binary (.tar.gz)</a> <a href="#">Linux Binaries (x64)</a> <a href="#">Linux Binaries (ARM)</a> <a href="#">Source Code</a>	<table border="1"><tr><td>32-bit</td><td>64-bit</td></tr><tr><td>32-bit</td><td>64-bit</td></tr><tr><td colspan="2">64-bit</td></tr><tr><td colspan="2">64-bit</td></tr><tr><td colspan="2">64-bit</td></tr><tr><td>ARMv6</td><td>ARMv7</td><td>ARMv8</td></tr><tr><td colspan="3">node-v10.16.3.tar.gz</td></tr></table>	32-bit	64-bit	32-bit	64-bit	64-bit		64-bit		64-bit		ARMv6	ARMv7	ARMv8	node-v10.16.3.tar.gz		
32-bit	64-bit																
32-bit	64-bit																
64-bit																	
64-bit																	
64-bit																	
ARMv6	ARMv7	ARMv8															
node-v10.16.3.tar.gz																	
<b>Additional Platforms</b>	<table border="1"><tr><td>64-bit</td></tr><tr><td>Official Node.js Docker Image</td></tr><tr><td>64-bit</td></tr><tr><td>64-bit</td></tr><tr><td>64-bit</td></tr></table>	64-bit	Official Node.js Docker Image	64-bit	64-bit	64-bit											
64-bit																	
Official Node.js Docker Image																	
64-bit																	
64-bit																	
64-bit																	
<a href="#">SmartOS Binaries</a> <a href="#">Docker Image</a> <a href="#">Linux on Power Systems</a> <a href="#">Linux on System z</a> <a href="#">AIX on Power Systems</a>																	

Figure 3: Node.js Downloads [4]



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Installation on Linux

To install the Node.js binary in the opt/ directory, run following commands.

```
1 ~ $ cd ~/Downloads
2 ~/Downloads $ wget \
3 > https://nodejs.org/dist/v10.16.3/node-v10.16.3-linux-x64.tar.xz
4
5 --2019-10-14 00:38:55-- https://nodejs.org/...
6 Resolving nodejs.org (nodejs.org)... 104.20....
7 ...
8
9 ~/Downloads $ cd opt/
10 /opt $ sudo tar xvf ~/node-v10.16.3-linux-x64.tar.xz
11
12 [sudo] password for <username>:
13 node-v10.16.3-linux-x64/
14 ...
15
16 /opt $ sudo ln -s node-v10.16.3-linux-x64 nodejs
17 /opt $ PATH=$PATH:/opt/nodejs/bin
18 /opt $ export PATH
```

Listing 1: Install Node.js binary



Alija Sasic  
Web Development

Node.js  
History  
Installation  
Linux, Windows, macOS  
Custom Build  
Node Version Manager  
Applications  
Basics  
Server

# Node.js - Installation on Linux

The command to install Node.js from a package repository depends on the package manager of your distribution. On Ubuntu, run following commands.

```
1 ~ $ sudo apt-get install nodejs
2
3 Reading package list... Done
4 Building dependency tree
5 Reading state information... Done
6 The following additional packages will be installed:
7   libc-ares2 libhttp-parser2.7.1 libuv1 nodejs-doc
8 ...
```

**Listing 2:** Install Node.js with package manager



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Installation on Linux

You can add a package repository to have always the latest version when installing with the package manager.

```
1 ~ $ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
2 ~ $ sudo apt-get install -y nodejs
3
4 ## Installing the NodeSource Node.js 10.x repo...
5
6 ## Populating apt-get cache...
7 ...
```

**Listing 3:** Install Node.js with package manager



Alija Sasic  
Web Development

Node.js  
History  
Installation  
Linux, Windows, macOS  
Custom Build  
Node Version Manager  
Applications  
Basics  
Server

# Node.js - Installation on Windows

To install the Node.js binary in the C: directory, download the provided archive and extract it to the drive.

You can now start Node.js from a terminal inside of the directory.

```
1 C:\node-v10.16.3-win-x64> node.exe -v  
2  
3 v10.16.3
```

**Listing 4:** Test Node.js version

To access the Node.js binary from every location, you have to add the folder containing the binary to the PATH environment variable.



Alija Sasic  
.....  
[Web Development](#)  
[Node.js](#)  
[History](#)  
[Installation](#)  
[Linux, Windows, macOS](#)  
[Custom Build](#)  
[Node Version Manager](#)  
[Applications](#)  
[Basics](#)  
[Server](#)

# Node.js - Installation on Windows

To install the Node.js binary and setup environment variables automatically, use the provided [MSI](#) installer.

