



Angularin ja Reactin erot front-end kehityksessä

Antti Vainikka

TkK-tutkielma
Huhtikuu 2017

TULEVAISUUDEN TEKNOLOGIOIDEN LAITOS
TURUN YLIOPISTO

TURUN YLIOPISTO
Tulevaisuuden teknologioiden laitos

VAINIKKA, ANTTI: Angularin ja Reactin erot verkkosovellusten kehityksessä

TkK-tutkielma, x-määrä s., x liites.

Tietotekniikka

Huhtikuu 2017

Angular ja React ovat molemmat laajalti käytössä olevia front-end-työkaluja, eli asiakaspuolen työkaluja, jotka ovat luotuja laajentamaan verkkosovellusten toiminnallisuuksia sekä helpottamaan niiden kehitystä. Nykyään harvemmin enää kirjoitetaan verkkosovelluksia tai edes yksinkertaisempia verkkosivuja ilman mitään kehikkoja tai kirjastoja, joten niistä on tullut teollisuusstandardeja.

Angular ja React ovat nykyisistä front-end-työkaluista kehittyneimpien joukossa. Angular on front-end-kehikko, jolle voidaan ohjelmoida käyttämällä Javascript-tai Typescript-ohjelmointikieltä. Typescriptillä kirjoitetut Angular-sovellukset käännetään Javascriptiksi ennen suoritusta. React on Javascript kirjasto, joten sitä kirjoitetaan enimmäkseen Javascriptiä käyttäen. DOM (document object model)-manipulaatio Reactilla tapahtuu käyttäen JSX Javascript laajennussyntaksia, jolla voidaan kirjoittaa HTML-tageja Javascriptin sekaan.

Asiasanat: Javascript, React, Angular, Front-end, verkkosovellukset, sovelluskehikset

Sisältö

1	Johdanto	1
2	Ohjelmointikäytännöt	3
3	Backendin kanssa kommunikointi	5
4	Datan sitominen ja DOM-manipulaatio	6
5	Suorituskyky ja kehittämisen vaikeudet	7

1 Johdanto

Nykyaikaisessa verkkosovelluskehityksessä on tavallisesti käytössä yksi tai useampia erilaisia kehyksiä tai kirjastoja, jotka auttavat kehittäjää tekemään sovelluksesta laajemman ja ominaisuuksiltaan monipuolisemman. Kehykset ja kirjastot tekevät usein vain tietyn osan sovelluksesta.

Angular ja React ovat tunnetuimpien joukossa front-end työkaluista. Ne ovat laajalti käytössä monissa verkkopalveluissa ympäri maailman. Front-end-työkaluilla tarkoitetaan kehyksiä tai kirjastoja, joilla luodaan näkymiä ja toiminnallisuuksia, jotka näkyvät käyttäjälle. Toinen verkkosovellusten puoli on back-end. Back-end-työkalut ovat palvelimen puolella käytettäviä työkaluja, kuten palvelinsovelluksia.

Angularia on käytetty muun muassa Wolfram Alpha laskentasovelluksessa[2], suoritinvalmistaja Intelin verkkosivuissa sekä uutisyhtiö NBC:n (National Broadcasting Company) sekä uutisyhtiö ABC Newsin (American Broadcast Company News) verkkosivuissa.[3] Reactilla toteutettuja tunnettuja verkkosovelluksia ja sivuja ovat esimerkiksi median suoratoistopalvelu Netflix, joka on erikoistunut elokuvaan ja televisiosarjoihin, rss-syötteiden lukupalvelu Feedly sekä vähittäismyyntiyritys Walmartin kotisivut. [4]

Angular on laajalti käytetyn MEAN-fullstack kehyspinon front-end kehys. MongoDB, ExpressJS, Angular, NodeJS ovat MEAN-pinon kehykset kokonaisuudessaan. Fullstack tarkoittaa kehyspinossa sitä, että se kattaa sovelluksen kaikki osa-alueet eli back-endin, front-endin sekä tietokannan.

Angularin ja Reactin kehittyneisyyttä ja laajaa levinneisyyttä selittää suuret yritykset niiden kehityksen takana. Angularia kehittää ja ylläpitää selainjätti Google ja Reactin pääkehittäjä on Facebook. Reactin kehityksessä on muka-

na myös Instagram sekä käyttäjien yhteisö.

