

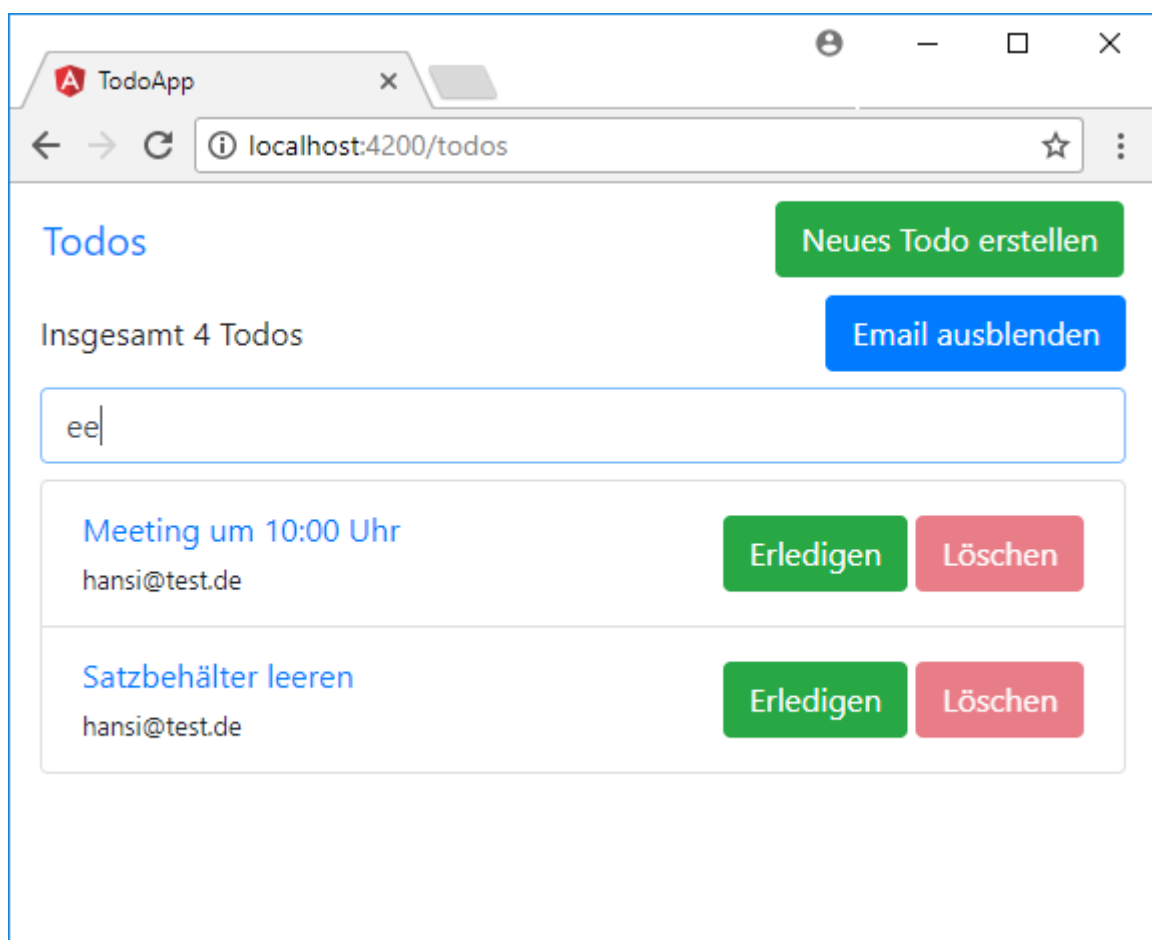
# 1. Pipes [U-100]

Als Nutzer möchte ich gezielt nach Todos suchen, um schnell und gezielt die gesuchte Information zu finden, insbesondere, wenn meine Todo-Liste sehr lang ist.

## 1. Lernziele

- ✓ Sie nutzen Pipes, um im Front-End Daten für die Darstellung zu filtern.

## 2. Ergebnis



### 3. Benötigte Dateien

- `src/app/todo-list.component.html`
- `src/app/todo-list.component.ts`
- `src/app/search.pipe.ts`
- `src/app/app.module.ts`

### 4. Anleitung

---

Schritt 1: Suchfeld im Template der `TodoListComponent` erstellen und mittels `ngModel` an eine neu zu erstellende Objektvariable `searchQuery` vom Typ `string` binden

1. Erstellen Sie im Template der `TodoListComponent` über der Liste von Todos ein `<input>`-Element. Geben Sie für das `placeholder`-Attribut des Elements den Text „Todo suchen“ an
2. Definieren Sie in der Klasse `TodoListComponent` eine neue Objektvariable `searchQuery` und weisen Sie dieser einen leeren `string` zu.
3. Binden Sie `searchQuery` mittels `ngModel` an das `<input>`-Element.

`<input placeholder="Todo suchen" [(ngModel)]="searchQuery">`

---

Schritt 2: Neue Pipe `SearchPipe` mit dem Namen `search` definieren

4. Erstellen Sie eine Klasse `SearchPipe` in `src/app/pipes/search.pipe.ts`.
5. Dekorieren Sie die Klasse mit dem `@Pipe`-Dekorator aus `@angular/core` und geben Sie dabei für `name` den Wert `search` an.
6. Nehmen Sie die Klasse `SearchPipe` unter `declarations` in `AppModule` auf.

---

Schritt 3: In Klasse **SearchPipe**, welche das **PipeTransform**-Interface implementiert, die Methode **transform()** mit den Parametern **todos: TodoItem[]** und **query: string** implementieren. Diese Methode wird Daten vom Typ **TodoItem[]** zurückgeben.

7. Implementieren Sie in der Klasse **SearchPipe** das Interface **PipeTransform** aus dem Modul **@angular/core**.
8. Definieren Sie die in der Klasse **SearchPipe** die Methode **transform** mit den Übergabeparametern **todos** vom Typ **TodoItem[]** sowie **query** vom Typ **string** und den Rückgabebetyp **TodoItem[]**.

---

Schritt 4: Geben Sie in der Methode **transform** aus einem Array von Todos nur diejenigen Todos zurück, deren Titel einen im zweiten Parameter übergebenen Suchstring beinhaltet.

9. Rufen Sie die Methode **filter** auf dem Array **todos** auf.
10. Übergeben Sie **filter()** eine Pfeilfunktion, welche ein **todo** vom Typ **TodoItem** als Parameter erhält und einen Rückgabewert vom Typ **boolean** hat.
11. Rufen Sie als Rückgabewert der Pfeilfunktion die Methode **includes** von **todo.title** auf. Als Parameter erhält **includes** die Variable **query**.

```
stringValue.includes(query)
```

---

Schritt 5: **SearchPipe** in **TodoListComponent** verwenden, um nach Todos zu filtern, die **searchQuery** im Titel beinhalten.

12. Im **ngFor**-Ausdruck des **TodoListComponent**-Template fügen Sie einen Pipe-Operator ( **"|"** ) an.
13. Nach dem Pipe-Operator geben Sie den Namen der eben erstellten Pipe an, der im **@Pipe**-Dekorator spezifiziert ist.
14. Getrennt durch einen Doppelpunkt, fügen Sie den Variablennamen an, in dem das Eingabefeld den aktuellen Wert ablegen soll (**searchQuery**)

## 5. Kontrollfragen

- Welchen Zweck erfüllen Pipes in Angular?
- Wie werden Argumente an Pipes übergeben?

## 6. Weiterführende Materialien

- Angular- Dokumentation über das Erstellen angepasster Pipes  
<https://angular.io/guide/pipes#custom-pipes>