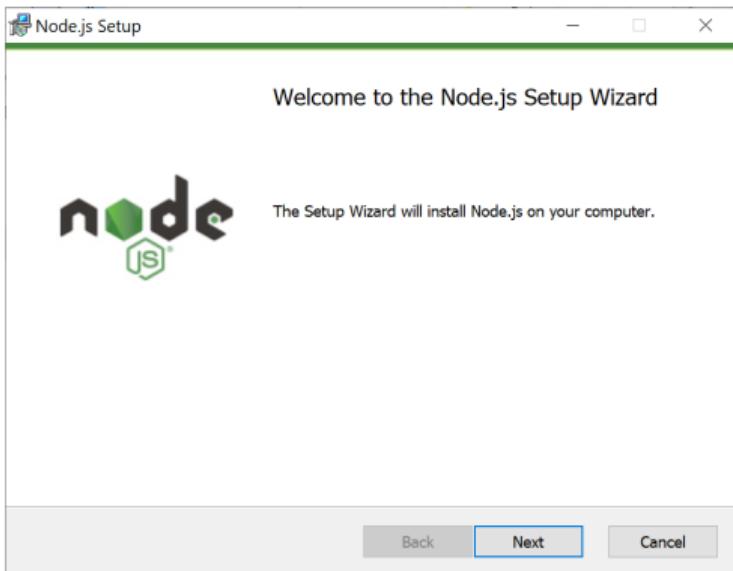


Figure 4: Start Node.js [MSI](#) installer



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Installation on Windows

After the installation is complete, you will find several programs in the Node.js start menu entry.

