

Atte Aspiola

ASP.NET Web API 2:ta hyödyntävä Angular 4 -sovellus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

13.9.2017

Tekijä(t) Otsikko	Atte Aspiola ASP.NET Web API 2:ta hyödyntävä Angular 4 -sovellus
Sivumäärä Aika	40 sivua + 2 liitettä 13.9.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaajat	Lehtori Simo Silander Osastopäällikkö Mikko Ristikangas
<p>Insinööriyön tavoitteena oli päivittää VäestöWeb-niminen sovellus käyttämään uusimpia web-ohjelmoinnissa käytettäviä tekniikoita: Angular 4:ää ja ASP.NET Web API 2:ta. Sovelluksen tarkoitus on tarjota kunnille dataa ikäjakauman, väestönmuutoksen sekä kuntaan ja kunnasta muuttaneiden muodossa. Sovellus myytiin ja otettiin käyttöön kahdeksassa kunnassa.</p> <p>Sovellus jakautuu kahteen osaan: käyttöliittymään ja back-endiin. Käyttöliittymä toteutettiin Angular-sovelluskehysellä ja ohjelmoitiin käyttäen TypeScript-, HTML- ja CSS-ohjelmointikieliä. Sovelluksen back-end ohjelmoitiin C#-ohjelmointikielillä ASP.NET Framework -sovelluskehystä käyttäen.</p> <p>Kirjallinen osuus jakautuu kahteen osaan: Angular 4- ja Web API -osuuksiin. Angular 4 -osuudessa keskitytään Angularin ominaisuuksiin ja niiden esittelyyn. Web API -osuudessa esitellään sovelluksessa käytettyjä ohjelmointiratkaisuja.</p>	
Avainsanat	Angular 4, C#, Web API, ASP.NET, TypeScript

Author(s) Title	Atte Aspiola Angular 4 Application with ASP.NET Web API 2
Number of Pages Date	40 pages + 2 appendices September 13, 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructors	Simo Silander, Senior Lecturer Mikko Ristikangas, Head of Department
<p>The goal of this bachelor's thesis was to update an application called VäestöWeb using the latest technologies, Angular 4 and ASP.NET Web API 2. The application offers data to municipalities in form of age distribution, population change and about people who have moved into or out of the municipality. The application was sold to and introduced to eight municipalities.</p> <p>The application is divided into two parts: front-end and back-end. The front-end of the application was programmed using Angular framework and TypeScript-, HTML- and CSS programming languages. The back-end of the application was programmed using C# programming language and ASP.NET Framework.</p> <p>The written part of the bachelor's thesis is divided into two parts: Angular 4 and Web API. Angular 4 part focuses on the features of the framework using examples. The Web API part is all about the solutions used in this particular application.</p>	
Keywords	Angular 4, C#, Web API, ASP.NET, TypeScript

Sisällys

Lyhenteet

1	Johdanto	1
2	Sovelluksen käyttötarkoitus	1
3	Tärkeimmät käytetyt tekniikat ja sovelluksen arkkitehtuuri	3
3.1	Angular 4	3
3.2	ASP.NET	3
3.3	SpatialWeb	3
3.4	Sovelluksen arkkitehtuuri	3
4	Sovelluksen käyttöliittymä	3
4.1	Komponentit	4
4.2	Näkymät	4
4.2.1	Ikäjakauma-näkymä	4
4.2.2	Kuntaan muuttaneet -näkyvä	4
4.2.3	Väestönmuutos-näkymä	4
4.3	Palvelut	4
5	Web API	4
5.1	Ohjaimet	4
5.2	DTO:t	4
5.3	Apuluokat	4
6	Yhteenveto	4
	Lähteet	6

