**Installationsanleitung  
Node.js + NPM + TypeScript  
Als clientseitige Laufzeitumgebung sowie als Komponente für MS Visual Studio**

**Inhalt**

[1. Einleitung 3](#_Toc80273554)

[2. Lizenzbedingungen 3](#_Toc80273555)

[3. Installation 4](#_Toc80273556)

[3.1. Option 1: Node Versions Manager (NVM) für Windows 6](#_Toc80273557)

[3.1.1. Download: NVM Installer 8](#_Toc80273558)

[3.1.2. Installation: NVM als Befehlsbibliothek der Windows-Konsolen 9](#_Toc80273559)

[3.1.3. Installation: Node.js + NPM + Abhängigkeiten 10](#_Toc80273560)

[3.1.4. Installation: TypeScript-Plugin für MS Visual Studio 13](#_Toc80273561)

[3.1.5. Installationspfade: Node.js NVM-Version 14](#_Toc80273562)

[3.1.6. Versionscheck: Als Installationsnachweis 15](#_Toc80273563)

[3.1.7. Updates: NVM + Node.js + NPM + Abhängigkeiten + TypeScript-Plugin 16](#_Toc80273564)

[3.1.8. Ausführung: NVM-Version von Node.js 21](#_Toc80273565)

[3.1.9. Deinstallation: NVM-Version von Node.js 22](#_Toc80273566)

[3.2. Option 2: Node.js Installer 23](#_Toc80273567)

[3.2.1. Download: Node.js Installer 25](#_Toc80273568)

[3.2.2. Installation: Node.js + NPM + Abhängigkeiten 26](#_Toc80273569)

[3.2.3. Installation: TypeScript-Plugin für MS Visual Studio 30](#_Toc80273570)

[3.2.4. Installationspfade: Node.js Installer-Version 31](#_Toc80273571)

[3.2.5. Versionscheck: Als Installationsnachweis 32](#_Toc80273572)

[3.2.6. Updates: Node.js + NPM + TypeScript-Plugin 33](#_Toc80273573)

[3.2.7. Ausführung: Node.js Installer-Version 36](#_Toc80273574)

[3.2.8. Deinstallation: Installer-Version von Node.js 36](#_Toc80273575)

[4. Ergebnis im Einrichtungsprozess der Entwicklungsumgebung für Web-Apps 37](#_Toc80273576)

[4.1. Einordnung der Installationsanleitung 39](#_Toc80273577)

[4.2. Dokumentationen Programmierung von Web-Apps 41](#_Toc80273578)

# Einleitung

Für die Einrichtung einer Entwicklungsumgebung von Web-Apps müssen clientseitige sowie serverseitige Laufzeitumgebungen bereitgestellt werden. Node.js stellt eine clientseitige Laufzeitumgebung für die Funktionalität von Web-Apps bereit.

Diese Installationsanleitung beschreibt die Installation sowie die Einrichtung von:

* Der Node Versions Manager (NVM)
* Node.js
* Dem Node-Package-Manager (NPM)
* Das TypeScript-Plugin für MS Visual Studio

# Lizenzbedingungen

Die Verwendung von Node.js mit allen Funktionserweiterungen sowie von NVM steht im vollen Funktionsumfang frei zur Verfügung. Es handelt sich bei beiden Tools um eine  
MIT-Lizenz vom Massachusetts Institute of Technology, bei der keine weiteren Kosten anfallen.

Die Lizenzbedingungen von NVM können unter folgendem Link nachgelesen werden:

* <https://github.com/nvm-sh/nvm/blob/master/LICENSE.md>

Die Lizenzbedingungen von Node.js können im Installationspfad von Node.js nachgelesen werden. Ausgehend vom Standard-Pfad sind die Lizenzbedingungen wie folgt zu finden:

* Pfad
  + Node.js NVM-Version
    - C:\Users\<user>\AppData\Roaming\nvm\<version>\node\_modules\npm
      * <user> als Platzhalter für das Windows Benutzerprofil
      * <version> als Platzhalter für die Versionsnummer von Node.js
      * <version> Beispiel „v14.17.5“
  + Node.js Installer-Version
    - C:\Program Files\nodejs\node\_modules\npm\
* Datei
  + LICENCE
    - Hat keine Dateiendung
    - Mit einem Texteditor öffnen

Die allgemeine MIT-Lizenz kann im folgenden Link nachgelesen werden:

* <https://mit-license.org/>

# Installation

Die Installation von Node.js kann als eine der folgenden Versionen durchgeführt werden:

|  |  |
| --- | --- |
| **LTS-Version** | **Current-Version** |
| * Stabile Version * Normaler Funktionsumfang * Geringere Fehleranfälligkeit | * Neueste Version * Größerer Funktionsumfang * Höhere Fehleranfälligkeit |

Es gibt zwei Optionen Node.js, den Node Package Manager (NPM) sowie das TypeScript-Plugin zu installieren:

* Über den Node Versions Manager (NVM)
  + NVM-Version
  + Für Windows x86 + x64
* Über den Node.js Installer
  + Installer-Version
  + Für Windows x64

Für die Einrichtung der Entwicklungsumgebung von Web-Apps wird die Installation der NVM-Version durchgeführt. Mit dem NVM lässt sich Node.js als Version:

* LTS
* Current

mittels Konsolenbefehle Installieren und Updaten.

Treten Probleme bei der ersten Option auf, ist optional eine Installationsanleitung für die  
Installer-Version enthalten. Bei der Installer-Version von Node.js ist das Updaten mit einem größeren Aufwand verbunden. Ein Update der Version, welche vom Installer entstammt, muss über:

* Die Deinstallation der alten Version
* Die Installation mit aktualisiertem Installer

manuell vorgenommen werden. Unter Umständen muss die Installation nach Anleitung dieser Option wiederholt werden, da in Ausnahmefällen die Abhängigkeiten nicht ordnungsgemäß installiert werden.

In dieser Installationsanleitung werden die Punkte:

1. Download
   * NVM-Version
   * Installer-Version
2. Installation
   * NVM-Version
     + NVM
     + Node.js + NPM + Abhängigkeiten
     + TypeScript-Plugin
   * Installer-Version
     + Node.js + NPM + Abhängigkeiten
     + TypeScript-Plugin
3. Installationspfade
   * NVM-Version
   * Installer-Version
4. Versionscheck: Als Installationsnachweis
   * NVM-Version
   * Installer-Version
5. Updates: Node.js
   * NVM-Version
   * Installer-Version

beschrieben.

Zusätzlich wird bei Installation der Einheiten von Node.js auf die Punkte

1. Ausführung: Node.js
   * NVM-Version
   * Installer-Version
2. Deinstallation: Node.js
   * NVM-Version
   * Installer-Version

eingegangen.

## Option 1: Node Versions Manager (NVM) für Windows

Bei der Einrichtung einer Entwicklungsumgebung für Web-Apps wird die Installation von Node.js sowie NPM als NVM-Version vorgenommen. Die Installation von Node.js sowie NPM über den NVM teilt sich in drei Schritte auf:

1. Download: NVM-Installer
2. Installation:
   * Node.js Einheiten
     + NVM als Befehlsbibliothek der Windows-Konsolen
     + Node.js + NPM + Abhängigkeiten
     + TypeScript-Plugin
   * Installationspfade
3. Versionscheck: Als Installationsnachweis
4. Updates:
   * NVM
   * Node.js
   * NPM
   * Abhängigkeiten
   * TypeScript-Plugin

Zusätzlich wird auf die Punkte:

1. Ausführung: NVM-Version von Node.js
2. Deinstallation: NVM-Version von Node.js

eingegangen.

Die Einheiten von der Node.js NVM-Version erweitern die aufgelisteten Konsolen um Befehlsbibliotheken:

* Windows Eingabeaufforderung
  + nvm
  + node
  + npm
  + tsc
* Windows PowerShell
  + nvm
  + node
  + npm
  + tsc
* MS Visual Studio Developer-Eingabeaufforderung
  + node
  + tsc
* MS Visual Studio Developer-PowerShell
  + node
  + tsc

Ist bereits eine Installer-Version von Node.js auf dem System installiert, muss diese Version vorher deinstalliert werden. Bei der Löschung von Node.js werden alle dazugehörigen Einheiten ebenfalls entfernt. Die Deinstallation muss vor der Installation von NVM erfolgen.

