Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Акманов Р.Р. Использование Angular в проектах ASP. NET Core // Академия педагогических идей «Новация». – 2018. – №4 (апрель). – АРТ 82-эл. – 0,2 п. л. – URL: http://akademnova.ru/page/875548

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004

Акманов Рамиль Рафикович
студент 2го курса магистратуры
ФГБОУ ВО «КНИТУ»
г.Казань, Российская Федерация
email: rami_a@mail.ru

ИСПОЛЬЗОВАНИЕ ANGULAR В ПРОЕКТАХ ASP.NET CORE

Аннотация: В статье рассматривается практическое применение клиентского фреймворка Angular в проектах ASP.NET Core, используя языки программирования TypeScript и C#.

Ключевые слова: фреймворк, клиентская часть, серверная часть, теги, маршрутизация, паттерн наблюдатель, массив, метод, класс, цикл.

Akmanov Ramil Rafikovich
second year student of the magistracy
FGBOU VO “KNITU”
Kazan city, Russian Federation

USING ANGULAR IN ASP.NET CORE PROJECTS

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Abstract: The article shows practical using client framework Angular in ASP.NET Core projects, with programming languages TypeScript and C#.
Keywords: framework, front-end, back-end, tags, routing, pattern observer, array, method, class, loop.

В современное время веб-сайтами пользуется огромное количество пользователей, что определяет необходимость создание приложений рассчитанных на высокую нагрузку. Классические веб технологии .NET, рассчитаны на то, что HTML страница генерируется на стороне сервера, а затем передается клиенту, это увеличивает нагрузку на сервер, хорошим способом снять нагрузку с сервера и передать ответственность на клиентскую часть является использование Javascript фреймворка Angular.

Angular – клиентский фреймворк Javascript, который представляет возможность создания одностраничных представлений, генерируемых на стороне клиента. Новые версии Angular используют Typescript, как язык описания логики, далее Typescript интерпретируется в Javascript. Далее мы попробуем разобраться, как использовать Angular в проектах ASP.NET Core.

Преждем чем мы начнем, нам необходимо установить .NET Core, Node.js и Visual Studio. Создадим проект «Приложение ASP.NET Core» в Visual Studio, используя пустой шаблон.

Настройку проекта мы начнем с файла csproj нашего приложения, в нем в секцию PropertyGroup необходимо добавить тег TypeScriptCompileBlocked со значением true, что позволит не компилировать код, написанный на Typescript, т.к. для его компиляции мы будем использовать Node.js, если не отключить, то .NET Core будет пытаться скомпилировать TypeScript, что вызовет ошибки компиляции. Так же в секцию ItemGroup добавим 2 тега PackageReference, для добавления в

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

проект Microsoft.AspNetCore.Mvc версии 1.1.2 и Microsoft.AspNetCore.StaticFiles версии 1.1.1, MVC пакет позволит нам добавлять апи контроллеры, а StaticFile разрешит нам настраивать сервер, чтобы можно было использовать статические файлы. После сохранения csproj необходимо правой кнопкой мыши из контекстного меню проекта вызвать операцию восстановления пакетов.

Теперь изменит файл Statup.cs, в котором содержатся методы конфигурирования приложения в режиме исполнения. В методе ConfigureServices вызовем метод AddMvc, от сервиса передаваемого как аргумент функции. В методе Configure одним из параметров является IApplicationBuilder, от него асинхронно вызовем метод next, передавая управление далее в контейнере обработки запроса, а так же используем методы UseMvcWithDefaultRoute, UseDefaultFiles и UseStaticFiles, чтобы использовать маршрутизацию по умолчанию, а так же разрешить использование статических файлов.

Переходим к настройке клиентской части. Откроем командную строку и установим Angular-CLI, введя команду npm install @angular/cli –global. Создадим шаблон проекта Angular с помощью команды ng new {HelloApp} –skip-install, файлы добавятся в текущую директорию.

В проект необходимо добавить эти файлы в отдельную папку, затем откроем файл app.module.ts и добавим следующие строки:

import {FormsModule} from ‘@angular/forms’

import {HttpModule} from ‘@angular/http’

так же их необходимо добавить в секцию NgModule в список imports.