Liitteet

Liite 1. VäestöWeb dokumentaatio

Liite 2. Koodiesimerkki - app.module.ts

Lyhenteet

API	Application programming interface. Sovelluksen osa, jonka tehtävänä on vastaanottaa HTTP -pyyntöjä ja vastata niihin.
CRUD	Create, read, update, delete. Tiedonhallinnan perusfunktiot.
DTO	Data transfer object. Oliot, jonka avulla siirretään dataa komponenttien välillä.
HTTP	Hypertext transfer protocol. Asiakas-palvelin-protokolla, joka koostuu pyynnöistä ja vastauksista. Datan siirto internetissä perustuu HTTP:aan.
JSON	JavaScript object notation. Tiedostotyyppi, jonka avulla voidaan siirtää oliot tekstinä avain-arvo muodossa.
LINQ	Language integrated query. C# -ohjelmointikieleen sisäänrakennettu komponentti, jonka avulla voi tehdä tietokantakyselyjä.
SPA	Single-page application. Web-sovellus, jossa ei liikuta sivulta toiselle, vaan sivulla olevat komponentit ja niiden sisältö muuttuvat käyttäjän valintojen perusteella.
SQL	Structured query language. Ohjelmointikieli, joka on suunniteltu datan hallintaan relaatiotietokannoissa.

1 Johdanto

Tämän insinööriyön päämääränä oli päivittää Siton VäestöWeb-sovellus käyttämään uusia tekniikoita, Angular 4:ää ja ASP.NET Web API 2:ta. Vanhassa sovelluksessa käytettiin Visual Basic-, Javascript- ja HTML-ohjelmointikieliä, joten sovelluksen kehittäminen aloitettiin ns. puhtaalta pöydältä. Projekti toteutettiin kolmen hengen ryhmässä, jossa tehtäväni oli Web API:n toteutus ja osittain Angular-projektin toteutus. Sovellus oli ryhmän ensimmäinen edellä mainituilla tekniikoilla toteutettu sovellus.

Raportin Angular-osuudessa keskitytään ikäjakauma-näkymään ja sen eri komponenttien välisiin vuorovaikutuksiin sekä Angularin ominaisuuksien esittelyyn. Web API -osuudessa esitellään projektissa käytettyjä ohjelmointiratkaisuja. En osallistunut sovelluksen ulkoasun toteutukseen, joten tässä raportissa ei käsitellä sovelluksen tyyliin tai ulkoasun rakenteeseen liittyviä seikkoja.

Sovellus valmistui ja toimitettiin asiakkaille ajallaan huolimatta vaikeuksista sovelluksen ydintekniikoihin liittyen. Esimerkiksi kehys, jonka avulla tietokantakyselyt piti suorittaa, jouduttiin korvaamaan toisella ratkaisulla, mikä aiheutti useamman henkilötyöviikon työn menettämisen.

Sovelluksen lisäksi insinööriyön tuloksena syntyi liitteen 1 dokumentaatio, joka sisältää sovelluksen asennusohjeen, sekä tietoa admin-paneelin asetuksista ja sovelluksen eri raporteista.

2 Sovelluksen käyttötarkoitus

VäestöWeb-sovellus on kunnille suunnattu palvelu, jonka avulla kunta voi suunnitella esimerkiksi, mille alueelle olisi suurin tarve rakentaa uusi päiväkotikiinteistö tai miten uudet koululaiset jaetaan kouluihin. Sovelluksessa on kolme näkymää: ikäjakauma, kuntaan muutaneet ja väestönmuutos.

Ikäjakauma-näkymässä käyttäjä voi luoda ikäjakaumaraportin kartalta valitsemaltaan alueelta tai listalta valitsemiltaan kaupunginosilta. Ikäjakauma tulostetaan ikävuosittain tai syntymävuosittain äidinkielen ja haluttaessa myös sukupuolen perusteella taulukon 1

mukaisesti. Raportissa olevat ihmiset voidaan piirtää kartalle asuinrakennuksen perusteella.