Eine Dokumentation für die Windows-Version von NVM ist im folgenden Link zu finden:

* <https://github.com/coreybutler/nvm-windows>

### Download: NVM Installer

NVM wird über GitHub wird zum Download bereitgestellt. Für die Installation wird die  
„nwm-setup.zip“ von der aktuellen Version ausgewählt.

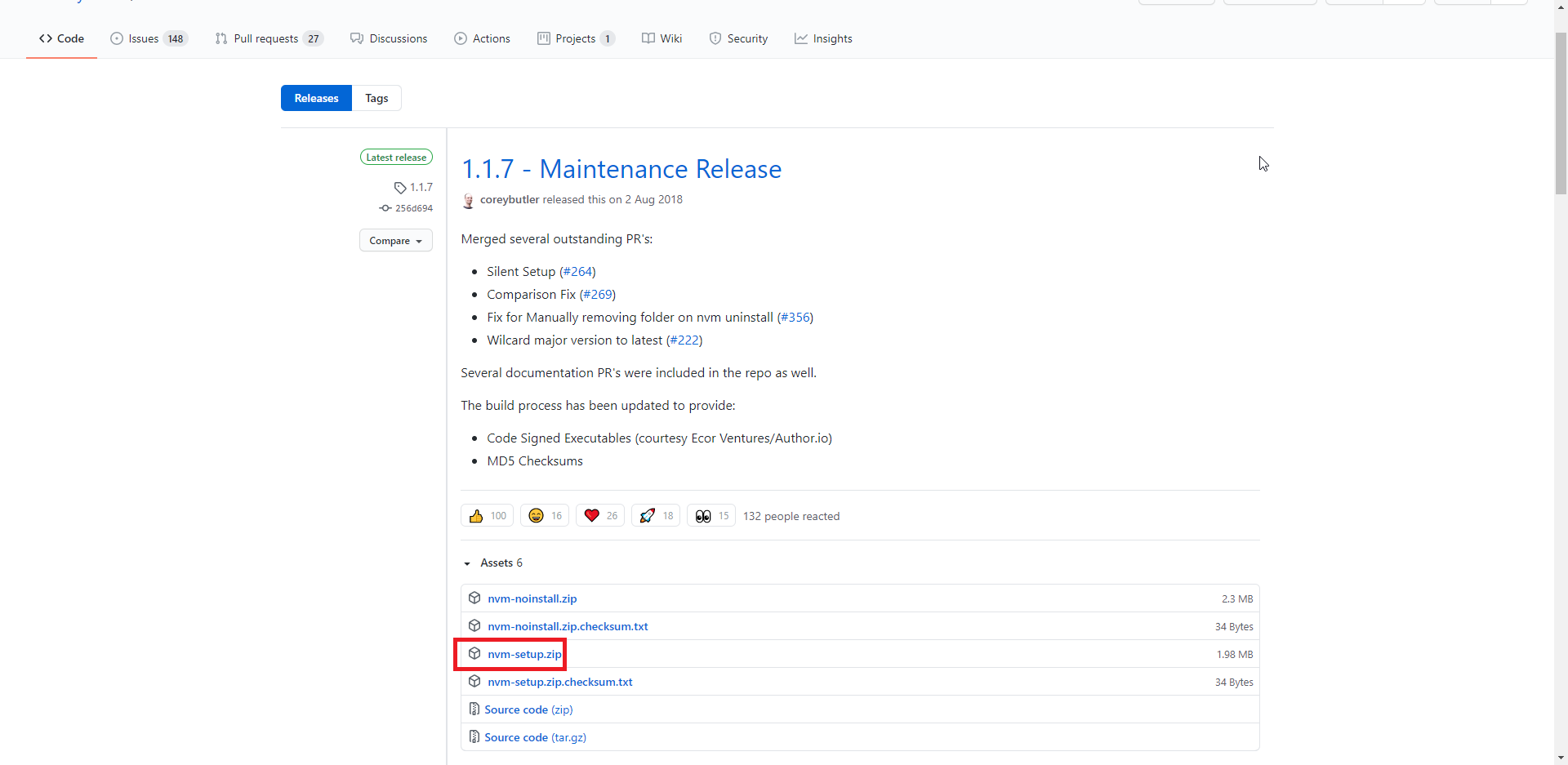


Abbildung 1: Download-Website von Node.js als NVM-Version

Der der Installer von NVM ist unter folgenden Link abrufbar:

* <https://github.com/coreybutler/nvm-windows/releases>

### Installation: NVM als Befehlsbibliothek der Windows-Konsolen

Die Installation von NVM erweitert den Umfang der Windows-Konsolen:

* Windows Eingabeaufforderung
* Windows PowerShell

um eine „nvm“-Befehlsbibliothek.

Zunächst wird der NVM installiert, um mittels „nvm“-Anweisungen in einer Windows-Konsole die nachfolgenden Einheiten zu installieren:

* Node.js
* NPM
* Abhängigkeiten

Die geladene „nwm-setup.zip“ wird entpackt und die enthaltene „nvm-setup.exe“ wird ausgeführt. Die Installation erfolgt im Assistenten wie folgt:

1. Als Administrator starten
2. Licence agreement
   * Akzeptieren
3. Select destination location
   * Standardpfad „C:\Users\**<user>**\AppData\Roaming\nvm“
     + **<user>** als Platzhalter für das Windows Benutzerprofil
4. Set Node.js Symlink
   * Standardpfad „C:\Program Files\nodejs\“
5. Ready to install
6. Install progress
7. Completing the NVM for Windows setup wizard
   * Fertig

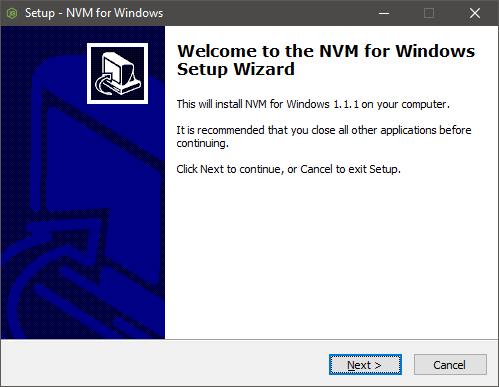


Abbildung 2: NVM-Installations-Assistent

### Installation: Node.js + NPM + Abhängigkeiten

Bei der Installation von Node.js kann aus zwei Versionen ausgewählt werden:

* Die LTS-Version
* Die Current-Version

Um die Fehleranfälligkeit von Web-Apps zu minimieren, wird die LTS-Version ausgewählt.

Nach der Installation von NVM wird in einer der Windows-Konsolen:

* die Windows Eingabeaufforderung
* die Windows PowerShell

als Administrator die „nvm“-Befehlsbibliothek verwendet, um die Einheiten:

* Node.js
* NPM
* Abhängigkeiten

zu installieren. Diese Einheiten erweitern den Umfang der Windows-Konsolen um eine „node“-Befehlsbibliothek sowie um eine „npm“-Befehlsbibliothek. Des Weiteren wird der Umfang der MS Visual Studio Konsolen um eine „node“-Befehlsbibliothek erweitert.

