

## FORMULARVALIDIERUNG MIT ANGULAR 2

# Komfortables Handling

Formulare sind eine der wichtigsten Aufgaben moderner Webframeworks.

Dies ist der vierte Artikel in unserer Reihe zum Framework Angular 2. In den vorangegangenen Artikeln haben wir bereits SystemJS, Templates, Dependency Injection, Unit-Testing und asynchrone HTTP-Kommunikation kennengelernt. Mit dabei ist stets das Car Dashboard, das kontinuierlich neue Funktionen erhält. In diesem Artikel soll nun die Verarbeitung von Formulardaten im Mittelpunkt der Betrachtung stehen (Bild 1).

Als Webentwickler haben wir regelmäßig mit HTML-Formularen zu tun. Sie dienen der Gewinnung relevanter Informationen durch Benutzereingaben, welche anschließend weiterverarbeitet werden sollen.

Neben der bloßen Eingabe der Daten spielt es eine wichtige Rolle, ein adäquates visuelles Feedback an den Benutzer weiterzugeben. Fehleingaben sollen frühzeitig signalisiert werden. Auf Änderung der Eingabedaten muss ein Formular schnell reagieren können. Angular 2 hilft dabei, diese Anforderungen umzusetzen.

Wie gewohnt steht ein komplettes, lauffähiges Beispiel auf GitHub unter <https://github.com/Angular2Buch/angular2-forms> zur Verfügung, das die hier besprochenen Inhalte demonstriert.

## Controls und ControlGroups als Basis der Formularverarbeitung

Die Basis der Formularverarbeitung sind die Klassen *Control* und *ControlGroup*.

Um sie in einer Komponente zu verwenden, müssen sie aus dem Namensraum *angular2/common* importiert werden:

```
import {Control, ControlGroup} from 'angular2/common';
```

Ein Control hat einen Wert und einen bestimmten Validierungsstatus, der durch eine zusätzliche Funktion bestimmt werden kann. Daraus ergeben sich für das Control drei optio-

Tabelle 1: Zustandsinformationen eines Controls

Status	Beschreibung
<i>dirty</i>	Der Wert wurde bearbeitet
<i>pristine</i>	Der Wert ist unberührt
<i>valid</i>	Der Wert ist gültig
<i>touched</i>	Das Control wurde verwendet/bedient
<i>untouched</i>	Das Control wurde noch nicht verwendet
<i>pending</i>	Eine asynchrone Eingabevalidierung läuft

Angular 2: Endlich im Beta-Status (Bild 1)

nale Parameter, ein Wert, sowie ein Validator und ein asynchroner Validator:

```
this.nameControl = new Control('Initialer Wert',
/* synchroner Validator -> */ Validators.required,
/* asynchroner Validator -> */ IdValidator.uniqueId);
nameControl.value; // -> 'Initialer Wert'
nameControl.errors; // -> undefined
```

Das Control kann im HTML-Template der Komponente über die Direktive *ngControl* gebunden werden:

```
<input type="text" ngControl="nameControl" required>
```

Je nach Zustand des Controls fügt *ngControl* automatisch CSS-Klassen an das jeweilige Eingabeelement hinzu:

```
<input type="text"
class="ng-pristine ng-valid ng-touched"
ngcontrol="nameControl"
required>
```

Da der Status des Controls (Tabelle 1) durch CSS-Klassen repräsentiert wird, kann das Element entsprechend gestaltet werden:

```
.ng-valid {
border-left: 5px solid #42A948;
}
```