

Kurs-Setup Angular Schulung

Betriebssystem: Windows 10

Alternativ kann ich die Schulung problemlos auch mit einem Linux-System durchführen. Falls der Kunde lieber Mac OSX einsetzen möchte, ist das in Ordnung. Ich selbst verfüge allerdings über keinen Mac.

Software

- Node.js: [Node.js](#) (LTS Version)
- Visual Studio Code: [Visual Studio Code - Code Editing. Redefined](#) (Standard-Download, User Setup)
- Git: [Git](#) (Standard Download)
- Google Chrome Browser (alternativ Mozilla Firefox oder Edge Chromium)
- Terminal/Console

Detaillierte Installation

Nach dem Starten des jeweiligen Installers habe ich für jedes einzelne Fenster des Installations-Dialoges (falls Einstellungsmöglichkeiten vorhanden sind) kurz beschrieben, welche Haken gesetzt oder Auswahl-Möglichkeiten getroffen werden sollten. Natürlich könnten in neueren Versionen, die ich hier noch nicht berücksichtigt habe, weitere Einstellungsmöglichkeiten vorhanden sein. In diesem Fall empfehle ich jeweils die Standard-Auswahl des Installers zu übernehmen.

Falls die Schulung unter einem anderen Betriebssystem als MS Windows durchgeführt werden soll, sind die Anweisungen unter den folgenden Punkten jeweils obsolet. Diese würde ich dann dem Schulungsteilnehmer bzw. Kunden selbst überlassen - oder gerne in den ersten Minuten der Schulung mit den Teilnehmern gemeinsam durchführen.

Node.js

1. Installer starten
2. Lizenz annehmen
3. Installationspfad übernehmen

4. Anpassung der Installation: alles auswählen, wichtig ist das Hinzufügen zur PATH Variable
5. Native Tool-Unterstützung: bitte nicht(!) einschalten

Visual Studio Code

1. Installer starten
2. Lizenz annehmen
3. Installationspfad übernehmen
4. Startmenü erstellen übernehmen
5. Zusätzliche Aufgaben: zu PATH Variable hinzufügen ist wichtig. Alternativ zusätzlich dem Ordner-Kontextmenü hinzufügen

Git

1. Installer starten
2. Lizenz annehmen
3. Komponenten-Auswahl: Standard übernehmen
4. Editor-Auswahl: Standard übernehmen
5. PATH-Variable: mittlere empfohlene (recommended) Option wählen (Git von der Kommandozeile)
6. HTTPS-Backend: OpenSSL
7. Zeilenendungen (Line-Endings): erste Option wählen (Checkout Windows-Style, commit Linux-Style)
8. Terminal-Emulation: MinTTY
9. Extra-Optionen: Standard übernehmen: die ersten zwei Haken gesetzt (symbolische Links nicht)
10. Weitere experimentelle Features neuerer Git-Versionen einfach ausgeschaltet lassen

Browser

Installieren Sie bitte im einfachsten Fall den Google Chrome - oder alternativ den Mozilla Firefox. Der Edge Chromium ist bei einem aktuellen Windows 10 möglicherweise bereits installiert. Natürlich werde ich - falls vom Kunden gewünscht - auf etwaige Besonderheiten bei der Berücksichtigung des IE 11 (Internet Explorer), des MS Edge (alte Legacy-Version) bzw. anderer Not-Ever-Green Browser eingehen.

Terminal/Console

Während der Schulung setze ich wiederholt Terminals ein, um Kommandozeilen-Befehle abzusetzen. Es werden aber nur rudimentäre eingebaute Befehle benutzt - hauptsächlich `cd` zum Verzeichniswechsel. Also darf jeder beliebige Terminal eingesetzt werden:

- Windows `cmd`
- Powershell
- Git Bash
- ...

Beim Einsatz der Powershell ist zu beachten, dass die Ausführungsrechte für Scripte konfiguriert sind:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Proxy

Falls der Rechner auf dem die Schulungsumgebung installiert wird, hinter einem Proxy sitzt, müssen noch npm und git für diesen konfiguriert werden:

```
npm config set proxy http://username:password@host:port
npm config set https-proxy http://username:password@host:port

git config --global http.proxy http://username:password@host:port
```

Protokoll (`http` oder `https`) muss natürlich angepasst werden. Genauso `username` und `password` , diese beiden könnten natürlich auch entfallen.