Die Installation wird in vier Schritten durchgeführt:

1. Eingabeaufforderung öffnen
   * Ausführung als Administrator
2. Eingabe Auflistung verfügbarer Versionen
   * nvm list available

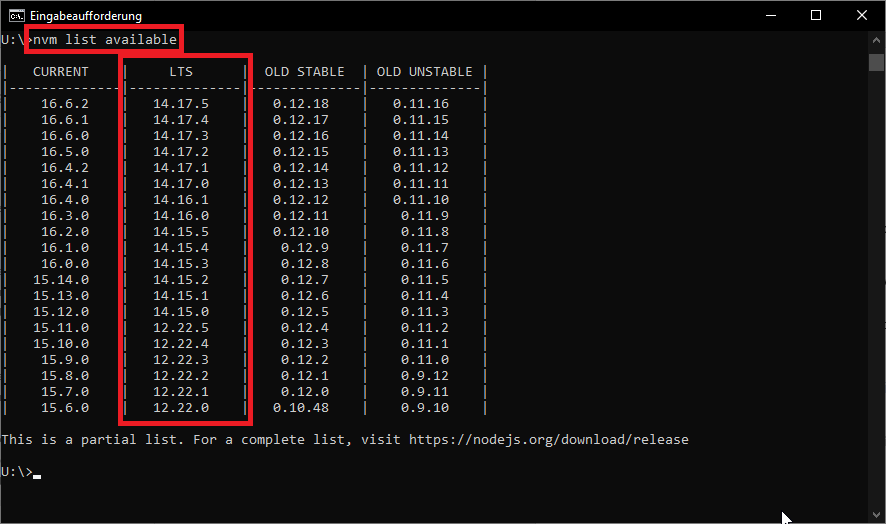


Abbildung 3: NVM - Auflistung von verfügbaren Node.js Versionen

1. Eingabe Installation aktuelle LTS-Version
   * Auswahl oberster Versionseintrag unter „LTS“
   * nvm install **<version>** *[arch]*
     + **<version>** steht für die Eingabe einer Versionsnummer
     + *[arch]* steht für die Eingabe einer Windows Architektur x86 sowie x64
       - Verwendung der Architektur x86 mit Eingabe von „32“
       - Verwendung der Architektur x64 mit Eingabe von „64“
     + Beispielhafte Eingabe für die Auswahl einer Version für x64
       - nvm install **14.17.5** *64*

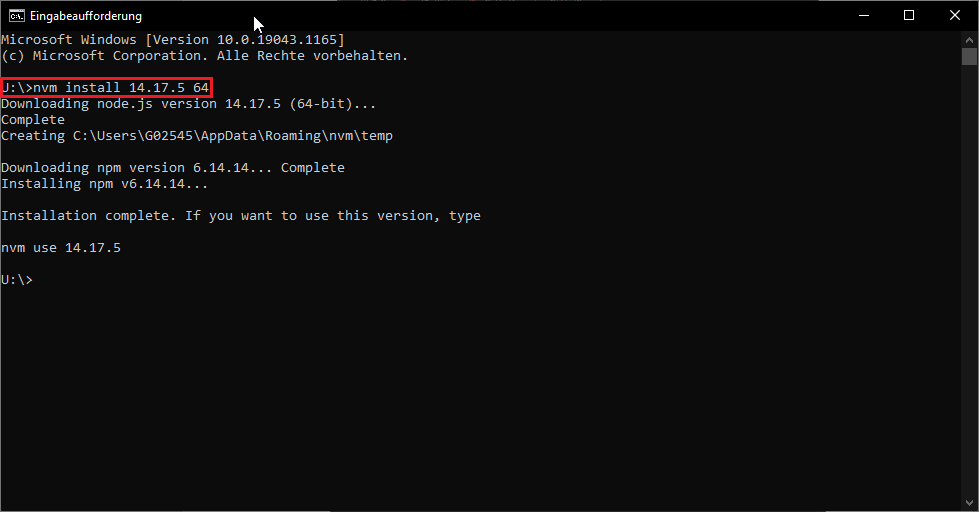


Abbildung 4: NVM - Installation einer stabilen Version von Node.js

1. Eingabe Aktivieren der installierten Version
   * nvm use **<version>** *[arch]*
     + **<version>** steht für die Eingabe einer Versionsnummer
     + *[arch]* steht für die Eingabe einer Windows Architektur x86 sowie x64
       - Verwendung der Architektur x86 mit Eingabe von „32“
       - Verwendung der Architektur x64 mit Eingabe von „64“
     + Ausführung als Administrator zulassen
     + Beispielhafte Eingabe für die Auswahl einer Version für x64
       - nvm use **14.17.5** *64*

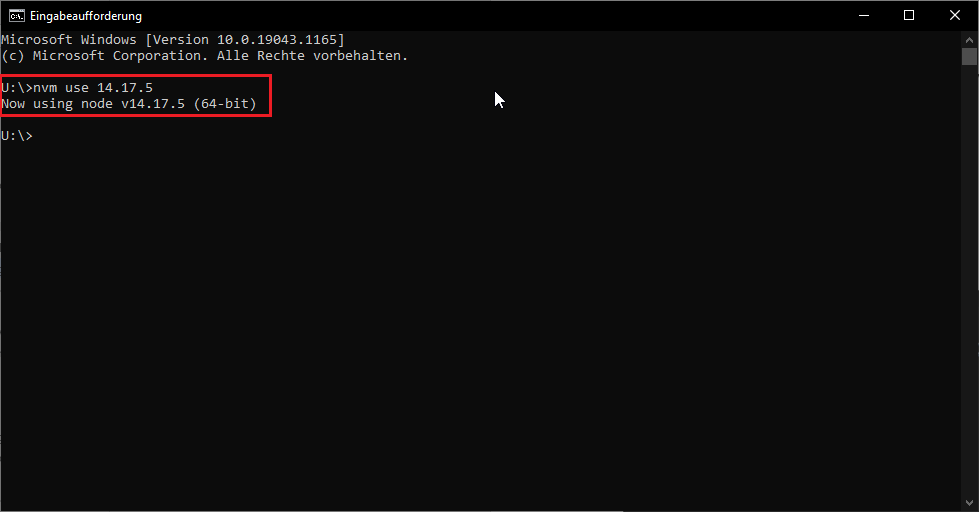


Abbildung 5: NVM - Aktivierung einer stabilen Version von Node.js

Nach der Installation der Einheiten der NVM-Version von Node.js wird in einer der  
Windows-Konsolen die Installation:

* vom TypeScript-Plugin

durchgeführt.

### Installation: TypeScript-Plugin für MS Visual Studio

Das TypeScript-Plugin von Node.js ist ein für MS Visual Studio erstelltes Plugin. Es ermöglicht die Programmierung, die Kompilierung sowie die Ausführung von TypeScript-Code in MS Visual Studio mit Node.js.

Nach der Installation der Einheiten von der Node.js NVM-Version wird in einer der Windows-Konsolen:

* die Windows Eingabeaufforderung
* die Windows PowerShell

als Administrator die „npm“-Befehlsbibliothek verwendet, um:

* das TypeScript-Plugin

zu installieren.

Das Plugin erweitert den Umfang der Windows-Konsolen sowie der MS Visual Studio Konsole um eine „tsc“-Befehlsbibliothek.

Die Installation wird in vier Schritten durchgeführt:

1. Eingabeaufforderung öffnen
   * Als Administrator ausführen
2. Eingabe Installation des TypeScript-Plugins
   * npm install -g typescript
3. Eingabeaufforderung Installationsprozess
4. Eingabeaufforderung Abschluss der Installation

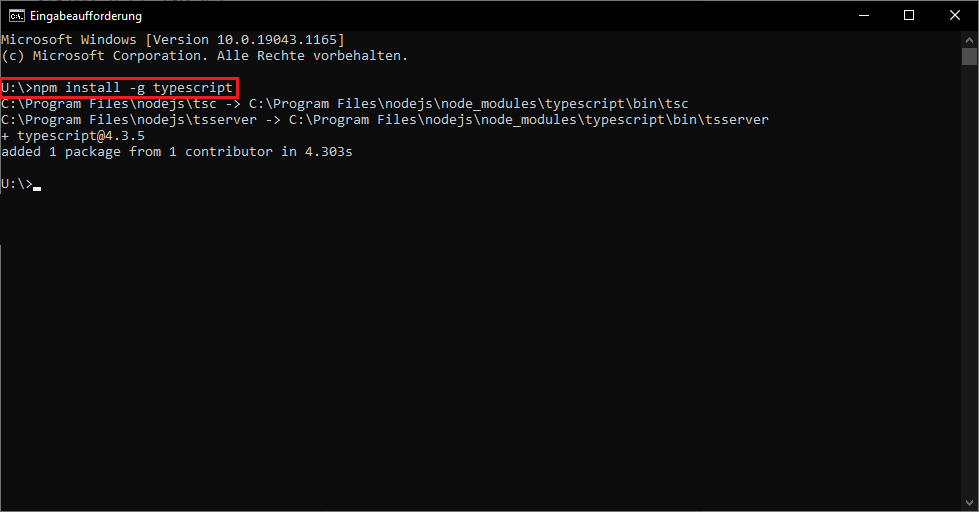


Abbildung 6: Installation: TypeScript-Plugin

### Installationspfade: Node.js NVM-Version

Nach erfolgreichem Abschluss der Installation teilen sich die Einheiten der NVM-Version von Node.js in folgende Pfade auf:

