

## Создание проекта с использованием Angular

*Вавилов Егор Дмитриевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье описана возможность создания приложения с web-интерфейсом. При создании проекта использовался фреймворк Angular и инструмент написания код Visual Studio Code. Готовый проект выводит приветствие, после ввода имени.

**Ключевые слова:** Angular, Приложение, Разработка

## Creating a project using Angular

*Vavilov Yegor Dmitrievich*

*Sholom-Aleichem Priamursky State University*

*student*

### **Abstract**

This article will describe the basic architecture of angular applications along with using Visual Studio Code. The project will also be created and finalized. The finished project will display a greeting after entering the name.

**Keywords:** Angular, Application, Development

На сегодняшний день Angular один из самых быстрых и мощных фреймворков. Он используется для создания одностраничных web-приложений. Так же позволяет контролировать масштабируемость. Позволяет выполнять огромные требования к данным, создавать модели данных RxJS.

Цель исследования – разработать проект с web – интерфейсом, где при вводе имени, выводится ответ с приветствием.

Исследованиями в данной теме занимались следующие авторы. А.К.Оглукян ознакомила читателей с данным фреймворком, рассказав что это такое, как он работает и как часто происходят обновления [1]. Т.А.Алешина, В.Ю.Белаш рассказала про многие актуальные возможности данного фреймворка [2]. А.Л.Мархакшинов, А.С.Иванов рассмотрели проект, который может быть использован для знакомства обучающихся с популярным фреймворком разработки веб-приложений Angular [3]. Д.В.Щербина, Н.Ю.Богданова представили проект создания мобильного приложения, функционал которого позволит находить ближайшую автозаправочную станцию к местоположению мобильного устройства [4].

Для начала скачиваем Visual Studio Code с официального сайта и устанавливаем. После установки редактора, чтобы открыть приложение в редакторе кода, откроем файл из опции: “Файл - Открыть папку”.

Теперь приложение содержит множество файлов и папок, которые создаются angular CLI, который необходим для приложения. Папка “node\_modules” будет содержать все необходимые пакеты зависимостей, папка “e2e” используется для тестирования E2E. Также файл “package.json” содержит весь список пакетов зависимостей. Необходимо сосредоточиться в основном на папке “src”, поскольку пишем собственный код, внутри этого каталога. Ниже приведены файлы и структуры папок, созданные внутри папки “src” (Рис.1).

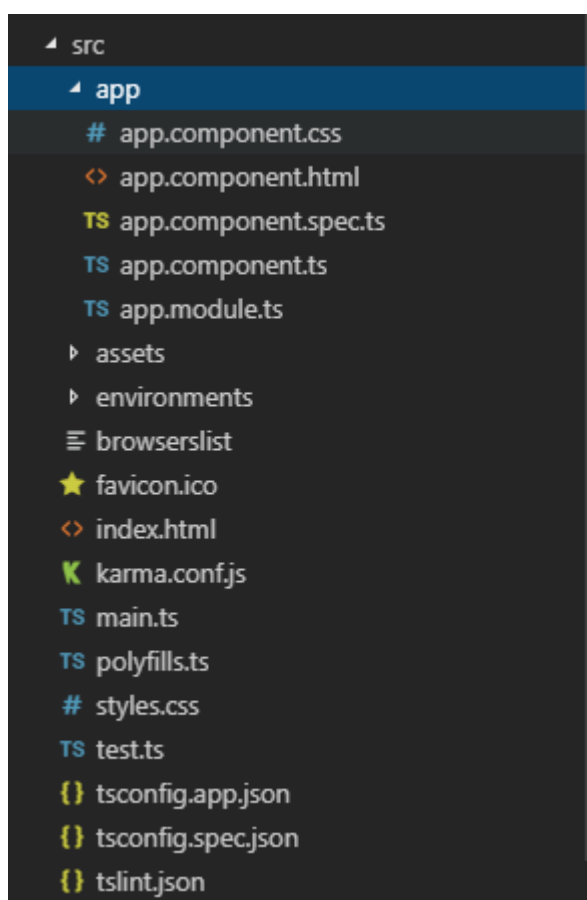


Рисунок 1 – Структура папок

Папка “app”: эта папка содержит несколько из следующих файлов:

- app.component.css: этот файл может содержать детали стиля CSS для конкретного приложения.
- app.component.ts: это файл машинописного текста (с расширением .ts). Этот файл содержит определение компонента приложения. Этот файл используется для динамической привязки данных к html-файлу компонента приложения (app.componet.html).
- app.componet.html: этот файл содержит компоненты HTML приложения.

- `app.module.ts`: этот файл содержит все необходимые модули, импортированные для приложения.

Можем запустить приложение из терминала Visual Studio Code. Выберите “Терминал - Новый терминал” на панели инструментов редактора и выполните команду “`ng serve`” (Рис.2). Любые изменения в файлах приложения будут напрямую отражены в браузере, если на терминале запущена команда “`serve`”.

```
D:\AngularApps\my-first-app>ng serve
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
0% compiling

Hash: af17e41cd5ebcef4951f
Time: 39389ms
chunk {main} main.js, main.js.map (main) 9.77 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 223 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 16.3 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.43 MB [initial] [rendered]
i [wdm]: Compiled successfully.
i [wdm]: Compiling...
```

Рисунок 2 – Запуск проекта

Теперь изменим содержимое “`app.component.html`” следующим содержимым. Здесь даем пользователю возможность ввести свое имя со стандартным тегом “`html <input>`”, и отображаем введенное имя в следующей строке. “[`(NgModel)`” компонент здесь используется для связывания переменного имени, оно динамически отображает введенное значение в следующей строке (рис.2).

```
1 | <div style="text-align:center">
2 |   Введите ваше имя: <input type="text" [(ngModel)]:
3 |   <h1>
4 |     Привет, {{ name }}!
5 |   </h1>
6 | </div>
```

Рисунок 3 – HTML - компонент

Теперь изменим файл “`app.component.ts`” следующим образом(рис.4).

```
1 | import { Component } from '@angular/core';
2 |
3 | @Component({
4 |   selector: 'app-root',
5 |   templateUrl: './app.component.html',
6 |   styleUrls: ['./app.component.css']
7 | })
8 | export class AppComponent {
9 |   name = 'Egor';
10 | }
11 |
```

Рисунок 4 – Изменение файла компонента

Если попытаться запустить приложение в браузере, то можно увидеть пустую страницу. Это связано с тем, что идет попытка привязать элемент ввода, для чего нужно импортировать “FormsModule” из библиотеки “@angular / forms” внутри файла “app.module.ts”.

Изменим “app.module.ts” следующим содержанием. Здесь добавляем импорт в строку 3 и регистрируем этот импорт внутри “@NgModule” в разделе импорта, как показано в приведенном ниже коде (Рис.5).

```
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3  import { FormsModule } from '@angular/forms';
4
5  import { AppComponent } from './app.component';
6
7  @NgModule({
8    declarations: [
9      AppComponent
10   ],
11   imports: [
12     BrowserModule,
13     FormsModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
```

Рисунок 5 – Изменение модуля

Теперь сохраняем файлы и проверяем в браузере (рис.6).

Введите ваше имя:

**Привет, Егор!**

Рисунок 6 - Проверка

В данной статье была рассмотрена работа с Visual Studio и изменение проекта Angular, а так же создание собственного проекта.

### Библиографический список

1. Оглуляк 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.
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. 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.
  4. Щербина Д.В., Богданова Н.Ю. Создание мобильного приложения для сети автозаправочных станций // Молодой исследователь Дона. 2018. №1(10). С. 80-87.