

My To Do List

What would you like to do today?

add an item

Add

4 items

☒ eat

Edit

Delete

☐ sleep

Edit

Delete

☐ play

Edit

Delete

☐ laugh

Edit

Delete

todo > src > app > item > TS item.component.ts > ...

```
1  import { Component, Input, Output, EventEmitter } from "@angular/core";
2  import { Item } from "../item";
3
4
5  @Component({
6    selector: 'app-item',
7    templateUrl: './item.component.html',
8    styleUrls: ['./item.component.css'],
9  })
10
11  export class ItemComponent {
12
13    editable = false;
14
15    @Input() item!: Item;
16    @Output() remove = new EventEmitter<Item>();
17
18    saveItem(description: string) {
19      if (!description) return;
20      this.editable = false;
21      this.item.description = description;
22    }
23  }
```

todo > src > app > item > <> item.component.html > ...

```
1  <div class="item">
2    <input
3      [id]="item.description"
4      type="checkbox"
5      (change)="item.done = !item.done"
6      [checked]="item.done" />
7    <label [for]="item.description">{{item.description}}</label>
8
9    <div class="btn-wrapper" *ngIf="!editable">
10      <button class="btn" (click)="editable = !editable">Edit</button>
11      <button class="btn btn-warn" (click)="remove.emit()">Delete</button>
12    </div>
13
14    <!-- This section shows only if user clicks Edit button -->
15    <div *ngIf="editable">
16      <input
17        class="sm-text-input"
18        placeholder="edit item"
19        [value]="item.description"
20        #editedItem
21        (keyup.enter)="saveItem(editedItem.value)" />
22
23      <div class="btn-wrapper">
24        <button class="btn" (click)="editable = !editable">Cancel</button>
25        <button class="btn btn-save" (click)="saveItem(editedItem.value)">
26          Save
27        </button>
28      </div>
29
30    </div>
31  </div>
32
```

todo > src > app > <> app.component.html > ...

```
1  <div class="main">
2    <h1>My To Do List</h1>
3    <label for="addItemInput">What would you like to do today?</label>
4
5    <input
6      #newItem
7      placeholder="add an item"
8      (keyup.enter)="addItem(newItem.value); newItem.value = ''"
9      class="lg-text-input"
10     id="addItemInput" />
11
12    <button class="btn-primary" (click)="addItem(newItem.value)">Add</button>
13
14
15    <h2>
16      {{items.length}}
17      <span *ngIf="items.length === 1; else elseBlock">item</span>
18      <ng-template #elseBlock>items</ng-template>
19    </h2>
20
21    <ul>
22      <li *ngFor="let i of items">
23        <app-item (remove)="remove(i)" [item]="i"></app-item>
24      </li>
25    </ul>
26
27  </div>
```

todo > src > app > ts app.component.ts > ...

```
1  import { Component } from "@angular/core";
2  import { Item } from "../item";
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9
10 export class AppComponent {
11   title = "todo";
12   filter: "all" | "active" | "done" = "all";
13   allItems = [
14     { description: "eat", done: true },
15     { description: "sleep", done: false },
16     { description: "play", done: false },
17     { description: "laugh", done: false },
18   ];
19   get items() {
20     if (this.filter === "all") {
21       return this.allItems;
22     }
23     return this.allItems.filter((item) =>
24       this.filter === "done" ? item.done : !item.done
25     );
26   }
27   addItem(description: string) {
28     this.allItems.unshift({
29       description,
30       done: false
31     });
32   }
33   remove(item: Item) {
34     this.allItems.splice(this.allItems.indexOf(item), 1);
35   }
36 }
```

TS app.component.ts M ● TS item.ts U X



todo > src > app > TS item.ts > ...

```
1  export interface Item {  
2      description: string;  
3      done: boolean;  
4  }  
5
```

Reference:

https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Angular_getting_started