



2017

Angular 5

Beispiele

Submitted By:

BearingPoint Technology

Seering 6

8141 Premstätten/Graz

Austria

Authors

Ing. Christoph Kerschenbauer

1. Angular JS Beispiele

1.1 *Beispiel 1 - Taschenrechner*

Erstelle einen Taschenrechner im Menüpunkt "Rechner"

- Der Taschenrechner soll in einer HTML Form sein
- Es soll eine Überschrift „Taschenrechner“ geben
- Er soll einen Eingabewert, einen Operator und einen 2. Eingabewert haben. Erstelle dies mit 3 Textfeldern.
- Der Operator soll die vier Grundrechnungsarten beherrschen
- Führe die Berechnung in deiner Component durch
- Das Ergebnis soll aufgrund eines Submit-Events berechnet werden und das Ergebnis ausgegeben werden
- Die Seite darf bei einem Form Submit (Abschicken der Form) nicht neu geladen werden
- Wird kein gültiger Wert eingegeben, so wird eine Fehlermeldung unterhalb des Eingabefeldes angezeigt

Optional:

- Es gibt einen „Zurücksetzen“ Button, der die Eingabefelder wieder leert
- Das Ergebnisfeld soll nur angezeigt werden, wenn ein valides Ergebnis vorhanden ist
- Erweitere deinen Rechner um weitere Funktionen wie Prozentrechnen, Wurzelberechnung etc.
- Erstelle für die Rechnungsarten Buttons. Durch auf den jeweiligen button klicken wird die Art der Operation angegeben.
- Erstelle anstatt der Zahleneingabefelder Buttons von 1-9 und erhalte die Zahlen anhand der geklickten Buttons

1.2 Beispiel 2 – Gästeliste

- Erstelle einen neuen Menüpunkt "Gästeliste" auf der selben Ebene wie der Menüpunkt "Rechner"
- Erstelle die zugehörige Route und verweise darin auf deine neu erstellte Component, der die View anzeigt. (Siehe Beispiel Taschenrechner)
- Erstelle in der View eine Überschrift „Gästeliste“
- Erstelle eine Unterüberschrift „Eingeladene Gäste“
- Es soll darunter eine Tabelle geben, die folgendes Format besitzt

Beispieltabelle

Nachname	Vorname	Alter	Geschlecht
Eisenhauer	Dwight	30	Male
Bucannan	Johanna	28	Female
Mars	Mary	15	Female
.			
.			
.			

Die Daten von der Tabelle sollen von einem JSON Object ausgelesen werden

Folgendes Format: Array von Gast Objekten

```
{
  "guestList": [
    {
      "surname": "Wayne",
      "name": "John",
      "age": 25,
      "gender": "male",
      "attending": false
    },
    {
      "surname": "Eisenhauer",
      "name": "Dwight",
      "age": 30,
      "gender": "female",
      "attending": false
    },
    {
      "surname": "Bucannan",
      "name": "Johanna",
      "age": 28,
      "gender": "female",
      "attending": false
    }
  ]
}
```

```
    },
    {
      "surname": "Mars",
      "name": "Joy",
      "age": 15,
      "gender": "female",
      "attending": false
    },
    {
      "surname": "Queen",
      "name": "Mary",
      "age": 28,
      "gender": "female",
      "attending": false
    },
    {
      "surname": "Krause",
      "name": "Peter",
      "age": 95,
      "gender": "male",
      "attending": false
    },
    {
      "surname": "Bach",
      "name": "Sebastian",
      "age": 50,
      "gender": "male",
      "attending": false
    },
    {
      "surname": "Planet",
      "name": "Erika",
      "age": 27,
      "gender": "female",
      "attending": false
    },
    {
      "surname": "Smith",
      "name": "Patrick",
      "age": 40,
      "gender": "male",
      "attending": false
    },
    {
      "surname": "Eisenhauer",
      "name": "Samantha",
      "age": 60,
      "gender": "female",
      "attending": false
    }
  ]
}
```

Optional:

