

Создание списка дел с использованием Angular

Вавилов Егор Дмитриевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будет рассмотрен процесс создания приложения списка дел, где имеется возможность добавления и удаления. Помощью в разработке служит фреймворк Angular. Практическим результатом является готовое приложение списка дел.

Ключевые слова: Angular, Приложение, Разработка

Creating a to-do list using Angular

Vavilov Yegor Dmitrievich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article will walk you through the process of creating a to-do list application where you can add and remove. Development is assisted by the Angular framework. The bottom line is a complete to-do list app.

Keywords: Angular, Application, Development

Angular – это фреймворк разработанный инженерами компании Google. Он служит для разработки динамических веб-приложений и становится все более востребованным фреймворком. Это надежный фреймворк для интерфейсной разработки и на сегодняшний момент в его пользу склоняются многие разработчики, также он отлично подходит для модульного тестирования. Angular расширяет синтаксис разметки HTML и так же упрощает процесс разработки.

Цель исследования – написать веб-приложение, которое будет сохранять список дел и при необходимости удалять.

Исследованиями в данной теме занимались следующие авторы. Д.В.Щербина, Н.Ю.Богданова представили проект создания мобильного приложения, функционал которого позволит находить ближайшую автозаправочную станцию к местоположению мобильного устройства [1]. Т.А.Алешина, В.Ю.Белаш рассказала про многие актуальные возможности данного фреймворка [2]. А.К.Оглуляк ознакомила читателей с данным фреймворком, рассказав что это такое, как он работает и как часто происходят обновления [3]. А.Л.Мархакшинов, А.С.Иванов рассмотрели

проект, который может быть использован для знакомства обучающихся с популярным фреймворком разработки веб-приложений Angular [4].

Для начала работы необходимо, чтобы на компьютере был установлен «Node.js», а так же Angular CLI.

После установки этих компонентов можно начать создание проекта. Для этого создадим новый проект командой «ng new my-app» и создадим структуру приложения «ng new todo-app». После этого структура проекта будет выглядеть следующим образом (Рис.1).

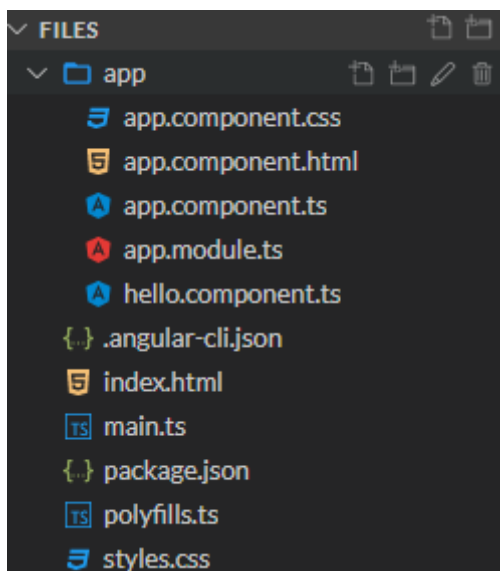


Рисунок 1 – Структура проекта

Вся работа будет проходить в файлах, которые находятся в папке «app».

Теперь можно заняться структурированием HTML. Здесь будут подключены классы Bootstrap (рис.1).

После вывода необходимых классов, файл html выглядит так (рис.2).

```
1 <div class="container">
2 <form #todoForm="ngForm" (submit)="todoSubmit(todoForm.value); todoForm.resetForm()">
3 <div class="form-group">
4 <h1 class="text-center">Список дел</h1>
5 <div class="input-group-prepend">
6 <input type="text" #todo class="form-control" placeholder=" " name="todo"
  ngModel>
7 <span class="input-group-text" (click)="addTodo(todo.value)">
8 <i class="material-icons">add</i></span>
9 </div>
10 </div>
11 <div class="data">
12 <ul class="list-unstyled">
13 <li [@moveInLeft] *ngFor="let todo of todoArray">{{todo}}
14 <i (click)="deleteItem(todo)" class="material-icons">delete</i></li>
15 </ul>
16 </div>
17 </form>
18 </div>
```

Рисунок 2 – файл html

Теперь, чтобы это выглядело все аккуратно и красиво, необходимо добавить стили CSS.

Импортируем шрифт с названием «Raleway» и сделаем подходящее оформление (рис.3-4).

```
1  @import url('https://fonts.googleapis.com/css?family=Raleway');
2
3  body{
4      padding: 0;
5      margin: 0;
6  }
7  form{
8      max-width: 30em;
9      margin: 4em auto;
10     position: relative;
11     background: #f4f4f4;
12     padding: 2em 3em;
13     overflow: hidden;
14 }
15 form h1{
16     font-family: "Raleway";
17     color: #F97300;
18 }
19 form input[type=text]::placeholder{
20     font-family: "Raleway";
21     color: #666;
22 }
23 form .data{
24     margin-top: 1em;
25 }
26 form .data li{
27     background: #fff;
28     border-left: 4px solid #F97300;
29     padding: 1em;
30     margin: 1em auto;
31     color: #666;
```

Рисунок 3 – файл CSS

```
31   color: #666;
32   font-family: "Raleway";
33 }
34 form .data li i{
35     float: right;
36     color: #888;
37     cursor: pointer;
38 }
39 form .input-group-text{
40     background: #F97300;
41     border-radius: 50%;
42     width: 5em;
43     height: 5em;
44     padding: 1em 23px;
45     color: #fff;
46     position: absolute;
47     right: 13px;
48     top: 68px;
49     cursor: pointer;
50 }
51 form .input-group-text i{
52     font-size: 2em;
53 }
54 form .form-control{
55     height: 3em;
56     font-family: "Raleway";
57 }
58 form .form-control:focus{
59     box-shadow: 0;
60 }
```

Рисунок 4 – файл css

После добавления структуры и стилей можно запустить приложение, чтобы посмотреть как оно выглядит (рис.5).