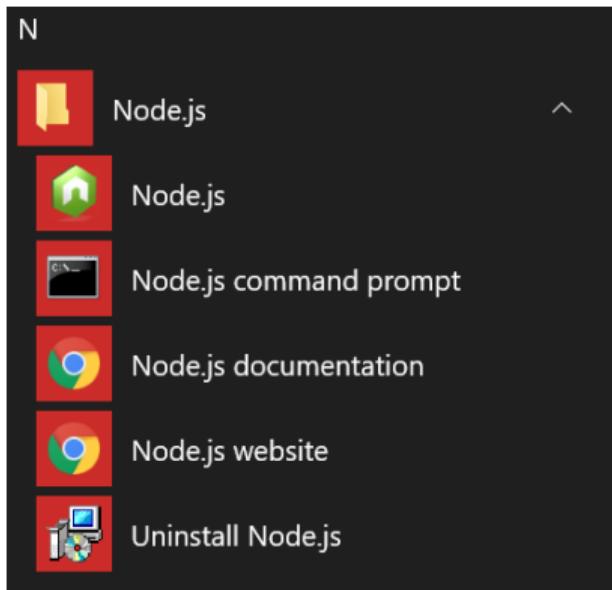


Figure 5: Start menu entry of Node.js



Alija Sabic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

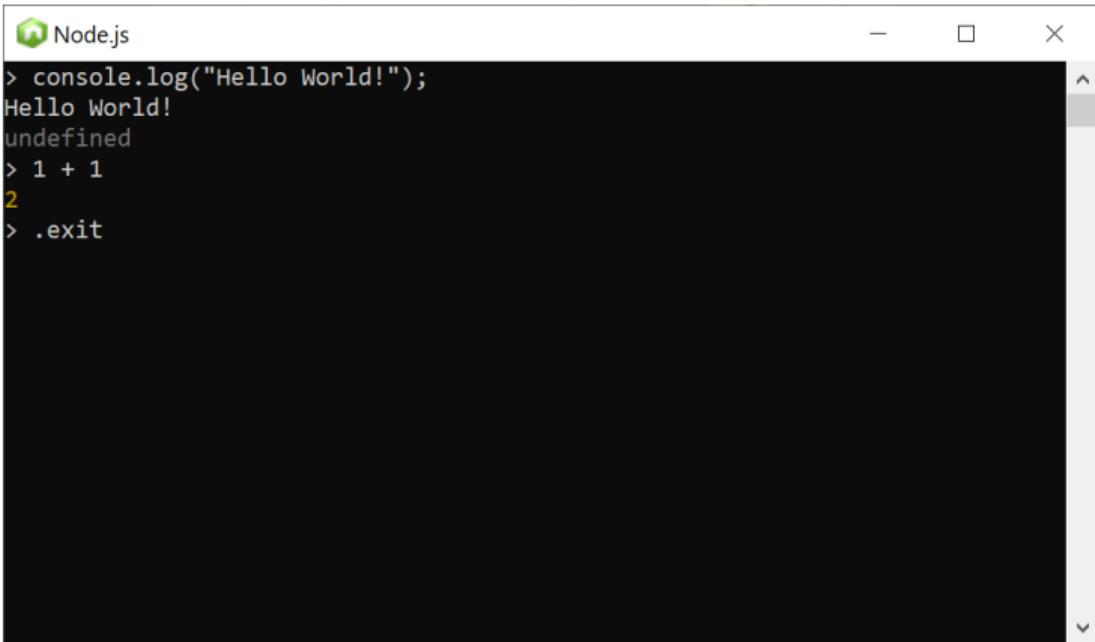
Applications

Basics

Server

# Node.js - Installation on Windows

The first entry, Node.js, starts Node.js in [REPL](#) mode.



```
Node.js
> console.log("Hello World!");
Hello World!
undefined
> 1 + 1
2
> .exit
```

Figure 6: Node.js [REPL](#) mode



Alija Sabic

[Web Development](#)

[Node.js](#)

[History](#)

[Installation](#)

[Linux, Windows, macOS](#)

[Custom Build](#)

[Node Version Manager](#)

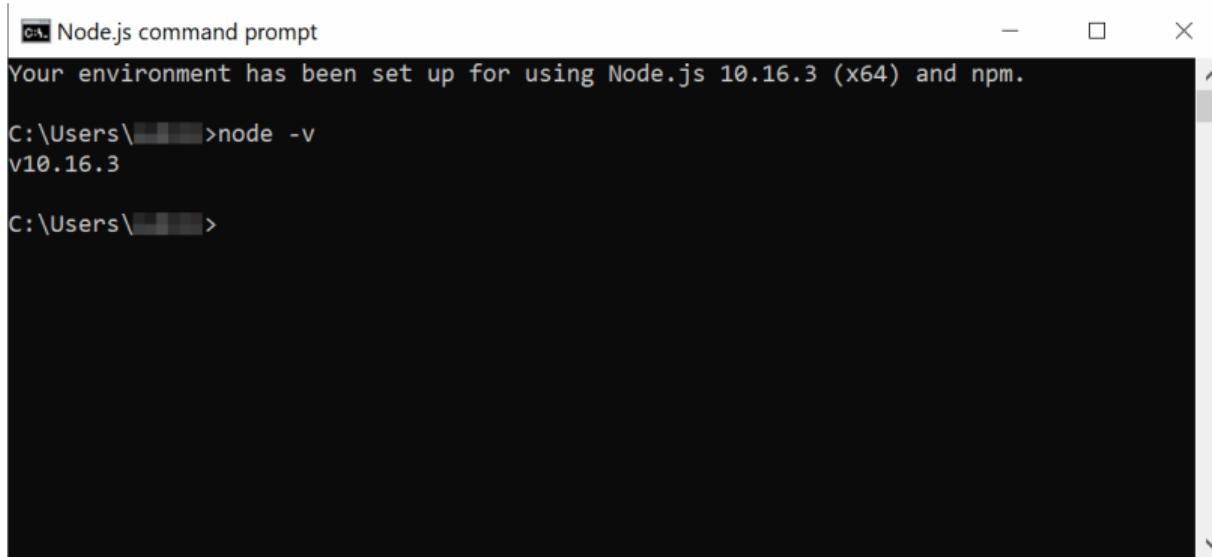
[Applications](#)

[Basics](#)

[Server](#)

# Node.js - Installation on Windows

The second entry, Node.js command prompt, opens a terminal window with a set up environment.



```
Node.js command prompt
Your environment has been set up for using Node.js 10.16.3 (x64) and npm.

C:\Users\[REDACTED]>node -v
v10.16.3

C:\Users\[REDACTED]>
```

Figure 7: Node.js command prompt



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Installation on macOS

To install the Node.js binary on macOS, run following commands.

```
1 ~ $ wget https://nodejs.org/dist/v10.16.3/node-v10.16.3-darwin-x64.tar.gz
2
3 --2019-10-14 00:38:55-- https://nodejs.org/...
4 Resolving nodejs.org (nodejs.org)... 104.20....
5 ...
6
7 ~ $ cd tar xvzf node-v10.16.3-darwin-x64.tar.xz
8
9 node-v10.16.3-linux-x64/
10 ...
11
12 ~ $ cd node-v10.16.3-darwin-x64
13 ~/node-v10.16.3-darwin-x64 $ bin/node -v
14
15 v10.16.3
```

Listing 5: Install Node.js binary



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Building from Source

Since Node.js is an open-source project, the source files can be downloaded and used to build an individual version of Node.js.