Taulukko 1. Esimerkki ikäjakaumasta ikävuosittain, äidinkielen perusteella (luvut esimerkkilukuja)

Ikäryhmä	Suomi	Ruotsi	Muut	Yhteensä
14 vuotiaat	132	72	42	246
15 vuotiaat	168	51	62	281
Yhteensä	300	123	104	527

Kuntaan muuttaneet -näkyvässä voidaan luoda raportti, josta nähdään valittujen päivämäärien välillä kaupunginosittain kuntaan muuttaneiden ihmisten lukumäärä. Käyttäjä voi halutessaan myös tulostaa samankaltaisen raportin kunnasta muuttaneista. Esimerkki tulostetusta raportista on taulukossa 2. Raportista voidaan muodostaa myös graafinen kuva kartalle, jossa kaupunginosat, joihin muutto on ollut vilkkaampaa, värjätään tummemmalla kuin kaupunginosat, joihin on muuttanut vähemmän ihmisiä. Näitä kuvia kutsutaan jatkossa *teemakartoiksi*.

Taulukko 2. Esimerkki kuntaan muuttaneet-näkymän raportista kaupunginosittain (luvut esimerkkilukuja)

Kaupunginosa	Kuntaan muuttaneet
Lauttasaari	64
Kulosaari	26
Yhteensä	90

Väestönmuutos-näkyvässä tulostettavasta raportista nähdään vuosittain ja kaupunginosittain kuntaan muuttaneet, kunnasta muuttaneet, kuolleet ja syntyneet ihmiset. Esimerkki tulosteesta on taulukossa 3. Myös tässä näkyvässä voidaan muodostaa teemakartta.

Taulukko 3. Esimerkki väestönmuutos-näkymän raportista. Tarkasteltavaksi kaupunginosaksi on valittu Kulosaari ja tiedot on ryhmitelty vuosittain (luvut esimerkkilukuja).

Kulosaari	2010	2011	2012	2013
Kuntaan muuttaneet	56	98	147	223
Kunnasta muuttaneet	14	9	22	6
Kuolleet	2	6	15	9
Syntyneet	10	11	8	16

Jokaisesta näkymästä on mahdollista viedä tulokset Exceeliin ja muuttaa tulokset tulosystävälliseen muotoon.

3 Tärkeimmät käytetyt tekniikat ja sovelluksen arkkitehtuuri

Luku 3 poistettu luottamuksellisena toimeksiantajan pyynnöstä.

3.1 Angular 4

3.2 ASP.NET

3.3 SpatialWeb

3.4 Sovelluksen arkkitehtuuri

4 Sovelluksen käyttöliittymä

Luku 4 poistettu luottamuksellisena toimeksiantajan pyynnöstä.

4.1 Komponentit

4.2 Näkymät

4.2.1 Ikäjakauma-näkymä

4.2.2 Kuntaan muuttaneet -näkyvä

4.2.3 Väestönmuutos-näkyvä

4.3 Palvelut

5 Web API

Luku 5 poistettu luottamuksellisena toimeksiantajan pyynnöstä.

5.1 Ohjaimet

5.2 DTO:t

5.3 Apuluokat

6 Yhteenveto

Insinööriyön Angular-osuudessa keskityttiin Angularin eri komponenttien välisiin vuorovaikutuksiin ja Angularin ominaisuuksien esittelyyn. Web API -osuudessa esiteltiin projektissa käytettyjä ohjelmointiratkaisuja.

Sovelluksen kehittämiseen kului aikaa noin kolme kuukautta, mikä vastasi Siton odotuksia. Suurimmat viivästykset sovelluskehityksessä johtuivat siitä, että tehtävän määrittely oli vajavainen. Olen todella tyytyväinen siihen, että saimme sovelluksen valmiiksi ja asennettua asiakkaille ajoissa, huolimatta kohtaamistamme vaikeuksista.

Projektissa jäi toteuttamatta ikäjakaumaan liittyvä yksityiskohtaisempi Excel-tiedosto. Muutoin onnistuimme toteuttamaan kaikki asiakkaiden toivomat uudet ominaisuudet, sekä joitain ominaisuuksia, joita ei oltu erikseen toivottu, mutta ne toteutettiin, koska niiden pois jättäminen olisi vienyt enemmän aikaa kuin niiden toteuttaminen.