* Installationspfad NVM
  + C:\Users\**<user>**\AppData\Roaming\nvm
    - **<user>** als Platzhalter für das Windows Benutzerprofil
* Installationspfad Node.js
  + C:\Program Files\nodejs
* Executable-Pfad
  + C:\Program Files\nodejs
  + node.exe

### Versionscheck: Als Installationsnachweis

Als Installationsnachweis kann ein Versionscheck für NVM, Node.js sowie NPM als Paket-Manager vorgenommen werden. Dafür werden folgende Eingaben in der Windows Eingabeaufforderung gemacht:

1. Eingabeaufforderung öffnen
2. Eingabe Versionscheck NVM
   * nvm version
3. Eingabe Versionscheck Node.js
   * node -v
4. Eingabe Versionscheck NPM
   * npm -v
5. Eingabe Versionscheck TypeScript-Plugin
   * tsc --version



Abbildung 7: NVM - Versionscheck für NVM + Node.js + NPM + TypeScript-Plugin

### Updates: NVM + Node.js + NPM + Abhängigkeiten + TypeScript-Plugin

Für die Entwicklung von Web-Apps muss die Funktionalität von Node.js aktuell gehalten werden.

Die Einheiten der Node.js NVM-Version:

* NVM
* Node.js
* NPM
* Abhängigkeiten
* TypeScript-Plugin

werden über einem NVM-Installer sowie mit einer Windows-Konsole:

* Windows Eingabeaufforderung
* Windows PowerShell

auf eine aktuelle Versionen geupdatet. Der Update-Prozess unterteilt sich in:

* Updates mit dem NVM-Installer
* Updates mit der Windows-Konsole

Eine aktuelle Version von NVM-Installer setzt ein Update für NVM auf. Dabei wird wie folgt vorgegangen:

1. Download
   * Aktueller NVM-Installer
2. Überinstallieren ohne vorheriges Löschen
   * Der älteren Version von NVM

Die Updates der weiteren Einheiten der Node.js NVM-Version werden, wie in den nächsten Schritten beschrieben, durchgeführt:

1. Eingabeaufforderung öffnen
   * Als Administrator ausführen
2. Eingaben Versionsabgleich von Node.js
   * nvm list
   * nvm list available
3. Wenn neue LTS-Version von Node.js vorhanden
   * Fortfahren mit Schritt Vier
   * Andernfalls Fortfahren mit Schritt Sieben
4. Eingabe Installation aktuelle LTS-Version
   * Auswahl oberster Versionseintrag unter „LTS“
   * nvm install **<version>** *[arch]*
     + **<version>** steht für die Eingabe einer Versionsnummer
     + *[arch]* steht für die Eingabe einer Windows Architektur x86 sowie x64
       - Verwendung der Architektur x86 mit Eingabe von „32“
       - Verwendung der Architektur x64 mit Eingabe von „64“
     + Beispielhafte Eingabe für die Auswahl einer Version für x64
       - nvm use **14.17.5** *64*

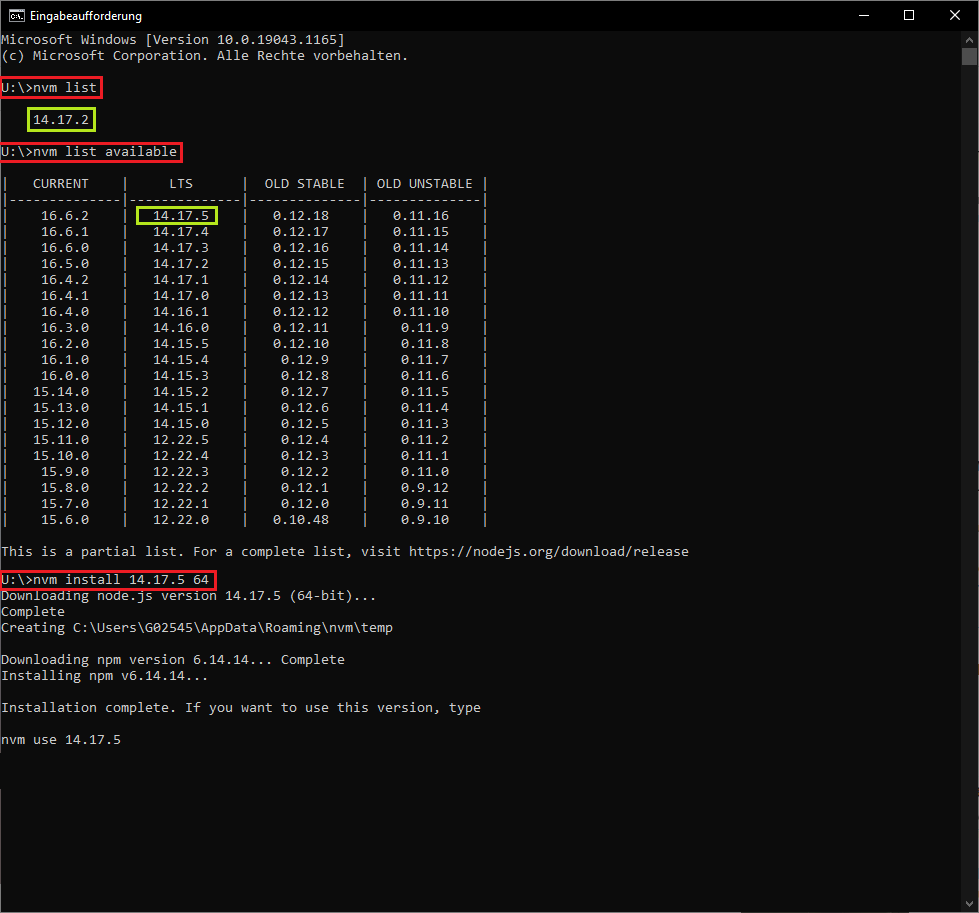


Abbildung 8: NVM - Node.js Versionsvergleich

1. Eingabe Aktivieren der installierten Version
   * nvm use **<version>** *[arch]*
     + **<version>** steht für die Eingabe einer Versionsnummer
     + *[arch]* steht für die Eingabe einer Windows Architektur x86 sowie x64
       - Verwendung der Architektur x86 mit Eingabe von „32“
       - Verwendung der Architektur x64 mit Eingabe von „64“
     + Ausführung als Administrator zulassen
     + Beispielhafte Eingabe für die Auswahl einer Version für x64
       - nvm use **14.17.5** *64*