The following commands outline how to build Node.js from source on Ubuntu.

```
1 ~ $ cd Downloads
2 ~/Downloads $ wget https://nodejs.org/dist/v10.16.3/node-v10.16.3.tar.gz
3 ~/Downloads $ tar xfz node-v10.16.3.tar.gz
4 ~/Downloads $ cd node-v10.16.3
5
6 # List available options
7 ~/Downloads/node-v10.16.3 $ ./configure --help
8
9 # Example: w/o npm, w/o crypto, https, etc.
10 ~/Downloads/node-v10.16.3 $ ./configure --without-npm --without-ssl
```

Listing 6: Building Node.js from source



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Building from Source

To configure, build and install Node.js with typical defaults, run following commands.

```
1 # Configure  
2 ~/Downloads/node-v10.16.3 $ ./configure --prefix=/opt/node/node-v10.16.3  
3 ...  
4  
5 # Build  
6 ~/Downloads/node-v10.16.3 $ make -j4  
7 ...  
8  
9 ...  
10  
11 # Install  
12 ~/Downloads/node-v10.16.3 $ sudo make install  
13 ...  
14
```

**Listing 7:** Building Node.js from source



Alija Sabic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Node Version Manager

If you need to run or test your scripts with different node versions, you can download the *Node Version Manager* ([nvm](#)) from the GitHub repository <https://github.com/nvm-sh/nvm>.

```
1 ~ $ wget -qO- \  
2 > https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.0/install.sh |\  
3 > bash  
4  
5 => Downloading nvm as script to '/home/<user>/.bashrc'  
6 => Appending bash_completion source string to /home/<user>/.bashrc  
7 => Close and reopen your terminal to start using nvm or run ...  
8 ...
```

Listing 8: nvm Installation



Alija Sasic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Node Version Manager

After the installation is finished, you can use `nvm` to install and use different versions of Node.js.

```
1 # Print help
2 ~ $ nvm --help
3
4 # List available node versions
5 ~ $ nvm ls-remote
6 ~ $ nvm ls
7
8 # Install current LTS version
9 ~ $ nvm install --lts node
10
11 Downloading and installing node v10.16.3
12 ...
13
14 # Use current LTS version
15 ~ $ nvm use --lts node
16
17 Now using node v10.16.3 (npm v6.9.0)
```

Listing 9: `nvm` Usage Examples



Alija Sabic

Web Development

Node.js

History

Installation

Linux, Windows, macOS

Custom Build

Node Version Manager

Applications

Basics

Server

# Node.js - Applications

Node.js is basically used to build two types of applications.

## ■ Command Line Applications

Node.js is used for a variety of utility applications for the terminal, like css compiler and linter.

- typescript
- eslint
- vue
- sass

## ■ Server applications

Most important application field, however, is building server software.



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Command Line

Server

Basics

Server

# Node.js - Command Line

Node.js supports different modes.

To start Node.js in the **REPL** mode, simply type in `node` in the terminal.

The terminal will show a different prefix, that is `>`.

```
1 ~ $ node
2 > 1 + 1
3 2
4 > console.log("Hello World");
5 Hello World
6 undefined
7 > .exit
8 ~ $
```

**Listing 10:** Starting Node.js in **REPL** mode

You can enter expression, which will be evaluated by Node.js. Further details are explained in later sections.



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Command Line

Server

Basics

Server

# Node.js - Command Line

Node.js can also run a script from a file.

```
1 ~ $ node hello.js  
2 Hello World!  
3 ~ $
```

**Listing 11:** Running Node.js scripts

```
1 console.log("Hello World!");
```

**Listing 12:** Node.js script `hello.js`



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Command Line

Server

Basics

Server

# Node.js - Server

The next listing show a “Hello World” server application.

```
1 const http = require("http");
2
3 const server = http.createServer((req, res) => {
4   res.statusCode = 200;
5   res.setHeader("Content-Type", "text/plain");
6   res.end("Hello World\n");
7 });
8
9 const hostname = "127.0.0.1";
10 const port = 3000;
11 server.listen(port, hostname, () => {
12   console.log(`Server running at http://${hostname}:${port}/`);
13 })
```

