B.Sc. í tölvunarfræði

Bílkaup

Rekstrarhandbók

Sendiráðið vefstofa ehf.

Hrókering:

Bríet Konráðsdóttir (briet15@ru.is)

Drífa Örvarsdóttir (drifa15@ru.is)

Fanney Þóra Vilhjálmsdóttir (fanneyv12@ru.is)

Ólöf Gyða Risten Svansdóttir (olof15@ru.is)

Leiðbeinandi: Sigurjón Ingi Garðarsson
Prófdómari: Gunnar Sigurðsson

SC-T-404-LOKA
Tölvunarfræðideild

Efnisyfirlit

1 Inngangur

Í þessari rekstrarhandbók mun vera farið yfir allt það hvernig hægt er að taka við forritinu, setja það upp á sinni vél og keyra. Einnig eru upplýsingar um hvernig áframhaldandi þróun á veflæga bílasölukerfinu, Bílkaup getur verið háttað.

2 Uppsetning þróunarumhverfa

Bílkaup byggist á tveimur þróunarumhverfum (e. development platform); .NET Core og Angular. .NET Core notast við forritunarmálið C# og er það umhverfi sem bakendi kerfisins er byggt á. Angular er skrifað í TypeScript og spannar framenda kerfisins. Til að halda áfram þróun kerfisins þurfa bæði þessara þróunarumhverfa að vera uppsett í vél.

Hér að neðan eru leiðbeiningar hvernig hægt er að setja upp þróunarumhverfin.

2.1 .NET Core

Hægt er að setja upp .NET Core með því að sækja uppsetningarpakka á Git síðu umhverfisins, sem finna má hér: https://github.com/dotnet/core/blob/master/release-notes/download-archives/2.0.3.md

2.2 Angular

Til þess að nálgast Angular uppsetningarpakkann þarf npm pakkatólið (e. *package manager*) að vera í vélinni. npm vinnur með Node.js og er hægt að sækja pakkana saman hér: https://www.npmjs.com/get-npm

Næsta skref væri að sækja Angular sjálft og setja upp í vélina. Það er hægt með því að opna terminal og setja inn skipunina

```
npm install @angular/cli --global
```

3 Gagnagrunnar

Kerfið notast við þrjá mismunandi gagnagrunna; SQL, Elasticsearch og Blob Storage. Hingað til hefur kerfið notast við MSSQL og Blob Storage á Microsoft Azure, og Elasticsearch þyrpingu (e. *cluster*) hjá Bonsai. Þegar tenging hefur fengist fyrir alla þrjá gagnagrunna, er hægt að setja inn tengi upplýsingar inn í skrá sem heitir `appsettings.json` sem má finna í Server/API.

Uppsetning á SQL gagnagrunninum hefur verið unnin code-first og fer öll stjórnun á gagnagrunninum í gegnum einingamódel (e. *entitiy model*) sem finna má í Server/Models/EntityModels, en þau virka sem spegill á gagnagrunninn. Hvert einingamódel er ein tafla í gagnagrunninum. Skráin sem stjórnar hvaða einingamódel verður að töflu má finna í Server/Repositories og kallast `ApplicationDbContext.cs`. Flestar breytingar á gagnagrunninum geta svo farið í gegn með þessum tveim skipunum í terminal á slóðinni Server/API:

```
dotnet ef migrations add NameOfMigration
dotnet ef database update
```

Við fyrstu skipun ættu allar töflur í gagnagrunninum að verða til.

3.1 Forkröfur fyrir aðgangsstýringu

Þar sem aðgangsstýringin notast við tvær aðgangstöflur sem þurfa að innihalda ákveðnar upplýsingar áður en notandi getur sótt um aðgang eða skráð sig inn. Til að setja þessi gögn inn er hægt að nota SQL tól sem tengist SQL gagnagrunninum. Við þróun kerfisins notast við forritið Microsoft SQL Server Management Studio. Þegar allar töflur eru komnar upp eins og lýst er að ofan, er hægt að setja inn eftirfarandi setningar til að fylla inn í gagnagrunninn.

```
INSERT INTO [dbo].[AspNetRoles] ([Id], [ConcurrencyStamp], [Name],
[NormalizedName]) VALUES (1, NULL, Admin, Admin);

INSERT INTO [dbo].[AspNetRoles] ([Id], [ConcurrencyStamp], [Name],
[NormalizedName]) VALUES (2, NULL, Carsale, Carsale);

INSERT INTO [dbo].[AspNetRoleClaims] ([ClaimType], [ClaimValue], [RoleId]) VALUES
(Admin, Admin, 1);

INSERT INTO [dbo].[AspNetRoleClaims] ([ClaimType], [ClaimValue], [RoleId]) VALUES
(Carsale, Carsale, 2);
```