Теперь откроем файл .angular-cli.json и добавим в секцию apps новое свойство outDir, со значением wwwroot, так мы установим выходную

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

директорию, из которой ASP.NET Core будет использовать статические файлы.

Чтобы приложение выполняло, какую то логику нужно добавить вызов серверных API из клиентской части, для этого создадим в файле app.component.ts класс AppComponent implements OnInit, с пустым конструктором принимающим _httpService: Http, в теле класса определим свойство apiValues представляющее собой массив строк, для его заполнения создадим метод ngOnInit, в котором будем инициализировать массив apiValues актуальными значениями, для этого напишем в теле метода:

this._httpService.get('/api/values').subscribe(values => {this.apiValues = values.json() as string[];});

Теперь клиентская часть умеет получать данные с серверной части, однако мы еще не добавили вывод этих данных в представление. Откроем файл app.component.html и добавим следующую разметку:

<h1> Данные полученные с сервера </h1>

<li *ngFor="значение">{{value}}

С помощью конструкции *ngFor мы будем проходить список apiValues и выводить содержащееся в нем значения.

Для того чтобы значения были, изменим метод с атрибутом HttpGet в контроллере, объявим массив строк и заполним его значениями, вернем полученный массив.

Перейдем в корень нашего приложения в проводнике и запустим из командной строки npm install, чтобы создать все необходимые зависимости, затем построим проект с помощью ng build, и запустим его с помощью dotnet

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

run. Откроем браузер по созданному адресу, который отобразится в командной строке, в моем случае это localhost:5000.

На страничке отобразится список значений и заголовок «Данные полученные с сервера».

Для того, чтобы процесс изменения был проще и рекомпиляция происходила автоматически внесем некоторые правки в проект, для начала настроим прокси вызов для наших вызов методов API, для этого создадим новый файл и назовем его proxy.config.json, и вставим туда следующий код:

```
{
    “/api”: {
        “target”: http://localhost:5000,
        “secure”: false
    }
}
```

Теперь в файле .csproj добавим новую секцию ItemGroup, с тегом DotNetCliToolReference Include=”Microsoft.DotNet.Watcher.Tools” Version=”1.0.0”. Тем самым добавляя механизм событий.

Запустим наше приложение с помощью dotnet watch run, откроем другую командную строку и запустим часть Angular командой

Ng serve –proxy-config proxy.config.json

Откроем в браузере нашу страницу по ссылке определенной в командном окне. Попробуйте изменить файл app.component.html и сохранить файл. Можно увидеть, что без перезапуска приложения значения на странице изменяются автоматически.

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

В данном примере мы работали с локальным сервером, однако при желании выложить на удаленный сервер, можно воспользоваться встроенным решением Microsoft Azure, но для этого необходимо иметь аккаунт Azure, настройка которого не входит в рамки этой статьи, однако если он у вас уже есть, то публикация происходит за несколько шагов. Для этого нужно изменить файл .csproj

```
<Target Name="Сборка Angular"
Condition="'$(Configuration)'=='Release'" BeforeTargets="Build">
  <Message Text="* * * * * * Сборка приложения Angular * * * * * *"
Importance="high" />
  <Exec Command="ng build -prod -aot" />
</Target>
```

Так же необходимо изменить версию проекта с Debug, т.е. с отладки, на Release, т.е. на релизную версию. Вверху вы найдете иконку Publish, кликнув на нее проект, соберется и готов к публикации. Если аккаунт Azure уже прикреплен к вашей Visual Studio, то ваш проект автоматически опубликуется на Azure.

Список использованной литературы:

1. Документация Node.js, URL: nodejs.org/en/docs
2. Документация Angular, URL: angular.io/docs
3. Документация Azure, URL: docs.microsoft.com/ru-ru/azure/
4. Руководство по ASP.NET Core 2.0, URL: metanit.com/sharp/aspnet5/

Дата поступления в редакцию: 10.04.2018 г.
Опубликовано: 11.04.2018 г.