Listing 13: Node.js server application



Alija Sobic

Web Development

Node.js

History

Installation

Applications

Command Line

Server

Basics

Server

# Node.js - Server

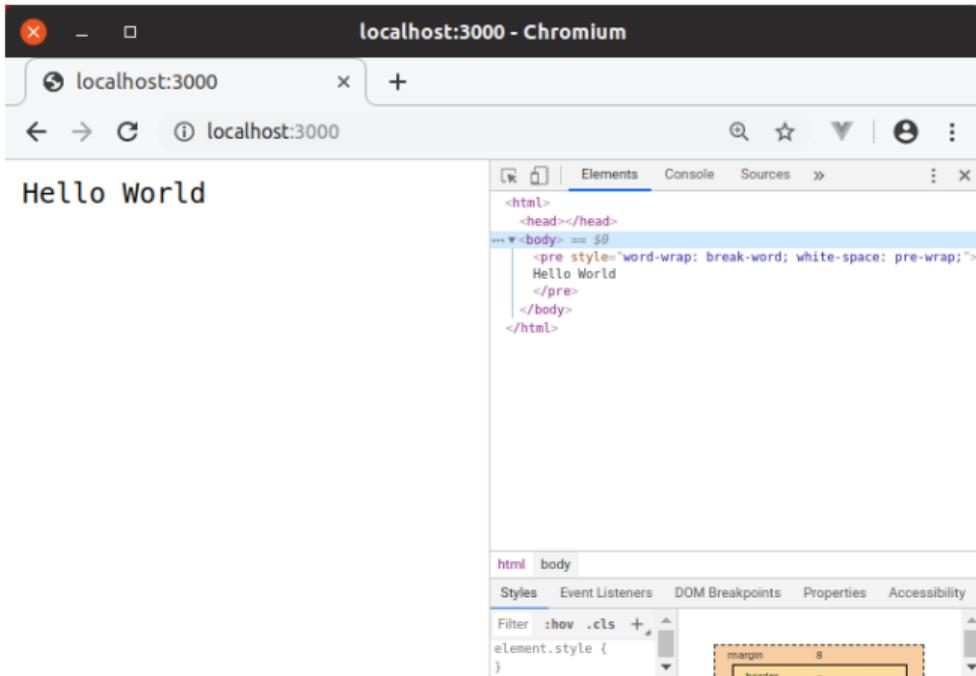


Figure 8: Node.js server application response in browser



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Command Line

Server

Basics

Server

# Server - System Calls



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Server - Blocking vs. Non-blocking



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Server - Scheduling



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - Dependencies

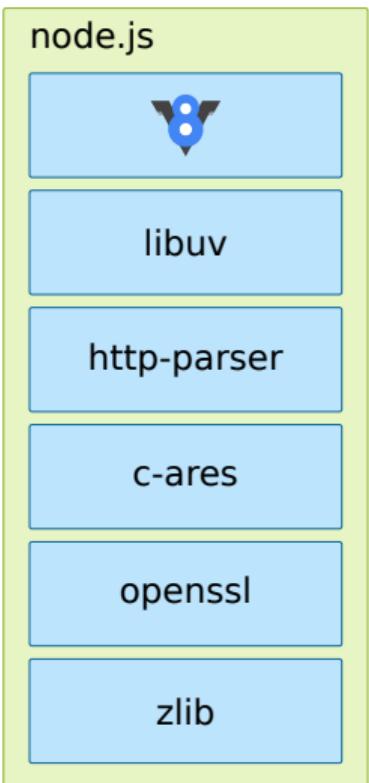


Figure 9: Node.js dependencies



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - Dependencies

## Node.js

### V8

The V8 library provides Node.js with a JavaScript engine, which Node.js controls via the V8 C++ API. V8 is maintained by Google, for use in Chrome.

### libuv

Another important dependency is libuv, a C library that is used to abstract non-blocking I/O operations to a consistent interface across all supported platforms. It provides mechanisms to handle file system, DNS, network, child processes, pipes, signal handling, polling and streaming.



Alija Sabic

Web Development

## Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - Dependencies

## Node.js

### http-parser

HTTP parsing is handled by a lightweight C library called `http-parser`. It is designed to not make any syscalls or allocations, so it has a very small per-request memory footprint.

### c-ares

For some asynchronous `DNS` requests, Node.js uses a C library called `c-ares`. It is exposed through the `DNS` module in JavaScript as the `resolve()` family of functions.



