

1. Komponenten und Interpolation [U-020]

Als Nutzer möchte ich auf einer einfachen Website ein Todo als Text und die E-Mail-Adresse des Erstellers angezeigt bekommen, um mich daran zu erinnern, was ich noch zu erledigen habe.

1. Lernziele

- ✓ Sie kennen die Struktur einer einzelnen Komponente.
- ✓ Sie nutzen die Template-Syntax von Angular, um eine Webseite mittels einer dynamischen Komponente zu implementieren.
- ✓ Sie binden Styles in Form von CSS-Dateien in ein Template ein.

2. Ergebnis

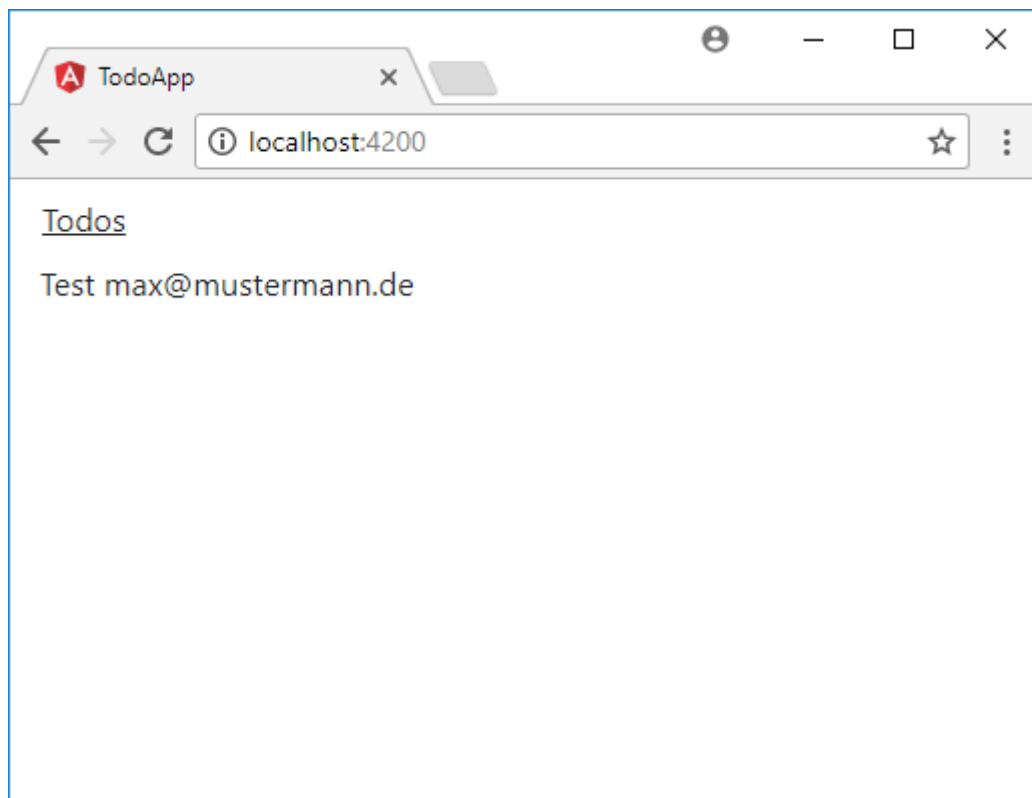


Abbildung 1 - Einfache Website, die ein Todo mit E-Mail-Adresse des Erstellers anzeigt.

3. Benötigte Dateien

- `src/app/app.component.ts`
- `src/app/app.component.html`
- `src/app/app.component.css`
- `src/app/shared/todo-item.ts`

4. Anleitung

Schritt 1: In der Klasse **AppComponent** eine öffentliche Objektvariable **title** vom Typ **string** definieren und im Template ausgeben

1. Definieren Sie in der Datei **app.component.ts** innerhalb der Klasse **AppComponent** ein öffentliches Feld **title** vom Typ **string**.
2. Weisen Sie dem Feld **title** einen beliebigen Wert (z.B. „**Todos**“) zu.
3. Entfernen Sie den gesamten Inhalt der Datei **app.component.html**.
4. Erstellen Sie ein **<nav>**-Element, in welchem Sie den eben definierten Wert von **title** ausgeben. Weisen Sie dem Element die CSS-Klassen „container navbar“ zu.
5. Führen Sie die Anwendung aus und betrachten Sie das Ergebnis.

Schritt 2: In der Klasse **AppComponent** eine neue öffentliche Objektvariable **todo** vom Typ **TodoItem** definieren und initialisieren

6. Importieren Sie in der Datei **app.component.ts** die Klasse **TodoItem** aus **src/app/shared/todo-item.ts**.
7. Definieren Sie in der Klasse **AppComponent** ein öffentliches Feld **todo** vom Typ **TodoItem**.
8. Initialisieren Sie das **todo**-Feld indem Sie ihm neues **TodoItem** mit von Ihnen erfundenen Werten zuweisen.

Schritt 3: Felder **title** und **email** von **todo** im Template anstelle des Begrüßungstextes ausgeben

9. Erstellen Sie nach dem **<nav>** Element in **app.component.html** ein Tagpaar **<main>**, in welchem Sie eine **<div>** schachteln. Weisen Sie **<main>** die CSS-Klasse „container“ zu. Woran erkennt Angular, dass diese Datei das Template für **AppComponent** ist?
10. Geben Sie innerhalb des **<div>** Paares den Wert für den Titel sowie die Email Adresse des Todos per Interpolation aus:

Beispiel für lediglich den Titel.

```
<div> {{todo.title}} </div>
```

Schritt 4: CSS-Datei für das Styling der Komponente nutzen, um dem Element **<nav>** in der CSS-Datei ein Styling zuzuweisen

11. Erstellen Sie in der bislang leeren CSS-Datei **app.component.css** eine neue Regel für **<nav>**-Elemente.
12. Spezifizieren Sie **text-decoration: underline** (unterstreichen) als Regel für alle **<nav>** Elemente.

5. Kontrollfragen

- Was sind die typischen Bestandteile einer Komponente?
- Welche Möglichkeiten existieren, um ein Template für eine Komponente anzugeben?
- Mittels welcher Syntax können z.B. Objektvariablen einer Komponentenkasse im Template ausgegeben werden?

6. Weiterführende Materialien

- Angular-Komponenten
<https://tinyurl.com/gs-angular-arch-components>
- Angular-Template-Syntax
<https://tinyurl.com/gs-angular-template-syntax>
- Angular Komponenten Style
<https://tinyurl.com/gs-angular-component-style>
- Angular Interoperation
<https://tinyurl.com/gs-angular-interoperation>
- Angular Life Cycle Hooks
<https://tinyurl.com/gs-angular-lifecycle-hooks>