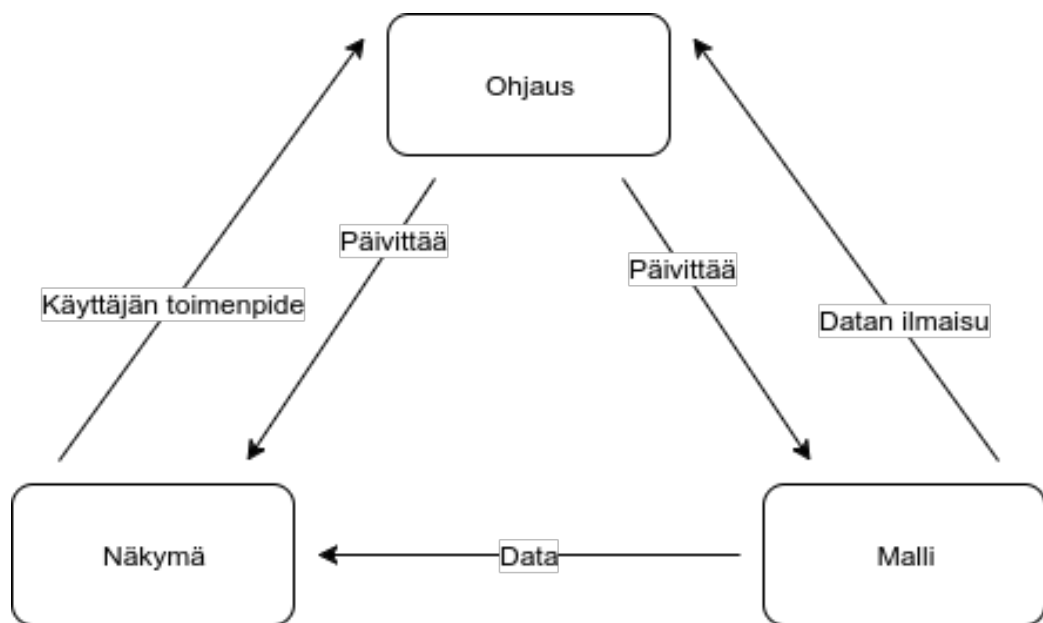
Angular on jaettu versioihin, joista tällä hetkellä vakaa käytössä oleva versio on 1.6.1, mutta versio 2 on ollut jo pitkän aikaa kehityksessä. Versio 2 tai vain Angular 2 on toteuttanut monia asioita hyvin erilailla aikaisempiin versioihin verrattuna. Käynkin asioita läpi pääasiallisesti Angular 2:n kannalta, vaikkakin monet asiat pätevät vanhempiinkin versioihin.

Reactia puolestaan kehittää yhteistyössä yhteisön kanssa Facebook sekä tämän tytäryhtiö Instagram. Reactissa ei ole suurempia muutoksia tuovia versioita vaan sitä kehitetään kasaamalla edellisen päälle uusia ominaisuuksia ja korjauksia.

Google vaihtoi Angularin kielen Javascriptistä Typescriptiin versionumeron vaihtuessa 1-alkuisista numeroista 2-alkuisiin numeroihin.

2 Ohjelmointikäytännöt

Angularin toteutus oli ennen versiota 2 lähellä perinteistä MVC(Model View Controller suomeksi malli, näkymä ja ohjaus)-rakennetta, jossa ohjelma on jaettu kolmeen osaan. Malli-osa määrittää dataan liittyvät asiat, näkymäosio hoitaa esitystavan ja ohjaus päivittää näkymiä, ottaa vastaan ja validoi syötteitä sekä hakee malli-osalta tietoja. Angularin malli sisältää tavallisia Javascript olioita, näkymä HTML-mallin ja ohjauksen hoitaa käytännössä näkvyysalueobjektit (englanniksi scope). Näkvyysalueobjektit ovat ohjauksen ja näkymän välisiä rajapintaobjekteja. Näkvyysalueobjektit poistettiin Angular 2 päivityksessä.[6]



React ei sovellu MVC-rakenteeseen, mutta jos sitä haluaisi ajatella kyseisen rakenteen kannalta, se toteuttaisi näkymäosion siitä. Reactissa näkymän osat jaetaan komponentteihin ja ne täydennetään esimerkiksi JSON tiedostoista tai tietokannoista saadulla datalla. Näkymän muokkaus tapahtuu kompo-

nentin metodeissa. Usein käyttäen reactiin valmiiksi luotuja elinkaarimeto-
deja kuten `componentDidMount()`, joka suoriutuu kun objekti on kiinitetty
DOM:iin (dokumenttioliomalli). [1]

Angular ja React -sovellukset toteuttavat tavallisesti olio-ohjelmoinnin para-
digmaa. Olio-ohjelmoinnissa ohjelmakoodi on jaettu olioihin ja oliot toteut-
tavat oman osansa ohjelman toiminnallisuudesta. [7] Koodi on siis modulaa-
rista ja usein sen abstraktiotaso pyritään pitämään korkealla, eli toteutetaan
sitä, että osia voidaan uudelleenkäyttää muualla.

3 Backendin kanssa kommunikointi

Angular käyttää sovelluksen backendin kanssa kommunikointiin palveluita, jotka lähettävät kutsuja backendille ja saavat vastauksena halutun datan. Palvelut ovat erillisessä tiedostossa määriteltyjä koodinpätkiä, jotka tekevät kutsuja backendille.

Reactissa puolestaan ei ole mitään sisäänrakennettua standardia tapaa kommunikoida backendin kanssa sillä se on vain kirjasto, eikä täysi front-end-kehys. Sensijaan back-end-kommunikaatio toteutetaan usein tavallisilla Javascriptin ajax-kutsuilla tai käyttäen muita kehyksiä apuna.