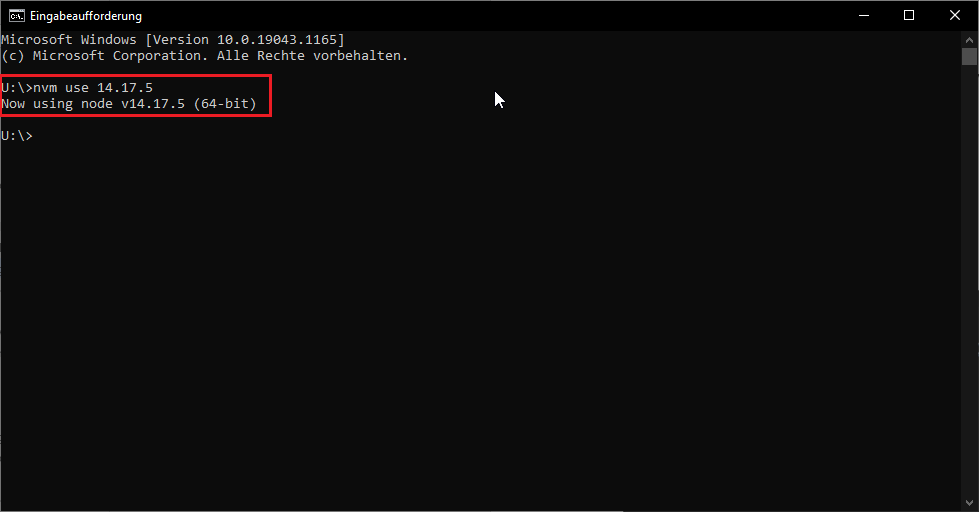


Abbildung 9: NVM - Aktivierung einer stabilen Version von Node.js

1. Eingabe Deinstallation alte LTS-Version
   * nvm uninstall **<version>**
     + **<version>** steht für die Eingabe einer Versionsnummer
     + Beispielhafte Eingabe für die Auswahl einer Version für x64
       - nvm uninstall **14.17.2**

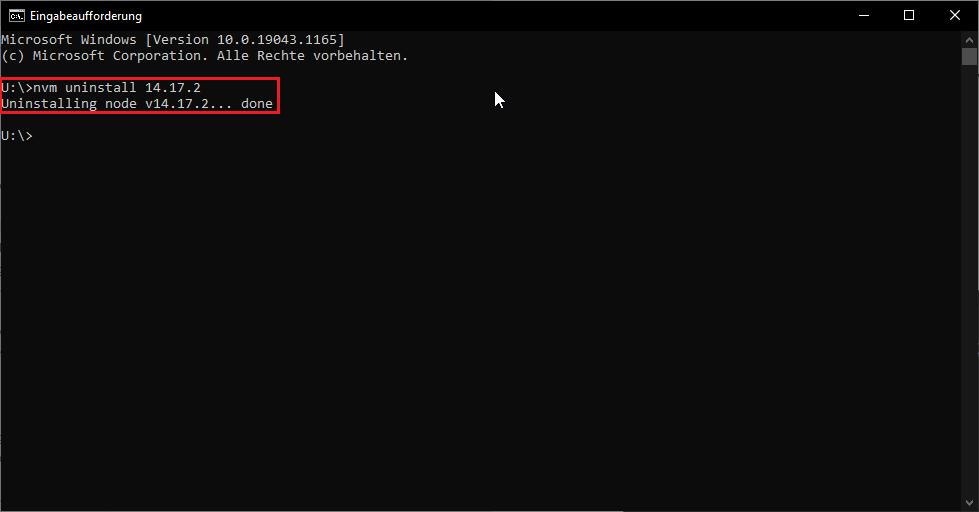


Abbildung 10: NVM - Deinstallation einer Version von Node.js

1. Eingabe Update TypeScript-Plugin
   * npm update

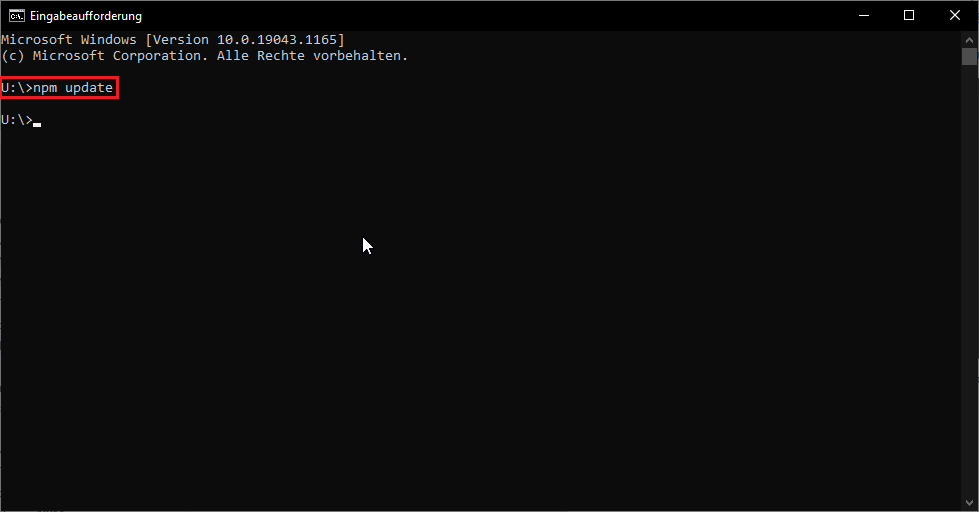


Abbildung 11: Update: TypeScript-Plugin mittels eines NPM-Updates

1. Eingaben für Versionschecks
   * nvm version
   * node -v
   * npm -v
   * tsc --version



Abbildung 12: NVM - Versionscheck für NVM + Node.js + NPM + TypeScript-Plugin

### Ausführung: NVM-Version von Node.js

Die Aktivierung einer Installierten Version von Node.js erfolgt über eine Windows-Konsole:

* Windows Eingabeaufforderung
* Windows PowerShell

mit der Eingabe von:

* nvm use **<version>** *[arch]*
  + **<version>** steht für die Eingabe einer Versionsnummer
  + *[arch]* steht für die Eingabe einer Windows Architektur x86 sowie x64
    - Verwendung der Architektur x86 mit Eingabe von „32“
    - Verwendung der Architektur x64 mit Eingabe von „64“
  + Ausführung als Administrator zulassen
  + Beispielhafte Eingabe für die Auswahl einer Version für x64
    - nvm use **14.17.5** *64*

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 13: NVM - Aktivierung einer stabilen Version von Node.js

Node.js kann über den folgenden Pfad ausgeführt werden:

* Pfad
  + C:\Users\**<user>**\AppData\Roaming\nvm\v14.17.5
    - **<user>** als Platzhalter für das Windows Benutzerprofil
* Executable
  + node64.exe

### Deinstallation: NVM-Version von Node.js

Die Deinstallation der NVM-Version von Node.js erfolgt in zwei Schritten:

1. Deinstallation Node.js über NVM deinstallieren
   * Eingabeaufforderung als Administrator
     + Als Administrator ausführen
   * Auflistung installierter Versionen
     + nvm list
   * Deinstallation alte LTS-Version von Node.js + NPM
     + nvm uninstall <version>
     + <version> steht für die Eingabe der LTS-Version aus Eingabe „nvm list“
2. Deinstallation von NVM über „Programme hinzufügen oder entfernen“
   * Bezeichnung Eintrag in der Liste
     + „NVM for Windows **<version>**“
       - **<version>** steht für die installierte Version

## Option 2: Node.js Installer

Sollte die Installation der Einheiten von Node.js anhand der ersten Option fehlschlagen, kann die alternativ die Installation mit dem Node.js Installer durchgeführt werden. Die Installation über den Installer teilt sich in drei Schritte auf:

1. Download: Node.js-Installer
2. Installation:
   * Node.js Einheiten
     + Node.js
     + NPM
     + Abhängigkeiten
   * Installationsprozess
     + Node.js Setup
     + Fortgeführte Installation in der Windows Eingabeaufforderung
     + Fortgeführte Installation in der Windows PowerShell
   * Installationspfade
3. Versionscheck: Als Installationsnachweis
4. Updates:
   * Node.js
   * NPM
   * Abhängigkeiten
   * TypeScript-Plugin

Zusätzlich wird auf die Punkte:

1. Ausführung: NVM-Version von Node.js
2. Deinstallation: NVM-Version von Node.js

eingegangen.