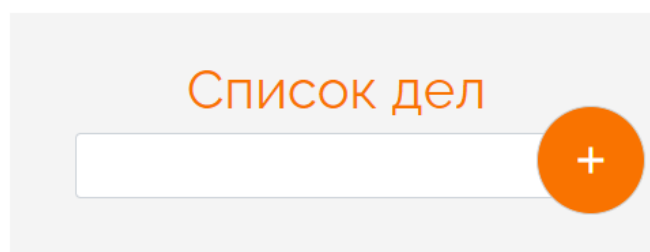


Рисунок 5 – Проверка в браузере

Выглядит все аккуратно, теперь можно начать добавлять механику добавления дел и их удаление. Для начала напишем связь между компонентами и немного анимации в файле «app.component.ts» (рис.6).

```
1  import { Component,trigger,animate,style,transition,keyframes } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css'],
7    animations:[
8      trigger("moveInLeft",[
9        transition("void=> *",[style({transform:"translateX(300px)"}),
10         animate(200,keyframes([
11           style({transform:"translateX(300px)"}),
12           style({transform:"translateX(0)"}
13         ]))]),
14       ])),
15     transition("*=>void",[style({transform:"translateX(0px)"}),
16     animate(100,keyframes([
17       style({transform:"translateX(0px)"}),
18       style({transform:"translateX(300px)"}
19     ]))])
20   ]))
21 ]
22 })
23
24 ]
25
26 ]
27 })
```

Рисунок 7 – Связь компонентов

Теперь добавим функции добавления, удаления и пуш уведомление при пустом вводе (рис.8).

```
44  ✓  addToDo(value) {
45  ✓      if (value !== "") {
46          this.todoArray.push(value);
47          todoForm.resetForm();
48  ✓      } else {
49          alert("Введите значение!");
50      }
51  }
52
53  ✓  deleteItem(todo) {
54  ✓      for (let i = 0; i <= this.todoArray.length; i++) {
55  ✓          if (todo == this.todoArray[i]) {
56              this.todoArray.splice(i, 1);
57          }
58      }
59  }
60
61  ✓  todoSubmit(value: any) {
62  ✓      if (value !== " ") {
63          this.todoArray.push(value.todo);
64  ✓      } else {
65          alert("Введите значение!");
66      }
67  }
68  }
```

Рисунок 8 - Функции

После добавления функций можно провести тестирование в браузере, заходим на локальный адрес и добавляем необходимые дела, которые удаляются, при нажатии значка корзины (рис.9).

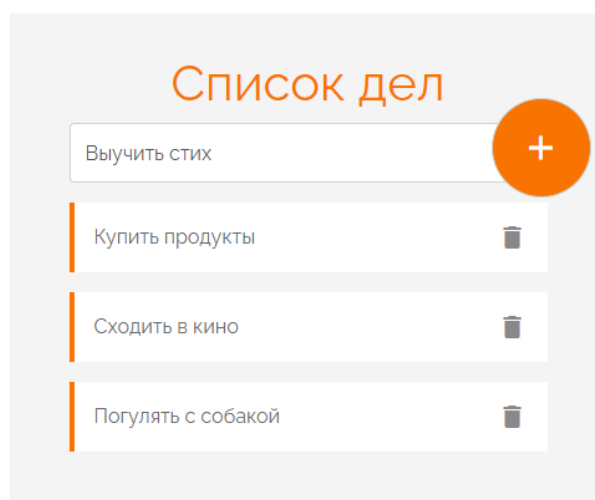
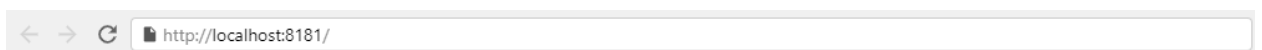


Рисунок 9 – Готовое приложение

В данной работе была рассмотрена возможность создания web-приложения списка дел, которые можно добавлять и удалять. При работе над реализацией использовались инструменты Angular и Node.js.

Библиографический список

1. Щербина Д.В., Богданова Н.Ю. Создание мобильного приложения для сети автозаправочных станций // Молодой исследователь Дона. 2018. №1(10). С. 80-87.
2. Алешина Т.А., Белаш В.Ю. Актуальные возможности фреймворка angular/angularjs для web-разработок // Научные труды Калужского государственного университета имени К.Э. Циолковского. 2019. №3. С. 507-510.
3. Оглулян А.К. Overview of the angular framework: pros and cons // Теория и практика модернизации научной деятельности в условиях цифровизации. 2020. №4(18). С. 33-37.
4. Мархакшинов А.Л., Иванов А.С. Проект для изучения основ фреймворка angular // Информационные системы и технологии в образовании, науке и бизнесе. 2020. №1(27). С. 26-29.