

WEB FRONTEND DEVELOPMENT MIT ANGULAR 2.0

Template-Syntax

Das Templating mit AngularJS war bereits ein mächtiges Werkzeug. Mit Angular 2.0 legen die Entwickler nun kräftig nach.

Der vorangegangene Artikel »Angular 2.0 und modularer Code« erläuterte, wie mit SystemJS Bibliotheken und eigene Client-Side-Skripts geladen und ausgeführt werden können. Ein erstes Hello-World-Beispiel mit Angular 2.0 wurde entwickelt. Das heißt, dass die Ausführung von ECMA-Script-6-Modulen nun keine Hürde mehr ist. Es wird Zeit, tiefer in das Framework einzutauchen.

Dieser Artikel stellt die neue Template-Syntax von Angular 2.0 vor. Es halten zahlreiche, neue Möglichkeiten Einzug, um Oberflächenelemente zu beschreiben. Die Entwickler von Angular verfolgen hierbei ein großes Ziel: das Konzept der Template-Syntax eindeutiger und nachvollziehbarer zu formulieren, als es bei der Vorgängerversion der Fall ist. Auch der Support durch Editoren, etwa durch bessere automatische Vervollständigung, soll nun verbessert werden und die Produktivität des Entwicklers steigern.

Zur näheren Erläuterung wird ein Prototyp genutzt, der als Dashboard für Schäden an Autos dienen soll (Bild 1).

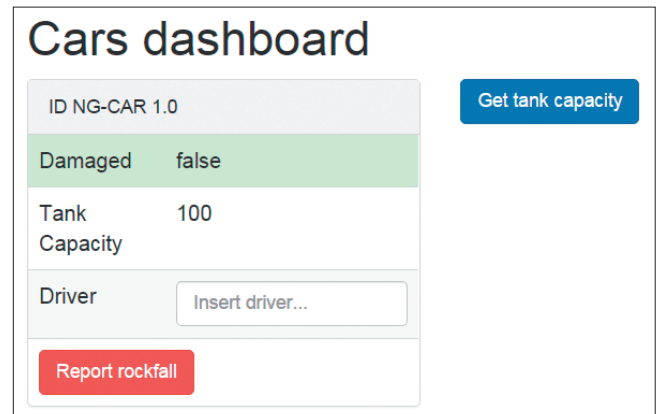
Neben einer ID und dem Schadensstatus kann auch der aktuelle Füllstand des Fahrzeugs abgefragt werden. Des Weiteren kann mit einem Klick ein Steinschlag (rockfall) gemeldet werden.

Übrigens Sie finden das hier vorgestellte Beispiel auf GitHub unter http://bit.ly/web_und_mobile_angular2_artikel2 und auf www.webundmobile.de.

Components und Views

Angular-2-Anwendungen bestehen aus verschiedenen Komponenten (Components), die miteinander agieren können. Für das Dashboard wird eine Komponente benötigt. Im Dashboard wird eine Liste von Autos abgebildet werden. Hierfür wird eine weitere Komponente implementiert. Den Aufbau einer Angular-2.0-Komponente zeigt Listing 1.

Von Angular werden zunächst zwei Module `@Component()` und `@View()` importiert. Diese beiden Module sind im Speziellen TypeScript-Dekoratoren. Dekoratoren ermöglichen es, Klassen durch Meta-Angaben erweitern. `@Component()` spezifiziert, dass die Dashboard-Komponente über den Selector `<dashboard>` im DOM des HTML-Dokuments eingesetzt wird. Mit `@View()` definiert man das Template, das mit der Komponenten verknüpft ist. In diesem Beispiel wird das Feld `id` aus der Klasse `DashboardComponent` im Template gebunden und angezeigt. An dieser Stelle wird deutlich, was eine Komponente ist: Komponenten sind die neuen zentralen Bausteine von Angular 2.0. Sie übernehmen die Rolle von Direktiven und Controllern aus AngularJS.



Dashboard-Prototyp: Dashboard für die Eingabe von Schäden an Autos (Bild 1)

Eine Komponente ist ein angereichertes Template, das im Browser zur Anzeige gebracht wird. Das Template wiederum verfügt über ein spezifisches Verhalten, das in Angular 2.0 durch TypeScript-Dekoratoren beschrieben wird.

Interpolation

Wie wird nun aus dem Ausdruck `{{ id }}` ein angezeigter Text im Browser?

Bereits in AngularJS 1.x konnten Daten mit Hilfe zweier geschweifter Klammern an ein HTML-Template gebunden werden. Der Wert wurde dann mittels Interpolation ausgewertet und angezeigt. Dieses Konzept bleibt in Angular 2.0 erhalten:

```
<p>{{ id }}</p>
```

Listing 1: Aufbau eine Komponente in Angular 2.0

```
import { Component, View } from 'angular2/angular2';

@Component({ selector: 'dashboard' })
@View({
  template: `<p>{{ id }}</p>`
})
export default class DashboardComponent {
  id: string = 'NG-Car 2015';
}
```