

Veranstaltungseinheit Node.js

Erste Erfahrungen mit Node

Ziel der Übung

Sie haben Node.js auf ihrem Rechner installiert und können ein einfaches Programm starten und testen.

Aufgaben

Installation von node.js unter Unix(Linux/OSX) (Für Windows, siehe weiter unten.)

Installieren Sie den Node Version Manager (NVM).

Mit dem Node Version Manager können Sie auf Ihrem System viele Versionen Node verwalten und pro Verzeichnis und Projekt eine andere Version nutzen. Installieren Sie Node wenn möglich mit NVM. Falls Ihnen das nicht gelingen sollte, dann installieren Sie Node wie auf [node.js.org](https://nodejs.org) beschrieben.

Die Installation von NVM ist abhängig von Ihrem Betriebssystem. Lesen Sie die entsprechende Dokumentation (siehe Ressourcen unten). Anbei ist eine Möglichkeit für OSX, welche davon ausgeht, dass Sie bereits Homebrew zum Installieren von Paketen auf dem Mac nutzen.

1. `brew install nvm`
2. `mkdir ~/.nvm`
3. `cp $(brew --prefix nvm)/nvm-exec ~/.nvm/`
4. Fügen Sie in die `.bashrc` / `.zshrc` hinzu: `export NVM_DIR=~/.nvm`
`source $(brew --prefix nvm)/nvm.sh`
5. Starten Sie Ihre Shell neu
6. `nvm install v4.0.0`

Sie können nun die Version Node zu jedem beliebigen Zeitpunkt wählen.

Mit `nvm list` sehen Sie eine Liste der installierten Versionen. Aktuell sollte v4.0.0 installiert sein. Diese Version können Sie wählen mit `nvm use 4.0.0`.

Installation von node.js unter Windows

Installieren Sie den Node Installer von nodejs.org.

Weiter für alle Plattformen

Prüfen Sie, dass node funktioniert:

```
$ node --version  
v4.0.0
```

Einfaches Node Programm

Im Ordner **file-system** finden Sie ein einfaches Programm, dass prüft, ob eine Datei verändert wurde.

Lassen Sie dieses Programm wie folgt laufen:

```
$ node watcher.js
```

In einem zweiten Terminalfenster wechseln Sie in das Verzeichnis und führen folgenden Befehl aus:

```
$ touch target.txt
```

Node Module installieren

Der Node Package Manager **npm** dient dazu, Node Module und abhängige Pakete zu installieren.

Eines der Module, die wir brauchen ist **jasmine-node**, die Node Version des Jasmine Testtools, dass Sie bereits kennen.

Mit npm wird es wie folgt installiert:

```
$ sudo npm install jasmine-node -g
```

npm Module werden entweder **lokal** für ein bestimmtes Projekt, oder **global** auf ihrem Computer installiert. Das **-g** weist npm an, ein Modul global zu installieren.

Neben dem eigentlich Modul werden auch sämtliche Abhängigkeiten automatisch installiert.

Einfacher Jasmine Test

Im Directory `labor/tdd-jasmine-node` finden Sie ein Verzeichnis `spec` und `lib` mit dem Anfang eines Berechnungmoduls. Lassen Sie jasmine laufen:

```
$ jasmine-node spec
```

Failures:

```
1) Multiplication should multiply 2 and 3
```

```
Message:
```

```
Expected undefined to be 6.
```

```
Stacktrace:
```

```
Error: Expected undefined to be 6.
```

```
at null.<anonymous> (/Users/munen/Dropbox/ZHAW/web3-unterlagen/001_nodejs/labor/tdd-jasmin
```

Finished in 0.009 seconds 1 test, 1 assertion, 1 failure, 0 skipped

Ergänzen Sie `lib/calculator.js` entsprechend der Anforderung aus dem Spec, sodass der Test fehlerfrei durchläuft:

```
.
```

Finished in 0.008 seconds

1 test, 1 assertion, 0 failures, 0 skipped

Es ist mühsam, jasmine immer von Hand zu starten:

Führen Sie folgenden Befehl aus:

```
$ jasmine-node spec --autotest --watch .
```

Jetzt überwacht jasmine das gesamte Projekt (`--watch .`) und testet automatisch (`--autotest`) alle specs, die sich im Ordner `spec` befinden.

Fügen Sie einen weiteren Spec hinzu:

```
it("should multiply 2 and 7", function() {  
    var product = calculator.multiply(2,7);  
    expect(product).toBe(14);  
});
```

und vervollständigen Sie dann die `multiply` Funktion.

Erwartete Resultate und Abgabe

- Installierte Node.JS Umgebung
- Sie können Programme mit Node laufen lassen
- Sie können jasmine-node benutzen
- Abgabe während dem Labor

Ressourcen

- [Homebrew](#)
- [NVM](#)
- [NPM](#)
- [Node.js](#)