Die Einheiten von der Node.js Installer-Version erweitern die aufgelisteten Konsolen um Befehlsbibliotheken:

* Windows Eingabeaufforderung
  + node
  + npm
  + tsc
* Windows PowerShell
  + node
  + npm
  + tsc
* MS Visual Studio Developer-Eingabeaufforderung
  + node
  + tsc
* MS Visual Studio Developer-PowerShell
  + node
  + tsc

Ist bereits eine NVM-Version von Node.js auf dem System installiert, muss diese Version sowie NWM vorher deinstalliert werden. Bei der Löschung von Node.js werden alle dazugehörigen Einheiten ebenfalls entfernt. Die Deinstallation muss vor der Installation von der Installer-Version erfolgen.

Unter Umständen muss die Installation nach Anleitung dieser Option wiederholt werden, da in Ausnahmefällen die Abhängigkeiten:

* über die Windows Eingabeaufforderung
* über die Windows PowerShell

nicht ordnungsgemäß installiert werden. Vorher muss die Installer-Version von Node.js deinstalliert werden.

### Download: Node.js Installer

Beim Download vom Node.js Installer können zwei Versionen ausgewählt werden:

* Die LTS-Version
* Die Current-Version

Um die Fehleranfälligkeit von Web-Apps zu reduzieren, wird die LTS-Version zum Download ausgewählt. Im nächsten Schritt wird Windows-Installer als „.msi“-Variante in der 64-Bit Version gedownloadet.

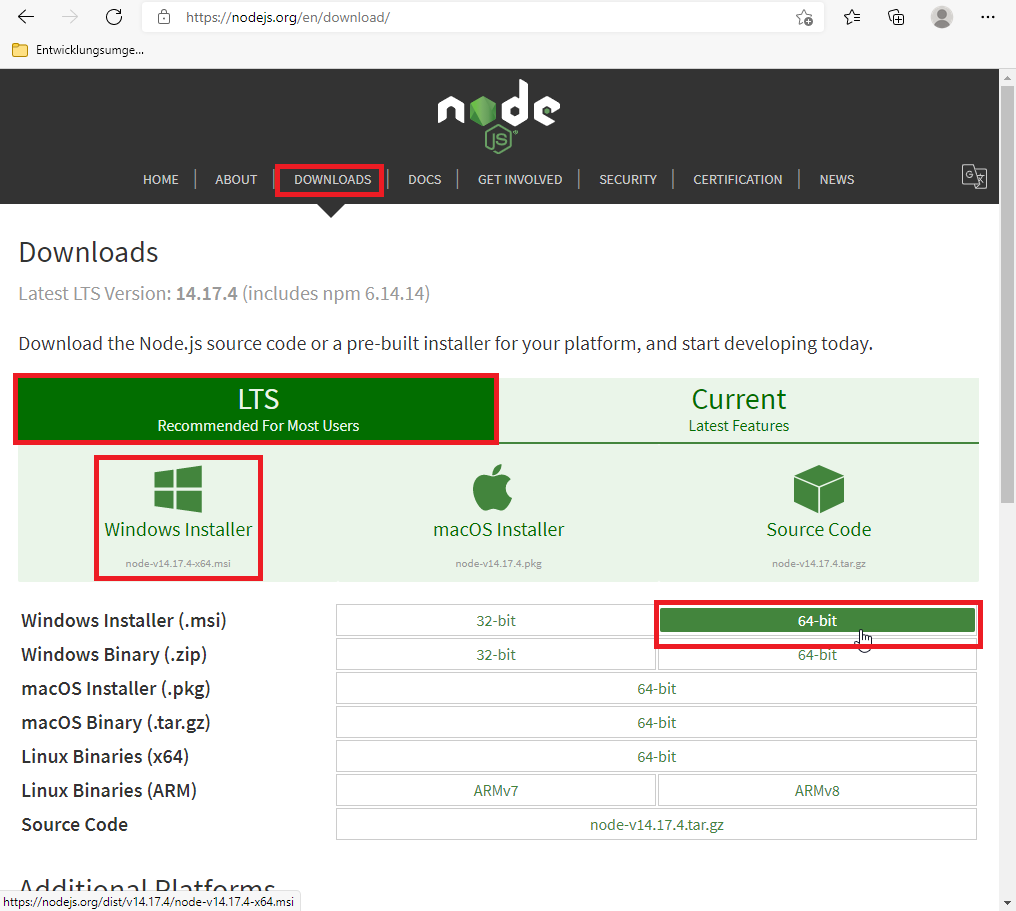


Abbildung 14: Download-Website von Node.js

Der Installer von Node.js ist unter folgendem Link abrufbar:

* <https://nodejs.org/en/download/>

### Installation: Node.js + NPM + Abhängigkeiten

Die Installation von Node.js unterteilt sich in drei Teilschritte, welche nacheinander zusammenhängend ausgeführt werden. Dabei sollte ab dem zweiten Teilschritt der Installation kein Installationsabbruch vorgenommen werden. Die Installation läuft wie folgt ab:

1. Node.js Setup im Installations-Assistenten
2. Fortgeführte Installation in der Windows Eingabeaufforderung
3. Fortgeführte Installation in der Windows PowerShell

Bei der Installation werden die Einheiten von Node.js installiert:

* Node.js
* NPM
* Abhängigkeiten

Diese Einheiten erweitern den Umfang der Windows-Konsolen:

* Windows Eingabeaufforderung
* Windows PowerShell

um eine „node“-Befehlsbibliothek sowie um eine „npm“-Befehlsbibliothek. Des Weiteren wird der Umfang der MS Visual Studio Konsolen:

* Developer-Eingabeaufforderung
* Developer-PowerShell

um eine „node“-Befehlsbibliothek erweitert.

**Node.js Setup**

Der erste Teil der Installation der Kernkomponente erfolgt über einen Installations-Assistenten. Die folgenden Schritte werden dabei durchlaufen:

1. Welcome
2. End-User License Agreement
   * Akzeptieren
3. Destination Folder
   * Standardpfad „C:\Program Files\nodejs\“
4. Custom Setup
   * Alle Features installieren

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 15: Node.js Installations-Assistent: Custom Setup

1. Tools for Native Modules
   * Feld zum automatischen Installieren aller benötigten Tools markieren

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 16: Node.js Installations-Assistent:Tools for Native Modules

1. Ready to install Node.js
2. Abfrage Administationsrechte
   * Zulassen
3. Completed the Node.js Setup Wizard
   * Fertig

Nach Abschluss der Installation im Installations-Assistenten wird die Installation in der Windows Eingabeaufforderung automatisch fortgeführt.

**Fortgeführte Installation in der Windows Eingabeaufforderung**

Der zweite Teil der Installation der Kernkomponente erfolgt über der Windows Eingabeaufforderung. Die folgenden Schritte werden dabei durchlaufen:

1. Automatisches Öffnen der Eingabeaufforderung
2. Eingabeaufforderung „Tools for Node.js Native Modules Installation Script“
   * Beliebige Taste zum akzeptieren
3. Eingabeaufforderung „Third Party Tools“
   * Beliebige Taste zum akzeptieren
4. Eingabeaufforderung Abschluss der Installation

Nach Abschluss der Installation der Windows Eingabeaufforderung wird die Installation in der Windows PowerShell automatisch fortgeführt.

**Fortgeführte Installation in der Windows PowerShell**

Der dritte Teil der Installation der Kernkomponente erfolgt über der Windows PowerShell. Die folgenden Schritte werden dabei durchlaufen:

1. Ausführung von PowerShell als Administrator zulassen
   * Ja
2. PowerShell Installieren der Pakete
   * Download und Installation der Pakete
   * Installation von einigen MS Visual Studio 2017 Tools kann fehlschlagen
     + Da Version 2019+ genutzt
   * Nehmen einige Zeit in Anspruch
3. PowerShell Abschluss der Installation
   * Eingabe „ENTER“

Mit erfolgreichem Durchlauf des dritten Teilschrittes ist die Installation der Kernkomponente des Node.js abgeschlossen.

### Installation: TypeScript-Plugin für MS Visual Studio

Das TypeScript-Plugin von Node.js ist ein für MS Visual Studio erstelltes Plugin. Es ermöglicht die Programmierung, die Kompilierung sowie die Ausführung von TypeScript-Code in MS Visual Studio mit Node.js.

