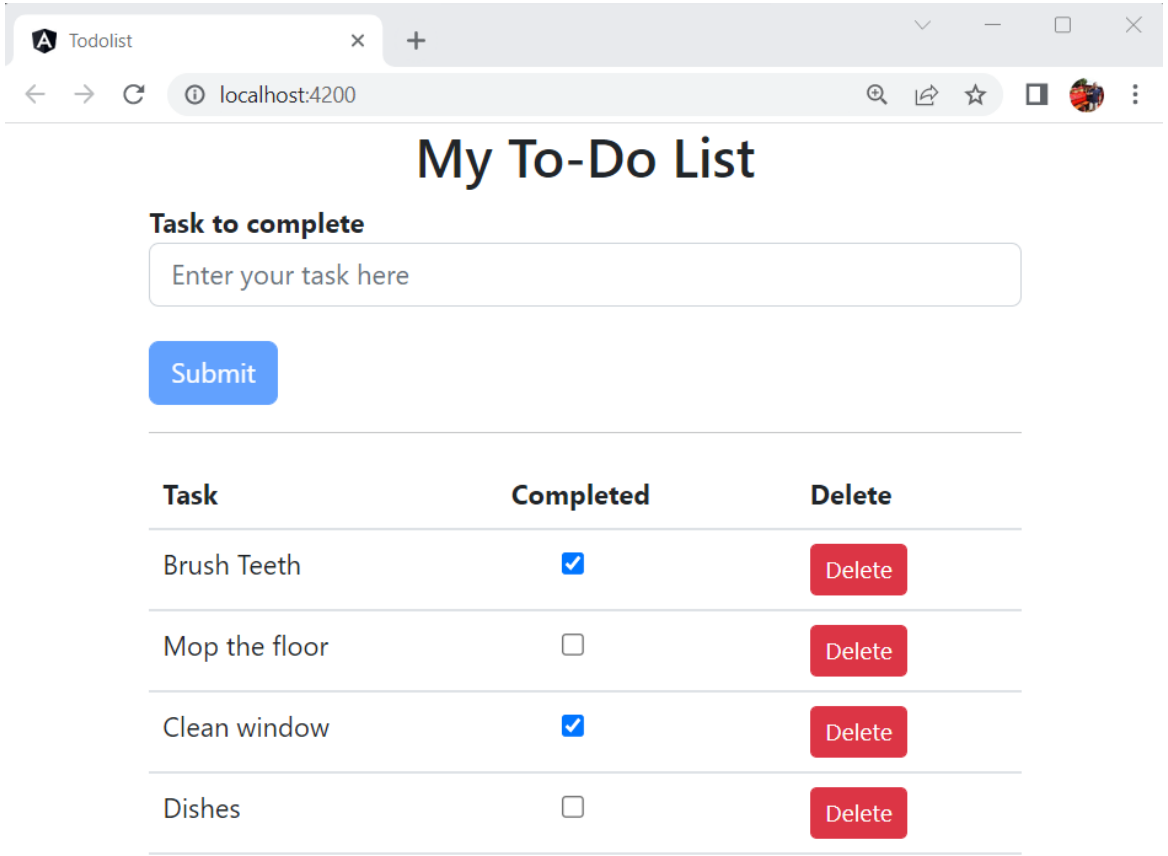
To-Do List



ANGULAR PROJECTS

Angular Form Types

https://angular.io/guide/forms-overview

	REACTIVE	TEMPLATE-DRIVEN
Setup of form model	Explicit, created in component class	Implicit, created by directives
Data model	Structured and immutable	Unstructured and mutable
Data flow	Synchronous	Asynchronous
Form validation	Functions	Directives

Angular template-driven form

Template-driven forms focus on simple scenarios and are not as reusable.

Angular template-driven form (steps)

- Add ‘FormsModule’ to imports in **app.module.ts**
- Modify <form></form> in the template of the component (**html file**)
- Modify the **ts file** of the component
- Add form validation if needed

Modify app.module.ts

- Add ‘FormsModule’ to imports in **app.module.ts**

```
import { FormsModule } from '@angular/forms';
```

```
imports: [
    BrowserModule,
    FormsModule
],
```

Modify html file of the component

- Add local reference to the form ‘#myForm’ and assign it to ‘ngForm’. Add ‘ngSubmit’ directive and assign it to the method located in the ts file. Pass the form reference to that method.

```
<form (ngSubmit)="onSubmit(myForm)" #myForm="ngForm">
```

- Mark controls (that you want to include) with ‘ngModel’ directive and add names to those controls

```
<input type="text"  ngModel  name="myControl">
```

Modify the ts file of the component

- Add import of the ‘NgForm’ class and create method with a parameter of type NgForm

```
import { NgForm } from '@angular/forms';
```

```
onSubmit(form: NgForm) {
  console.log(form);
}
```

Add validation if needed

- Validate controls if needed.

  Example: disable the ‘Submit’ button if the input field is empty

```
<input type="text" required ngModel name="myControl">
```

```
<button [disabled]="myForm.invalid" type="submit">Submit</button>
```

Validators

(built-in validators that can be used by form controls)

```
class Validators {
  static min(min: number): ValidatorFn
  static max(max: number): ValidatorFn
  static required(control: AbstractControl<any, any>): ValidationErrors | null
  static requiredTrue(control: AbstractControl<any, any>): ValidationErrors | null
  static email(control: AbstractControl<any, any>): ValidationErrors | null
  static minLength(minLength: number): ValidatorFn
  static maxLength(maxLength: number): ValidatorFn
  static pattern(pattern: string | RegExp): ValidatorFn
  static nullValidator(control: AbstractControl<any, any>): ValidationErrors | null
  static compose(validators: ValidatorFn[]): ValidatorFn | null
  static composeAsync(validators: AsyncValidatorFn[]): AsyncValidatorFn | null
}
```

https://angular.io/api/forms/Validators