4 Að keyra kerfið

4.1 Sækja Angular Node Modules

Angular byggist á að nota svokölluð „node modules“ og þeir eru allir skráð í framendanum. Hver vél sem keyrir kerfið þarf að hafa sín eigin „node modules“ og það er hægt að ná í þá með skipun sem kallað er á í Client möppunni.

```
npm install
```

Þetta þarf aðeins að gera í byrjun verkefnisins og þegar nýjum „node modules“ er bætt við verkefnið.

4.2 Að þýða og keyra kerfið

Þar sem kerfið er tvískipt (framendi og bakendi), þarf hvor hluti að vera keyrður á staðbundinni vél (e. *localhost*) og þarf því að notast við tvo terminal glugga.

4.2.1 Bakendi

Til að þýða og keyra kerfið þarf terminalinn að vera staðsettur í slóðinni Server/API. Til að þýða kerfið þarf að keyra eftirfarandi skipun.

```
dotnet build
```

Þegar búið er að þýða bakendann er hægt að keyra hann. Eftirfarandi skipun heldur vélinni gangandi og lætur hana þýða kerfið aftur ef breytingar verða á skrám í bakendanum. Vélin hlustar sjálfvirkt á `localhost:5000`.

```
dotnet watch run
```

4.2.2 Framendi

Til að þýða og keyra framendann þarf aðeins að keyra eina skipun í terminal sem er staðsettur í slóðinni Client. Eftirfarandi skipun virkar eins og fyrrum skipun í bakenda og þýðir framendann aftur ef breytingar verða í skrám hans. Vélin hlustar sjálfvirkt á `localhost:4200` og er hægt að opna í vafra til að sjá útlit síðunnar.

```
npm start
```

5 Að keyra prófin

Eins og uppsetning kerfisins gefur til kynna, þá þarf að keyra próf framenda og bakenda í sitthvoru lagi.

5.1 Bakendi

Próf bakendans eru keyrð úr annarri möppu en bakendinn sjálfur; úr möppunni Server/Tests. Prófin sjálf má finna í möppunni Server/Tests/Unit tests.

```
dotnet test
```

5.2 Framendi

Próf framenda fyrir hverja einingu (e. *component*) má finna í möppum sem geyma hverja einingu, en skráin sem geymir prófin endar á `*.component.spec.ts`. Til að keyra próf framendans þarf terminalinn að vera staðsettur á sama stað og framendinn er keyrður frá, eða í Client möppunni.

```
npm test
```

6 Framtíðarsýn

Til að Bílkaup myndi virka á sem bestan hátt væri ákjósanlegast að falast eftir öllum þeim upplýsingum sem í boði eru frá Samgöngustofu. Það sem er í notkun á þessari stundu eru prufu gögn.

6.1 Samningur við Samgöngustofu

Samgöngustofa er ábyrgðaraðili ökutækjaskrár sem annast miðlun á upplýsingum til vinnsluaðila. Bílkaup kerfið er hannað þannig að það sækir upplýsingar um bíla úr ökutækjaskrá. En til að fá þann aðgang þarf að ganga frá réttindamálum og gera samning um kaup á gögnunum. Sá samningur myndi þá hljóma upp á að bílasala borgaði fyrir hverja uppflettingu á bíl. Þar á meðal persónuupplýsingum. Með persónuupplýsingum er átt við upplýsingar sem hægt er að rekja til tiltekins einstaklings. Samgöngustofa þarf að samþykkja aðferðina sem notuð er til að miðla upplýsingunum.

6.2 „Mock“ gagnagrunnur

Samgöngustofa hefur veitt teyminu gagnagrunn sem er uppsettur eins og ökutækjaskrá, nema hann inniheldur ekki raunverulegar persónuupplýsingar. Þar sem gagnagrunnurinn kom í hendur teymisins seint, var ekki hægt að tengja hann við núverandi kerfi. Teymið sér þó að næsta skref í þróun kerfisins væri að nýta ökutækjaskrá til að sækja enn fleiri gögn til að auðvelda bílasalanum lífið. Við frekari þróun kerfisins er því fullkomið að nýta þennan „mock“ gagnagrunn áður en tenging við Samgöngustofu fæst og samningar eru gerðir. Hægt er að nálgast „mock“ gagnagrunninn með því að hafa samband við meðlimi Hrókeringar.