Nach der Installation der Einheiten von der Node.js NVM-Version wird in einer der Windows-Konsolen:

* die Windows Eingabeaufforderung
* die Windows PowerShell

als Administrator die „npm“-Befehlsbibliothek verwendet, um:

* das TypeScript-Plugin

zu installieren.

Das Plugin erweitert den Umfang der Windows-Konsolen sowie der MS Visual Studio Konsole um eine „tsc“-Befehlsbibliothek.

Die Installation wird in vier Schritten durchgeführt:

1. Eingabeaufforderung öffnen
   * Als Administrator ausführen
2. Eingabe Installation des TypeScript-Plugins
   * npm install -g typescript
3. Eingabeaufforderung Installationsprozess
4. Eingabeaufforderung Abschluss der Installation

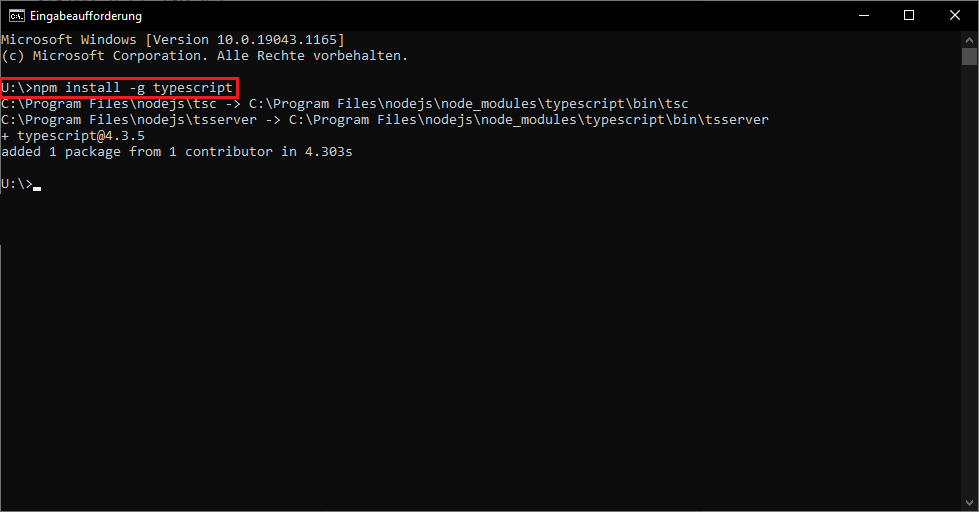


Abbildung 17: Installation: TypeScript-Plugin

### Installationspfade: Node.js Installer-Version

Nach erfolgreichem Abschluss der Installation teilen sich die Einheiten der NVM-Version von Node.js in folgende Pfade auf:

* Installationspfad Node.js
  + C:\Program Files\nodejs
* Executable-Pfad
  + C:\Program Files\nodejs
  + node.exe

### Versionscheck: Als Installationsnachweis

Als Installationsnachweis kann ein Versionscheck für Node.js sowie NPM vorgenommen werden. Dafür gibt es folgende Eingaben in der Windows Eingabeaufforderung:

1. Eingabeaufforderung öffnen
2. Eingabe Versionscheck Node.js
   * node -v
3. Eingabe Versionscheck NPM
   * npm -v
4. Eingabe Versionscheck TypeScript-Plugin
   * tsc --version



Abbildung 18: Installer - Versionscheck für Node.js + NPM + TypeScript-Plugin

### Updates: Node.js + NPM + TypeScript-Plugin

Für die Entwicklung von Web-Apps muss die Funktionalität von Node.js aktuell gehalten werden. Für die Variante des Updates von Node.js als Installer-Version wird eine erneute Installation mit einer aktuellen stabilen Version durchgeführt.

Die Einheiten der Node.js Installer-Version:

* Node.js
* NPM
* Abhängigkeiten
* TypeScript-Plugin

werden auf eine aktuelle Versionen geupdatet.

Die Updates der Einheiten der Node.js Installer-Version werden, wie in den nächsten Schritten beschrieben, durchgeführt:

1. Download: Node.js Installers
   * Nach Anleitung von Punkt
     + 3.2.1. Download: Node.js-Installer
   * Aktuelle stabile LTS-Version als 64-Bit Architektur (x64)

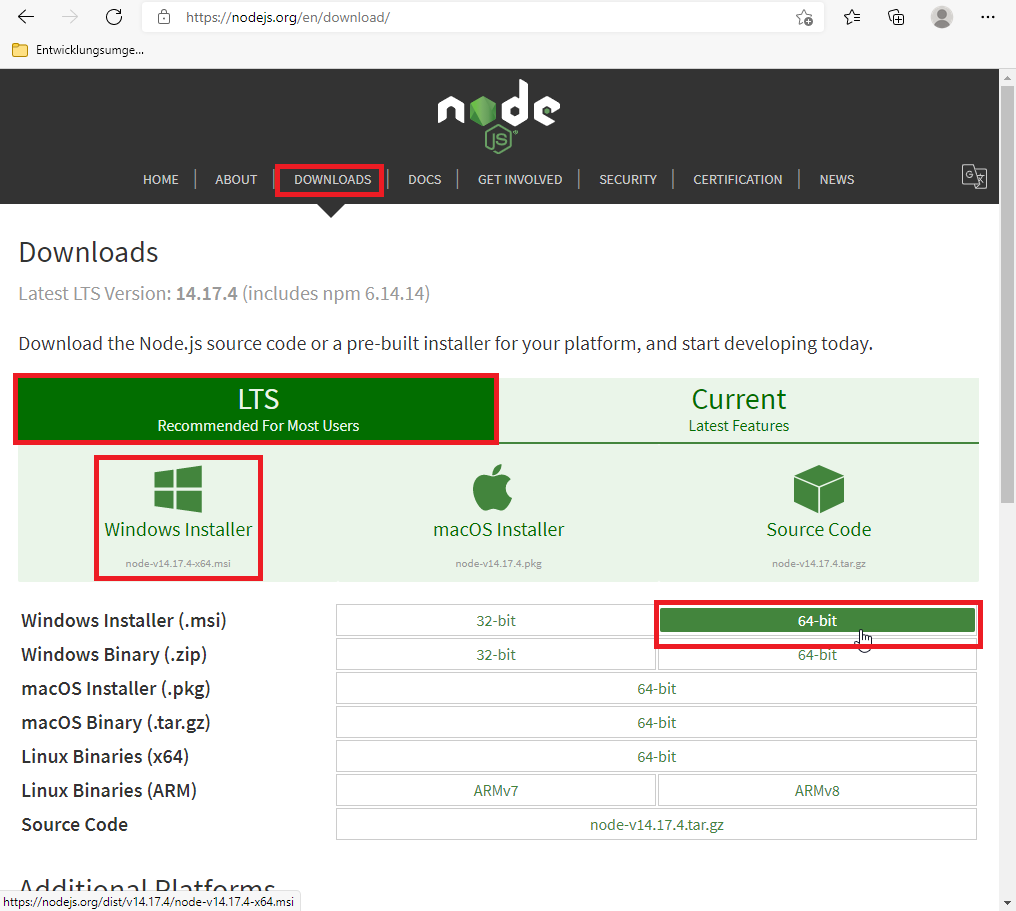


Abbildung 19: Download-Website von Node.js

1. Installation: Node.js + NPM + TypeScript-Plugin
   * Nach Anleitung von Punkt
     + 3.2.2. Installation: Node.js + NPM + TypeScript-Plugin
   * Installations-Ablauf
     + Node.js Setup
     + Fortgeführte Installation in der Windows-Eingabeaufforderung
     + Fortgeführte Installation in der Windows-PowerShell
2. Eingabe Update TypeScript-Plugin
   * Nach Anleitung von Punkt
     + 3.2.3. Installation: TypeScript-Plugin für MS Visual Studio
   * Eingabeaufforderung öffnen
     + Als Administrator ausführen
   * Eingabe Installation des TypeScript-Plugins
     + npm install -g typescript
   * Eingabeaufforderung Installationsprozess
   * Eingabeaufforderung Abschluss der Installation