Angularin palvelut määritellään tarvittavissa elementeissä riippuvuuksina ja niistä luodaan ilmentymä kun komponentti tarvitsee niitä.

4 Datan sitominen ja DOM-manipulaatio

Front-end työkalujen tärkein tehtävä on datan näyttäminen näkymissä. Data jota näytetään voi muuttua ja usein käyttäjälle halutaan näyttää muutokset niin pian kuin mahdollista. Tämä on mahdollistettu datan sitomisella monimmissa tapauksissa. Toteutus työkalujen välillä tosin eroaa huomattavasti.

Reactin toteutuksessa muuttujia voidaan lisätä JSX-koodin sekaan kaarisulkeilla. Kaarisulkeiden sisällä olevat muuttujat ottavat arvonsa koodista. React-komponenteilla on erilaisia tiloja, joihin kuuluu muunmuassa komponentin luonti ja tuhoutuminen. Lisäksi voidaan määritellä intervalli, jonka välein tietty funktio suoritetaan. Näkymän päivittäminen vaatii joko intervallin tai komponentin tilan muutoksen.

Angularin toteutus eroaa Reactin toteutuksesta monella tavalla. Angularissa voidaan sitoa data samalla lailla yksisuuntaisesti kaarisulkeilla kuten Reactissa, mutta sen lisäksi on myös mahdollista tehdä kaksisuuntainen sidonta käyttämällä kaksia sulkeita. Yksisuuntaisessa sidonnassa data päivittyy muuttujien muuttuessa. Mikäli muuttuja on sidottu kaksisuuntaisesti toimii tämä myös toisin päin, eli muuttujan sisältö muuttuu näkymää muutettaessa. Esimerkiksi kun käyttäjä syöttää tekstikenttään jotain, muuttujan arvo muuttuu tekstikentän sisällöksi. Näkymien päivitystä ei tarvitse sitomisen jälkeen tehdä itse vaan se tapahtuu automaattisesti muuttujan muuttuessa.

Angular on siis kehittäjän kannalta helpompi, sillä siinä ei tarvitse miettiä päivitysintervalleja tai komponentin tilaa, toisin kuin Reactissa. React-sovellukset ovat yleensä kevyempiä

5 Suorituskyky ja kehittämisen vaikeudet

Angular on käyttänyt versiosta 2 alkaen kielenään Typescriptiä, joka on Javascriptin syntaksin ja semantiikan perivä kieli, joka kääntyy Javascriptiksi kääntövaiheessa. Sekä Typescript, että Reactin JSX vaativat molemmat käännöksen ennen kuin selain voi niitä suorittaa.[5]

Toisaalta Typescript tuo uusia työkaluja tavallisen Javascriptin päälle. Typescriptissä voi tyypitellä muuttujia eikä tarvitse siten tyytyä Javascriptin automaattisesti antamaan muuttujatyyppeihin. Tyypittelyllä saadaan ehkäistyä epätoivottuja lopputuloksia.

Monet asiat, jotka Angularissa ovat parin rivin koodinpätkiä vaativat Reactissa pidemmän pätkän koodia. Esimerkiksi sidotun muuttujan päivitys, jota ei Angularissa tarvitse miettiä, vaatii React-ohjelmoijalta toimenpiteitä. Monet Angularin tarjoamat helppokäyttöisyyttä edistävät seikat saattavat tehdä Angularista Reactia raskaamman.

Vaikka Typescript on Javascriptin "paranneltu versio", joidenkin mielestä Reactin JSX on mukavampaa kirjoittaa. Reactin lähestymistapa on siitä erikoinen, että kaikki komponenttien HTML-määrittelyt tehdään JSX:llä, joka näyttää HTML:tä, mutta kääntyy Javascriptiksi kun ohjelma suoritetaan. Esimerkiksi komponentin luominen JSX:llä tapahtuu seuraavalla tavalla.

```
<MyButton color="blue" shadowSize={2}>
```

```
  Click Me
```

```
</MyButton>
```

Mutta se muuttuu käännösvaiheessa muotoon:

```
React.createElement(  
  MyButton,  
  {color: 'blue', shadowSize: 2},  
  'Click Me'  
)
```

Tämä onkin jo tyypillistä Javascript-koodia, jossa ollaan käytetty vain React-kirjaston metodia `createElement`.

Viitteet

- [1] A JavaScript library for building user interfaces - React
<https://facebook.github.io/react/>
- [2] Made With Angular
<https://www.madewithangular.com/>
- [3] AngularJS - Wikipedia
<https://en.wikipedia.org/wiki/AngularJS>
- [4] Sites Using React
<https://github.com/facebook/react/wiki/sites-using-react>
- [5] Typescript - JavaScript that scales
<https://www.typescriptlang.org/>
- [6] Journal of Global Research in Computer Science *AngularJS: A Modern MVC Framework in JavaScript* Nilesh Jain, Priyanka Mangal, Deepak Mehta, Mandsaur Institue of Technology
- [7] *Object Oriented Programming* Bhanu Prasad Pokkunuri, ES Group, Central Electronics Engineering Research Institute