Opin projektin aikana, kuinka tärkeää on saada asiakkaalta tarkka määrittely sovelluksen toiminnasta. Aika moni tehtävä kesti huomattavasti odotettua pidempään johtuen joko asiakkaan tai työnantajan puutteellisesta määrittelystä. Lisäksi opin, että on parempi kysyä kokeneemmilta työntekijöiltä apua kuin yrittää tuskailta tehtävän kanssa. Projektissa oli ongelma, jota yritin ratkaista useamman päivän ajan. Lopulta kysyin apua ongelmaan, ja se ratkesi viidessä minuutissa.

Kehityin projektin aikana ohjelmoijana ja ennen kaikkea sosiaalisessa kanssakäymisessä. En ollut ennen projektia ohjelmoinut C#-kielellä, joten on selvää, että opin ainakin jotain uutta. VäestöWeb oli ryhmämme ensimmäinen tällä kokoonpanolla kehitetty sovellus. Löysin oman paikkani ryhmässä ja uskon, että seuraava projekti tulee onnistumaan vielä paremmin, koska olemme oppineet tuntemaan toisemme paremmin.

Insinööri työ oli kokonaisuudessaan positiivinen ja opettavainen kokemus.

Lähteet

- 1 Fain, Y., Moiseev A. 2017. Angular 2 Development with TypeScript. Shelter Island: Manning Publications Co.
- 2 Angular 2 tutorial – Services. 2017. Verkkodokumentti. Google. <<https://angular.io/tutorial/toh-pt4>>. Luettu 11.7.2017.
- 3 Angular 2 documentation – NgModules. 2017. Verkkodokumentti. Google. <<https://angular.io/guide/ngmodule>>. Luettu 11.7.2017.
- 4 Ngx-translate core documentation. 2017. Verkkodokumentti. NGX-Translate. <<https://github.com/ngx-translate/core>>. Luettu 11.7.2017.
- 5 Angular 2 documentation – Routing & Navigation. 2017. Verkkodokumentti. Google. <<https://angular.io/guide/router>>. Luettu 12.7.2017.
- 6 Pascal Precht. 2016. Two-way data binding in Angular. Verkkodokumentti. Thoughttram. <<https://blog.thoughttram.io/angular/2016/10/13/two-way-data-binding-in-angular-2.html#two-way-data-binding-in-a-nutshell>>. Luettu 12.7.2017.
- 7 Angular 2 documentation – User Input. 2017. Verkkodokumentti. Google. <<https://angular.io/guide/user-input>>. Luettu 12.7.2017.
- 8 Angular 2 documentation – NgIf. 2017. Verkkodokumentti. Google. <<https://angular.io/api/common/NgIf>>. Luettu 13.7.2017.
- 9 Angular 2 documentation – NgForOf. 2017. Verkkodokumentti. Google. <<https://angular.io/api/common/NgForOf>>. Luettu 13.7.2017.
- 10 Angular 2 documentation – HTTP. 2017. Verkkodokumentti. Google. <<https://angular.io/guide/http>>. Luettu 14.7.2017.
- 11 What is an SRID. 2016. Verkkodokumentti. ESRI. <<http://desktop.arcgis.com/en/arcmap/10.3/manage-data/using-sql-with-gdbs/what-is-an-srid.htm>>. Luettu 14.7.2017.
- 12 Entity Framework Provider Support for Spatial Types. 2016. Verkkodokumentti. Microsoft. <<https://msdn.microsoft.com/en-us/data/dn194325>>. Luettu 15.7.2017.
- 13 How and Why to Use Parameterized Queries. 2008. Verkkodokumentti. Microsoft. <<https://blogs.msdn.microsoft.com/sqlphp/2008/09/30/how-and-why-to-use-parameterized-queries/>>. Luettu 14.7.2017.

Liitteet

Liite 1. VäestöWeb dokumentaatio

Liite 2. Koodiesimerkki - app.module.ts

Liitteet poistettu luottamuksellisena toimeksiantajan pyynnöstä.