Lab Exercises

Web Development Frameworks

zenika
ARCHITECTURE INFORMATIQUE

Lab 1: Angular Todo App

In this lab, we will explore building a Todo app using Angular!

Create the project

We will use `@angular/cli` to setup an application that provides:

- Application scaffolding and configuration
- Dependencies management

```
npm install @angular/cli
ng new angular-todo-app
```

We have created a new application. Take a look at the structure:

- Dependencies
- TypeScript configuration
- TypeScript files

To run the application

```
cd angular-todo-app
ng serve --open
```

This compiles the sources first and then starts the server

You can now open the broswer at `http://localhost:4200` to view the application

Updating HTML

In `app.component.html` , remove generated HTML and paste the following code

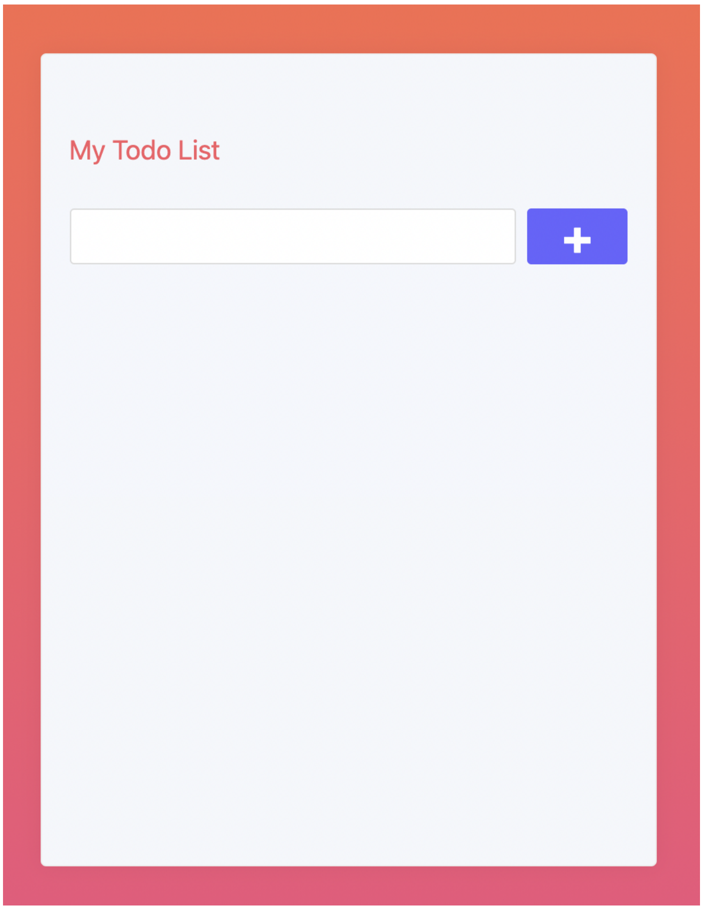
```
<style>
  html,
  body {
    margin: 0;
    height: 100%;
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
      Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;
  }

  main {
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: #fefefe;
    background-image: linear-gradient(#fc6c48 0%, #ef5081 100%);
    background-repeat: no-repeat;
```

```css
  background-size: cover;
  height: 100%;
}

.todo-wrapper {
  width: 400px;
  max-width: 100%;
  min-height: 500px;
  margin: 20px auto 40px;
  border: 1px solid #eee;
  border-radius: 4px;
  padding: 40px 20px;
  -webkit-box-shadow: 0 0 15px 0 rgba(0, 0, 0, 0.05);
  box-shadow: 0 0 15px 0 rgba(0, 0, 0, 0.05);
  background-color: #f4f7fc;
  overflow: hidden;
  position: relative;
}

.todo-title {
  font-size: 1.2em;
  color: #f65c65;
  font-weight: normal;
}

.btn,
input {
  line-height: 2em;
  border-radius: 3px;
  border: 0;
  display: inline-block;
  margin: 15px 0;
  padding: 0.2em 1em;
  font-size: 1em;
}

input[type="text"] {
  border: 1px solid #ddd;
  min-width: 70%;
  transition: all ease-in 0.25s;
}

input:focus {
  outline: none;
  border: 1px solid #a3b1ff;
}

.btn {
  text-align: center;
  font-weight: bold;
  cursor: pointer;
  border-width: 1px;
  border-style: solid;
}

.btn-add {
    background: #6664ff;
    border-color: #6664ff;
    pointer-events: visible;
    color: #fefefe;
    min-width: 12%;
    transition: all ease-in 0.25s;
    font-size: 2.2em;
    line-height: 0.5em;
    padding: 0.3em 0.3em;
    float: right;
  }

  ul.todo-list {
    padding: 0;
    margin-bottom: 30px;
  }

  ul.todo-list li {
    position: relative;
    list-style-type: none;
    display: block;
    margin: 10px 0;
    background: #e0e8f5;
    border-radius: 3px;
    padding: 12px;
    overflow: hidden;
  }
</style>

<main>
  <section class="todo-wrapper">
    <h2 class="todo-title">My Todo List</h2>
    <input type="text" />
    <div class="btn btn-add">+</div>
    <ul class="todo-list">
      <!-- list of todos here -->
    </ul>
  </section>
</main>
```

Also in `app.component.ts` file, update the code so that the above styles works properly

```
import { Component, ViewEncapsulation } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
  encapsulation: ViewEncapsulation.None,
})
export class AppComponent {}
```

The app should now look like this.



Adding Todos

Declare a todos variable in `app.component.ts` file and create an `addTodo` method to add new todo into the array.

```
export class AppComponent {
  todos: String[] = [];

  addTodo(todo: String) {
    this.todos.push(todo);
  }
}
```

Then in `app.component.html` file, update the input and button elements to use the the todos method.

```
    <input #inputValue type="text" />
    <div
      class="btn btn-add"
      (click)="addTodo(inputValue.value); inputValue.value = ''"
    >
      +
    </div>
```

Lastly, update the `ul` tag to display the added todos.

```
    <ul class="todo-list">
      <li *ngFor="let todo of todos">{{ todo }}</li>
    </ul>
```

That's all! You can now insert new todo into the list!