



Styrkor och svagheter hos Ember.js

Individuell inlämning i kursen JavaScript 2

Camilla Eriksson

Medieinstitutet i Göteborg

Februari 2020

Inledning och bakgrund

En av anledningarna till att Javascript har legat på mellan 7e och 9e plats på TIOBE Index över mest populära programmeringsspråk sedan 2002 är dess ramverk som signifikant förenklar kodningen för utvecklare. (Kravchenko, 2017) Men då det finns en mängd olika ramverk att välja på kan det vara svårt att veta när man ska välja vilket, speciellt som nybliven och oerfaren utvecklare. I denna rapport har jag valt att fördjupa mig i Ember.js för att ta reda på dess för- och nackdelar, när det är lämpligt att använda och vad skillnaden är mellan Ember och det mer populära React.

Bakgrund

Ember.js är ett ramverk baserat på MVVM (Model-view-viewmodel) som ska inkludera allt man behöver för att bygga kraftfulla UIs som fungerar på alla slags skärmar för ambitiösa webbutvecklare. Det används av utvecklingsteamerna hos bland annat Netflix, Microsoft, LinkedIn och Apple. (Emberjs.com, 2020) Det är ett open-source ramverk som används för att skapa single-page applications (SPA) och kombinerar de bästa egenskaperna hos sina större konkurrenter som Angular och React. (Kravchenko, 2017) Första releasen kom i december 2011 och det har sedan dess kontinuerligt släpps releaser som hela tiden har som grundtanke att ligga i framkant med framtida web-standards. Ember har varit tidiga adoptanter av många standarder runt JavaScript och webben, så som att inkludera components och ES6-syntax. (En.wikipedia.org, 2020)

Användning

Ember satsar hårt på att ha en bra "out-of-the-box-produkt" för att utvecklaren ska kunna komma igång och vara produktiv omedelbart. Till exempel så skapar Ember CLI en standard applikationsstruktur och en build-funktion. (En.wikipedia.org, 2020) Detta gör det väldigt smidigt att sätta upp ett nytt projekt, men till skillnad från React så tar det väldigt lång tid att lära sig Ember. (Ranjan, 2020)

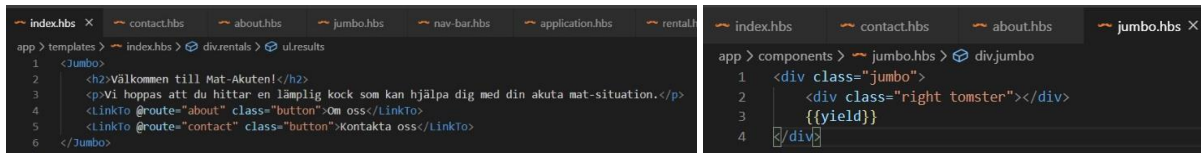
Ember är ett av de tyngsta ramverken vilket innebär att det tar väldigt lång tid att installera samt bygga projekt. Detta är speciellt ett problem för Windows-användare, eftersom Embers prestanda är bättre på både Linux och MacOS. Både teamet på Ember CLI och Microsoft jobbar dock på att förbättra detta. (Cli.emberjs.com, 2020) Med tanke på detta lämpar sig Ember mest för större, komplexa, funktionsfyllda applikationer, vilket de själva också menar när de säger att det riktar sig till "ambitiösa" utvecklare och applikationer. React, å andra sidan, lämpar sig för alla typer av projekt, både stora som små.

Ytterligare skillnader mellan de två ramverken är att Ember använder two-way data binding medans React använder one-way data binding (Ranjan, 2020), samt att React använder sig av en virtuell DOM (som är kompatibel med andra frontends som tex JSX och hyperscript) som renderingsmotor, medans Ember använder sig av sin egna renderingsmotor Glimmer. Tack vare att Ember använder sig av handlebar templating-biblioteket så kan Glimmer se skillnad på statiska och dynamiska komponenter och på så sätt reducera antalet element som behöver kollas när den letar efter ändringar. (Peyrott, 2020)

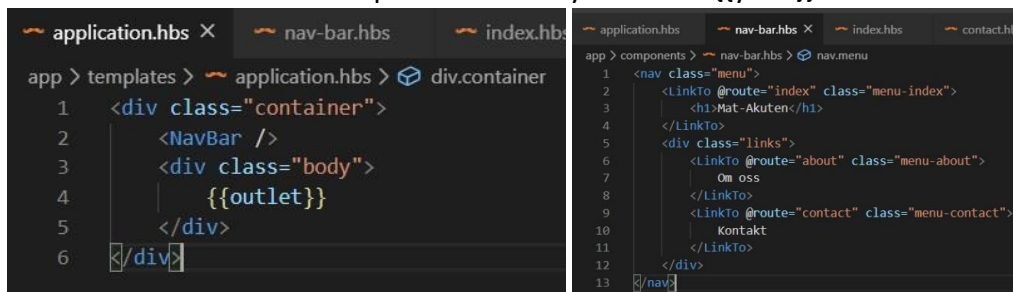
Genomförande av "proof of concept"

För mitt kodexempel har jag följt en tutorial på Ember.js's officiella hemsida. Jag tycker att den var mycket utförlig, pedagogisk och lätt att följa. Jag har dock hoppat över stegen med tester och istället koncentrerat mig på att lära mig syntaxen och strukturen. Jag har inte lagt ner någon tid på layouten, av samma anledning.