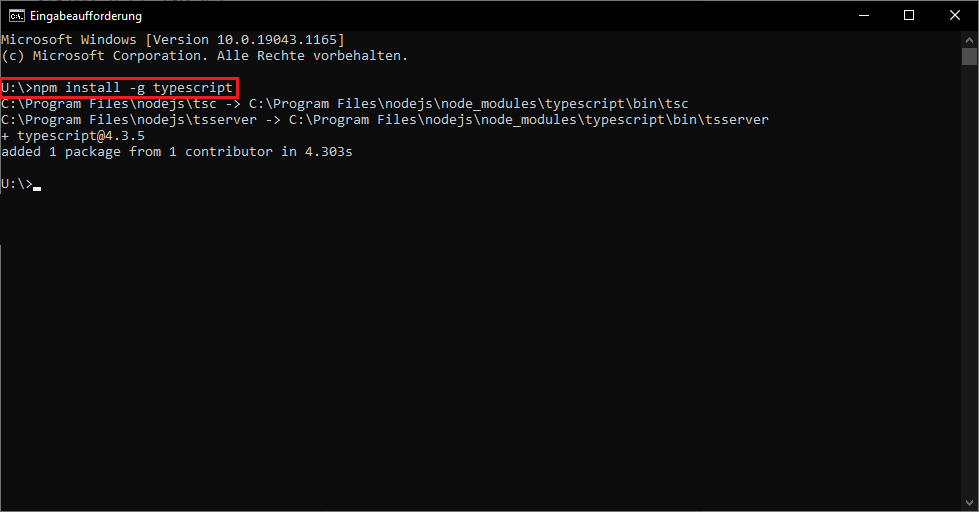


Abbildung 20: Installation: TypeScript-Plugin

1. Eingaben für Versionschecks
   * node -v
   * npm -v
   * tsc --version



Abbildung 12: NVM - Versionscheck für Node.js + NPM + TypeScript-Plugin

### Ausführung: Node.js Installer-Version

Node.js kann über den folgenden Pfad ausgeführt werden:

* Pfad
  + C:\Program Files\nodejs
* Executable
  + „node64.exe“

### Deinstallation: Installer-Version von Node.js

Die Installer-Version von Node.js kann über „Programme hinzufügen oder entfernen“ deinstalliert werden:

* Bezeichnung Eintrag in der Liste
  + „Node.js“

# Ergebnis im Einrichtungsprozess der Entwicklungsumgebung für Web-Apps

Die Installation von Node.js mit NPM als Paket-Manager sowie das TypeScript-Plugin für MS Visual Studio ist ein Teilprozess bei der Einrichtung einer Entwicklungsumgebung für Web-Apps. Folglich wird die Installation im gesamten Prozess eingeordnet.

Der Prozess zur Einrichtung einer Entwicklungsumgebung für Web-Apps sieht wie folgt aus:

1. *Installation: MS Visual Studio + Plugins*
   1. *Installation: MS Visual Studio – Version 2019 Community Edition*

* Wird abgelöst durch Version 2022 Professional Edition
  + Nach Release
  1. *Installation: MS Visual Studio – Version 2022 Professional Edition*
* Ablösung Version 2019 Community Edition
  + Erneute Anwendung dieser Installationsanleitung
  1. *Installation: MS Visual Studio – Version 2022+ Professional Edition*
* Ablösung der älteren Version
  + Erneute Anwendung dieser Installationsanleitung

1. **Installation: Node.js + NPM + TypeScript**

* Laufzeitumgebung für den Client
  + Mit Node Paket Manager
  + Mit TypeScript Plugin

1. Installation: MS Internet Information Service

* Laufzeitumgebung für den Server

1. Installation: Postman

* Wep-API Testframework

1. Installation: Git

* Als Versionskontrolle

1. Updates/Versionsupgrade

* Versionsupgrade: MS Visual Studio
  + Nach Release einer neuen Version
* Updates: MS Visual Studio + Plugins sowie Tools/Frameworks
  + Regelmäßig auf Updates überprüfen
* Updates: Node.js mit Funktionserweiterungen
  + Regelmäßig auf Updates überprüfen
* Updates: MS Internet Information Service
  + Regelmäßig auf Updates überprüfen
* Updates: Postman
  + Regelmäßig auf Updates überprüfen
* Updates: Git
  + Regelmäßig auf Updates überprüfen

Die Arten der Schriftdarstellung geben folgende Informationen

* Normale Schrift
  + Noch bevorstehender Teilprozess im Installationsprozess
* Fett gedruckte Schrift + Blaue Färbung
  + Aktuell anstehender Teilprozess im Installationsprozess
* Kursive Schrift
  + Abgeschlossener Teilprozess im Installationsprozess

Mit Fertigstellung des Installationsprozesses dienen Dokumentationen als Vorbereitung für die Programmierung von Web-Apps. Zuletzt ist für die Einrichtung einer Entwicklungsumgebung für Web-Apps die Beschaffung weiterer Hardware notwendig, welche im Konzept beschrieben wird.

## Einordnung der Installationsanleitung

Die Einrichtung einer Entwicklungsumgebung für Web-Apps benötigt die Bereitstellung eines Servers. Nach der Installation von MS Visual Studio ist die Installation von Node.js mit NPM als Paket-Manager sowie das TypeScript-Plugin für MS Visual Studio vorgesehen.

Die Installation von Node.js mit den Funktionserweiterungen umfasst folgende Tools sowie Frameworks aus dem Konzept für die Entwicklungsumgebung von Web-Apps:

* Node.js
  + Laufzeitumgebung
    - Clientseitig
  + Node-Pakage-Manager (NPM)
    - Paket-Manager
  + TypeScript
    - Als MS Visual Studio Plugin

Für die Einrichtung der Entwicklungsumgebung für Web-Apps ist die Installation weiterer Tools Frameworks notwendig:

1. Installation: MS Internet Information Service

* Laufzeitumgebung für den Server

1. Installation: Postman

* Wep-API Testframework

1. Installation: Git

* Als Versionskontrolle

1. Updates/Versionsupgrade

* Versionsupgrade MS Visual Studio
  + Nach Release einer neuen Version
* Updates MS Visual Studio + Plugins sowie Tools/Frameworks
  + Regelmäßig auf Updates überprüfen
* Updates: Node.js mit Funktionserweiterungen
  + Regelmäßig auf Updates überprüfen
* Updates: MS Internet Information Service
  + Regelmäßig auf Updates überprüfen
* Updates: Postman
  + Regelmäßig auf Updates überprüfen
* Updates: Git
  + Regelmäßig auf Updates überprüfen

## Dokumentationen Programmierung von Web-Apps

Mit Fertigstellung der Installation der Tools/Frameworks einer prototypischen lokalen Entwicklungsumgebung erfolgte die Dokumentation für wichtige Programmierelemente für Web-Apps. Diese Dokumentationen dienen als Hilfestellung um die Programmierung von Web-Apps vorzubereiten.

Für folgende Elemente wird es Dokumentationen geben:

* Tool - MS Visual Studio
  + Einrichtung eines Projektes
  + Views
* Tool - Postman Web-API Testframework
* Tool - Git Versionskontrolle
* Programmierung - ASP.NET + NET.Core 5.0 + C#
  + Mit „Hello World“
  + Wichtige Programmierelemente
    - Kommunikation mit Datenbank des Servers
    - Zugriff auf eine Domäne
  + Boilerplates
  + Code Convention + Style Guide
* Programmierung - JavaScript + TypeScript
  + Mit „Hello World“
  + Wichtige Programmierelemente
  + Boilerplates
  + Code Convention + Style Guide
* Programmierung - React + React Native
  + Mit „Hello World“
  + Wichtige Programmierelemente
  + Boilerplates
  + Code Convention + Style Guide
* Programmierung - HTML + CSS
  + Mit „Hello World“
  + Wichtige Programmierelemente
  + Boilerplates
  + Code Convention + Style Guide