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - Dependencies

## Node.js

### openssl

OpenSSL is used extensively in both the `tls` and `crypto` modules. It provides battle-tested implementations of many cryptographic functions that the modern web relies on for security.

### zlib

For fast compression and decompression, Node.js relies on the industry-standard zlib library, also known for its use in gzip and libpng.



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - V8



Figure 10: JavaScript support across browsers and server runtimes



Alija Sabic  
b Development

# Node.js

## History

## Installation Application

# Basics

Server

## Dependencies

JavaScript Engine

## Event Loop

Server

# Node.js - V8

	Node.js	ES2015 Support	Learn more	Nightly	13.0.0	12.12.0	12.10.0	12.8.1	12.4.0	11.15.0	10.16.3	10.8.0	10.3.0	9.11.2	8.9.4	8.6.0	8.2.1	7.10.1	7.5.0	6.17.1	6.4.0	5.12.0	4.9.1	0.12.18	0.10.48
<b>optimisation</b>																									
<b>proper tail calls (tail call optimisation)</b>																									
direct recursion																									
mutual recursion																									
<b>syntax</b>																									
<b>default function parameters</b>																									
basic functionality																									
explicit undefined defers to the default																									
defaults can refer to previous params																									
arguments object interaction																									
temporal dead zone																									
separate scope																									
new Function() support																									
<b>rest parameters</b>																									
basic functionality																									
function 'length' property																									
arguments object interaction																									
can't be used in setters, new Function() support																									
<b>spread syntax for iterable objects</b>																									
with arrays, in function calls																									
with arrays, in array literals																									
with sparse arrays, in function calls																									
with sparse arrays, in array literals																									
with strings, in function calls																									
with strings, in array literals																									
astral plane strings, in function calls																									
astral plane strings, in array literals																									
with generator instances, in calls																									
with generator instances, in arrays																									
with generic iterables, in calls																									
with generic iterables, in arrays																									
with instances of iterables, in calls																									
with instances of iterables, in arrays																									
spreading non-iterables is a runtime error																									
<b>object literal extensions</b>																									
computed properties																									
shorthand properties																									
shorthand methods																									
string-keyed shorthand methods																									
computed shorthand methods																									
computed accessors																									
<b>for loops</b>																									
with arrays																									
with sparse arrays																									
with strings																									
with astral plane strings																									
with generator instances																									

Figure 11: JavaScript support across browsers and server runtimes



- Alija Sabic
- Web Development
- Node.js
- History
- Installation
- Applications
- Basics
- Server
- Dependencies
- JavaScript Engine
- Event Loop
- Server

# Node.js - Event-Loop



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Dependencies

JavaScript Engine

Event Loop

Server

# Node.js - Routing



Alija Sabic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Routing

Request

Response

File System

# Node.js - Request



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Routing

Request

Response

File System

# Node.js - Response



Alija Sasic

Web Development

## Node.js

History

Installation

Applications

Basics

Server

Routing

Request

Response

File System

# Node.js - File System



Alija Sasic

Web Development

Node.js

History

Installation

Applications

Basics

Server

Routing

Request

Response

File System

# Acronyms I

DNS Domain Name System

I/O Input/Output

LTS Long Term Stable

MSI Microsoft Installer

npm Node Package Manager

nvm Node Version Manager

OS Operating System

PhD Doctor of Philosophy (philosophiae doctor)

REPL Read Eval Print Loop

TSC Technical Steering Committee



Alija Sasic

Web Development

Acronyms

References

# References I

- [1] Ryan Dahl, “10 Things I Regret About Node.js,”  
<https://www.youtube.com/watch?v=M3BM9TB-8yA>, 2018.
- [2] S. Springer, *Node.js: Das umfassende Handbuch*, 3rd ed. Bonn: Rheinwerk Verlag, 2018. [Online]. Available:  
<http://d-nb.info/1168478928>
- [3] Wikipedia contributors, “Ryan Dahl,”  
[https://en.wikipedia.org/w/index.php?title=Ryan\\_Dahl](https://en.wikipedia.org/w/index.php?title=Ryan_Dahl), 10 2019.
- [4] Node.js Foundation, “Node.js - JavaScript Runtime,”  
<https://nodejs.org/>, 2019.



Alija Sabic

Web Development

Acronyms

References

# References II

- [5] Ryan Dahl, "Node.js,"  
<https://www.youtube.com/watch?v=EeYvFl7li9E>, 2009.



Alija Sasic

Web Development

Acronyms

References