Angular Dependencies

package.json

**webpack**

Diesen Module-Loader nutzen wir in unserer Anwendung. Mit einem Module-Loader müssen wir nicht mehr dutzende Script Tags in unserer index.html einbinden. Stattdessen importieren wir das Haupt- bzw. Startmodul in diesem Module-Loader und auf Grund der „import“ Statements, die wir in unserem Code verwenden, lädt es automatisch die Module in der richtigen Reihenfolge. In produktiven Anwendungen können wir auch ein Bundle all unserer Anwendungsmodule erstellen. Es gibt noch weitere Module-Loader wie beispielsweise: browserify und system.js

**ie-shim**

Diese Bibliothek stellt die Kompatibilität zu aktuellen Internet Explorer Versionen her, die unsere Anwendung benötigt um auf allen gängigen Browsern fehlerfrei dargestellt zu werden. Mit dem Microsoft Edge ist dies nicht mehr nötig, da dieser volle Typescript Unterstützung erhielt.

**core-js**

Modulare Standard-Bibliothek für JavaScript, inklusive Polyfills [1] für ECMAScript 5 [2].

ECMAScript 6: Promises, Symbols, Collections, Iteratoren, Typed Arrays

ECMAScript 7+: Proposals, setImmediate usw.

Sie enthält außerdem einige zusätzliche Funktionen wie Dictionaries oder Extended Partial Application. Du kannst diese Bibliothek vollumfänglich benutzen oder nur einzelne Features mittels „require“ einbinden.

**@angular/...**

Einzelne Module aus dem Angular 2 Package

1 <https://de.wikipedia.org/wiki/Polyfill>

2 <https://de.wikipedia.org/wiki/JavaScript>

**typescript**

Typescript ist eine vom Unternehmen Microsoft entwickelte Programmiersprache, die auf den Vorschlägen zum zukünftigen ECMAScript-6-Standard basiert. Sprachkonstrukte von Typescript wie Klassen, Interfaces, Vererbung, Module, anonyme Funktionen, Generics und eine statische Typisierung sollen auch in ECMAScript 6 übernommen werden.

**typings**

Der Typescript Compiler löst Node Modulnamen durch den Node.js Modul Auflöse Algorithmus auf. Typescript kann auch Typisierungen laden, die mit dem Node Package Manager (npm) installiert wurden.

**rxjs** [3]

Das ist die Bibliothek Reactive Extensions für JavaScript, die eine elegante Art und Weise bereitstellt, um mit asynchronen Operationen zu arbeiten. Angular verwendet ein Konzept namens "Observables" in der Umsetzung ihrer Http und JSONP Klassen. Angular 1 verwendet Promises, aber Angular 2 ausschließlich Observables. Diese wurden allerdings für die Verwendung in Angular 2 Applikationen im Funktionsumfang stark reduziert.

**zone.js**

Eine der Komplexitäten in Angular 1 ist die Digest Loop, die Angular 1 hilft Änderungen im Modell festzustellen und die View zu aktualisieren. In Angular 2 haben wir dieses Konzept nicht mehr. In Angular 2 sind alle Browser-Ereignisse "Monkey-Patched". Das heißt es werden auch Event-Handler außerhalb der Angular App noch benachrichtigt und können Änderungen in Objekten erkennen. Das Angular Team hat diesen Teil aus dem Core Angular Skript extrahiert und als eine separate Open Source Bibliothek zur Verfügung gestellt, die von anderen wiederverwendet werden kann.

3 <http://reactivex.io/>