Angular cli in een MVC5 applicatie

We gaan een visual studio 2017 solution maken met daarin twee projecten. Een MVC applicatie en een lege asp.net webapplicatie, beide zullen het .Net frawework gebruiken.

# De MVC5 applicatie

We voegen een project toe aan de solution. Rechterklik, add 🡪New Project. In categorie Visual C# kiezen we voor Web, dan nemen we de ASP.NET Web Application (.NET Framework) template (.NET framework 4.6.1.) We geven hier de naam MVC5. Enter brengt ons naar stap 2.

In stap 2 kiezen we de MVC template, dat is alles voor ons voorbeeld maar je kan hier Web API aanvinken, de authenticatie wijzigen, …

# De angular applicatie

## Laat ons een empty asp.net project maken om daarin onze angular te maken met de cli.

Rechterklik op de solution, Add 🡪 New Project 🡪C#, Web, Asp .net Web Application. Geef hier de naam NgCli in, kies in stap 2 voor Empty.

## Angular maken

We gaan de angular app maken met de cli in een gewone windows cmd line, niet de command prompt van visual studio die de system path settings niet respecteert en we hebben node nodig.

Klik rechts op de solution -> Open folder in file explorer, in file explorer klik je in het adresbalk, typ cmd en enter om een command prompt in die map te starten.

We zijn dus in een command prompt, in de map waar de solution file staat.

Node, npm en angular cli moeten geïnstalleerd zijn.

Je kan node [hier](https://nodejs.org/en/download/) downloaden en installeren, gebruik de msi. NPM wordt mee geïnstalleerd. Angular cli installeer je met volgende commando:  
npm install -g @angular/cli@latest  
Verifieer eventueel de installatie met ng version.

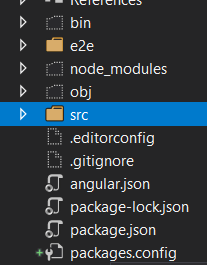
### App maken

Laat ons de angular app aanmaken. De bedoeling is dat de app files gecreëerd worden in de folder NgCli, dit is de naam van het zopas aangemaakte asp.net project. En die map staat dus één niveau lager dan waar we nu zitten.

We typen dus:  
ng new ngCli --routing --skip-git --directory ngCli  
We hebben waarschijnlijk wel routing nodig, git is niet nodig want het is de bedoeling van de hele solution in git te zetten en --directory laat de cli de app in ons reeds bestaande projectmap genereren.

### Include in project

Laat ons naar Visual Studio teerugkeren en alle gegenereerde files in het project includen behalve node\_modules, deze files kunnen worden gerestored adhv npm install.



### NgCli testen

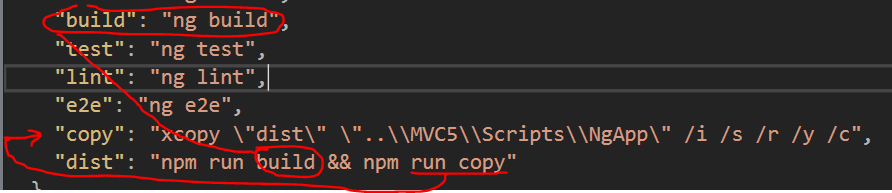
Laat ons eens zien of ons cli app functionneert. Typ ng serve aan de prompt in de NgCli map. Als het goed zit is de app nu te zien op loacalhost:4200. Surf ernaar en indien alles werkt kunnen we de browser afsluiten en de server ook. Dit laatste doen we met ctrl-c.

### Build Cli output in onze MVC5 app (met npm aan de prompt)

De bedoeling is dat de ouput van de build van onze cli app in de MVC5 app terechtkomt. En dit liefst elke keer we builden. Ofwel laten we Visual Studio daarvoor zorgen ofwel de cli. Hier gaan we de cli dat laten doen. Het gaat een beetje sneller en we moeten onze mvcapp niet telkens opnieuw builden. Enkel refreshen (ctrl-F5).

We zullen er dus voor zorgen dat de app gebuild wordt en dat de inhoud van de op dat moment gegenereerde dist folder gekopieerd wordt in de MVC5 map, onder scripts bijv.

We voegen 2 script taken aan package.json. build zit er al in, we moeten nog copy maken, en we bundelen die twee in één opdracht, dist. De copy opdracht doet een xcopy van alles wat er in de dist map zit naar de Scripts map in de MVC5 projectfolder onder een map NgApp.

  
Die aanpassing aan package.json kunnen we met om het even welken editor doen, met visual studio dus ook, wat wel handig is.

En dan kunnen we npm run dist gaan uitvoeren.

### Build Cli output in onze MVC5 app (vanuit visual studio)

Het is mogelijk om een zogenaamde “pre-build” event te beschrijven in visual studio. Het is dus een alternatief voor punt 2.2.4. We gaan van die mogelijkheid gebruik maken om de cli te “builden” en te kopiëren naar de MVC applicatie voor we de mvc applicatie zelf builden en starten.

Open de project properties van het MVC project. Ga naar tab Build Events. We gaan commandos invoeren in de pre-build event. Voer volgende code in:

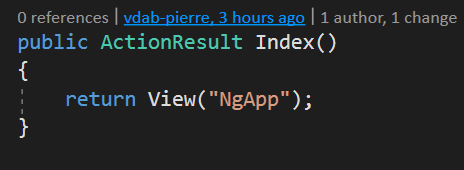
echo "cd $(SolutionDir)NgCli" &&^  
cd "$(SolutionDir)NgCli" &&^  
echo "building NgCli" &&^  
ng build &&^  
echo 'copy files' &&^  
xcopy /I /E /Y "$(SolutionDir)NgCli\dist" "$(ProjectDir)Scripts\NgApp"

## Gebruik maken van de cli app In MVC5

Rest ons een action te voorzien om een “angular” view op te roepen.

In visual studio openen we HomeController. We wijzigen de Index action zodanig dat die een view “NgApp” genaamd opent. Die view zullen we ook moeten maken.

De controller Index action. Wijzig Index door de string NgApp mee te geven aan View().



En de NgApp view.   
Rechterklik op de action method 🡪 Add View, type NgApp, vink Use a layoutpage uit. Er wordt een NgApp.cshtml aangemaakt. Hierna volgt de html.

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<base href="/" />

<title>NgApp</title>

</head>

<body>

<app-root>Loading...</app-root>

<script type="text/javascript" src="~/Scripts/NgApp/ngCli/runtime.js"></script>

<script type="text/javascript" src="~/Scripts/NgApp/ngCli/polyfills.js"></script>

<script type="text/javascript" src="~/Scripts/NgApp/ngCli/styles.js"></script>

<script type="text/javascript" src="~/Scripts/NgApp/ngCli/vendor.js"></script>

<script type="text/javascript" src="~/Scripts/NgApp/ngCli/main.js"></script>

</body>

</html>

Vergeet base href niet!

De scripttags verwjzen naar zopas in Scripts/NgApp gekopieerde dist folder van onze cli app.

Telkens je de cli app wijzigt, moet die gebuild worden en de dist folder gekopieerd naar de mvc app en dat doen we met npm run dist.

De MVC5 applicatie hoeft niet opnieuw gecompileerd te worden, gewoon de browser vernieuwen, let op de cache (ctrl-f5)

Gebruik je de andere optie (cli builden en kopiëren samen builden van de mvc app), vergeet dan niet om een ‘clean’ build te doen indien niks gewijzigd is geweest in de MVC app.