- Ich kann einen neuen Gast durch einen Buttonklick mit Dummywerten hinzufügen. (z.B. TestNachname, Testvorname usw.)
 - Diese Daten können nicht eingegeben werden und sind fix vordefiniert
 - Dieser bleibt, sofern die Seite nicht neu geladen wird, in meiner Liste gespeichert und wird in der Tabelle angezeigt
 - Ich kann diesen Dummy Gast öfters hinzufügen
- Erstelle ein Model für deine Gästeliste und verwende typisierte Methoden
- Erweitere deine View um eine Checkbox. Diese Checkbox wird als 1. Spalte angezeigt.
 - Wird die Zeile oder die Checkbox geklickt, so wird die Checkbox aktiviert. Setze dazu das Flag attending auf true bei Aktivierung bzw. auf false bei Deaktivierung der Zeile
- Lege eine Unterüberschrift „Zugesagt“ an. Zeige darunter eine Tabelle mit Nachname und Vorname derjenigen Personen an, die zugesagt haben. (Flag attending = true)
 - Die Liste soll bei jedem Klick aktualisiert werden
 - Tipp: Pipe erstellen – siehe Angular5 Dokumentation
 - Tipp: die Pipe muss auf pure: false sein – siehe Angular5 Dokumentation
- Es wird eine weitere spalte erstellt am Ende erstellt. Diese beinhaltet keine Überschrift und als Wert nur einen Button mit dem Label „Löschen“
 - Wird der Button geklickt, so wird die aktuelle Zeile gelöscht.
- Ich möchte einen Gast mit Eingabedaten hinzufügen
 - Es gibt 4 Inputs jeweils für ein Attribut und 1 Button zum Speichern
 - Beim Speichern wird der Gast in der Tabelle hinzugefügt.
 - Stelle sicher, dass nur Male und Female als Wert für Geschlecht steht.
 - Die 3 Unterpunkte des vorhergehenden Punktes gelten auch für diesen.
- Lade deine Gästeliste vom Backend mit dem Angular http Service.
 - URL: /api/maturaball/guest-list
 - Speichere das retournierte JSON Objekt in eine Variable anstatt der hardcodierten Liste

1.3 Beispiel 3 - Anmeldeformular

- Erstelle einen neuen Menüpunkt Anmeldeformular
- Erstelle die zugehörigen Route sowie Component (überlege dir einen sinnvollen Namen, siehe Beispiel Taschenrechner)
- Erstelle in der View eine Überschrift „Anmeldeformular“
- Das Formular soll folgende Eingabewerte haben

Anmeldeformular	
Anrede	Radio Button mit Herr und Frau (Pflichtfeld, Standard Frau)
Titel	Select mit Titel (Ing. BSc ..., Optional, Standard nichts ausgewählt)
Nachname	Textfeld (Max 255 Zeichen, Pflichtfeld)
Vorname	Textfeld (Max 255 Zeichen, Pflichtfeld)
Firma	Textfeld (Max 255 Zeichen, Optional)
Straße	Textfeld (Max 255 Zeichen, Optional)
PLZ	Textfeld, (Max 4 Zeichen, nur Nummern erlaubt, Pflichtfeld)
Ort	Textfeld, (Max 255 Zeichen, Pflichtfeld)
Telefonnummer	Textfeld (Max 32 Zeichen, Optional)
Bundesland	Select mit Bundesländer (Standardmäßig nichts ausgewählt, Pflichtfeld)
Bezirk	Select mit Bezirken (Standardmäßig nichts ausgewählt, Pflichtfeld)
Gerne würde ich ihren Newsletter abonnieren	Checkbox, Standardmäßig angehakt
	Felder mit * sind Pflichtfelder
	„Anmelden“ Button

- Bevor nicht alle Pflichtfelder ausgefüllt sind, kann ich das Formular nicht abschicken

1.4 Beispiel 4 - Bibliothek

- Erstelle einen neuen Menüpunkt "Bibliothek" auf der selben Ebene wie der Menüpunkt "Rechner"
- Erstelle die zugehörige Route und verweise darin auf deine neu erstellte Component, der die View anzeigt
- Erstelle in der View eine Überschrift „Bibliothek“
- Erstelle eine Unterüberschrift „Bücherliste“
- Es soll darunter eine Tabelle geben, die folgendes Format besitzt
Beispieltabelle

Titel	Author	Kurzbeschreibung	Jahr
Der Herr der Ringe	J.R.R. Tolkien	Ein Ring muss zerstört werden	1954
Harry Potter	Joanne K. Rowling	Ein Junge erkennt allmählich, dass er ein Zauberer ist	1997
.			
.			
.			

Die Daten von der Tabelle sollen von einem JSON Object ausgelesen werden

Folgendes Format: Array von Bücher Objekten

```
{
  "bookList": [
    {
      "title": "Der Herr der Ringe",
      "person": "J.R.R. Tolkien",
      "year": "1954",
      "description": "Ein Ring Muss Zerstört werden",
      "longDescription": "Frodo erhält einen Ring von seinem Onkel Bilbo. Gemeinsam..."
    }
  ]
}
```

- Erstelle in deinem Service ein Json Objekt mit sinnvollen Buchdaten und zeige diese in der Liste an
 - Mindestens 5 Einträge
- Erstelle das zugehörige Model/Klasse Bibliothekseintrag und typisiere die Methoden
- Erstelle eine weitere Spalte ohne Überschrift mit einem Button um die aktuelle Zeile zu löschen

- Erstelle eine weitere Überschrift „Filmliste“.
- Es soll darunter eine Tabelle geben, die folgendes Format besitzt
Beispieltabelle

Titel	Regisseur	Kurzbeschreibung	Jahr
Inception	Christopher Nolan	Traum oder Realität	2010
Der Pate	Francis Ford Coppola	Die Familie steht an erster Stelle	1972
.			
.			
.			

Die Daten von der Tabelle sollen von einem JSON Object ausgelesen werden

Folgendes Format: Array von Film Objekten (Aufbau ist Ident mit Bücherliste)

```
{
  "movieList": [
    {
      "title": "Der Herr der Ringe",
      "person": "J.R.R. Tolkien",
      "year": "1954",
      "description": "Ein Ring Muss Zerstört werden",
      "longDescription": "Frodo erhält einen Ring von seinem Onkel Bilbo. Gemeinsam..."
    }
  ]
}
```

- Erstelle in deinem Service ein Json Objekt mit sinnvollen Filmdaten und zeige diese in der Liste an
- Verwende das zugehörige Model/Klasse und typisiere die Methoden
- Erstelle eine weitere Spalte ohne Überschrift mit einem Button um die aktuelle Zeile zu löschen
- Lagere die Tabellen in eine eigene Komponente Bibliothekliste aus und rufe diese in deiner View auf.
 - Als Input Parameter der Komponente wird die Liste der anzuzeigenden Einträge erwartet.
 - Diese Komponente wird nicht über den Router angesteuert, sondern über den Selektor der Komponente
- Erstelle ein Formular unterhalb der Tabellen um ein weiteres Element anzulegen.
 - Erstelle eine Auswahl ob ein Buch oder ein Film angelegt wird und speichere das Element in die jeweilige Liste

- Erstelle einen Bereich auf deiner Seite, welcher bei Mausklick auf den Eintrag in der Liste den vollständigen Eintrag anzeigt.
 - inklusive längerer Beschreibung
- Bereich ist für beide Listen derselbe (Es kann immer nur 1 Element ausgewählt sein)

- Erstelle eine Backend-Mock-Schnittstelle (Folder Mock) mit folgenden URL's
 - GET: /api/bibliothek/buecher
 - Beinhaltet das im Beispiel erwähnte Bücher Json Object
 - GET: /api/bibliothek/filme
 - Beinhaltet das im Beispiel erwähnte Filme Json Object
 - Achte auf die server.js und erweitere den Pfad (Beispiel Maturaball)
 - Änderungen im Mock werden nur nach Neustart des Servers übernommen (npm start)
- Binde deine Listen über das http Service anstatt des JSON Objektes ein
- Bei einem Klick auf ein Listenelement sollen Daten editiert werden können
 - Diese bleiben solange gespeichert, solange ich die Seite nicht neu lade.
 - Der Bereich für die erweiterte Anzeige zuvor soll dafür genutzt werden
- Überlege welche Logik du zusammenfassen kannst und Lagere diese in ein Service aus bzw. lege weitere Komponenten an (Stichwort löschen und editieren der Einträge)

1.5 CSS Klassen

Folgende Klassen Können mit Bootstrap CSS

```
<form>  
  <div class="form-group">  
    <div class="input-group">  
      <input class="form-control" />  
      <button class="btn btn-default" />  
    </div>  
  </div>  
</form>
```

```
<table class="table table-bordered table-striped">
```

Für th, tr und td sind keine CSS Klassenattribute notwendig. Dies wird über das Bootstrap CSS gesteuert.