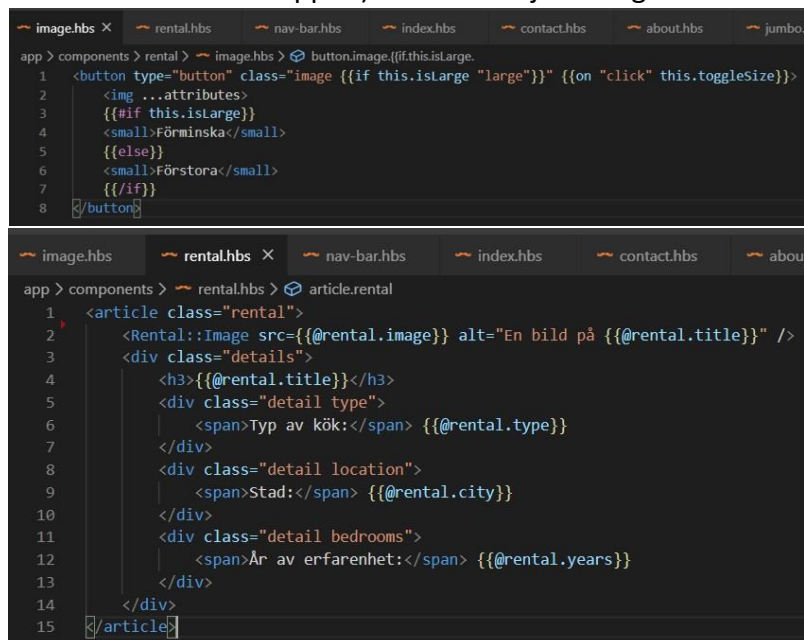
Kodexempel:



Här ser man hur jag skapar en instans av jumbo-komponenten i index-templaten, istället för att skriva ut två div:ar med rätt klass i varje enskild template. Jag skickar med det som ska stå inuti komponenten via nyckelordet `{{yield}}`.



Liknande fungerar det med nav-baren, med skillnaden att denna ska ligga på alla sidor och se precis likadan ut, och behöver därför inte `{{yield}}` och ligger i application-templaten (som är speciell eftersom att den inte har en egen url utan bara specificerar en gemensam layout för alla sidor i appen) där den är självstängande.

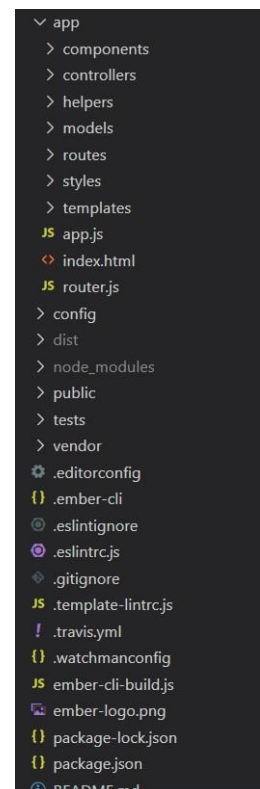


I image-komponenten ligger en `img`-tagg med "splattribut syntax" som tar emot `src` samt `alt` ifrån rental-komponenten. Jag har fejkat ett api med en json-fil med tre olika kockar som tas emot i rental-komponenten. I image-komponenten ligger också en if-sats som avgör om `isLarge` är `true` eller `false`, vilket ändras i on-click-funktionen på hela bilden.

Diskussion

När jag suttit och kodat med Ember för att skapa mitt "proof of concept" och läst på om ramverket har jag stött på både för och nackdelar. Den största nackdelen enligt mig är att det är så pass tungt och tar väldigt lång tid att bygga. Det blir efter ett tag irriterande att arbeta i en så långsam miljö. En annan nackdel är att trots att det ska vara enkelt att komma igång med den förinställda filstrukturen som skapas automatiskt så tyckte jag att det som nybörjare var väldigt förvirrande med alla filer och mappar. Även om man aldrig behöver gå in och kolla i många av filerna (innehåller i app.js, index.html och resolver.js tex genereras automatisk utan att man behöver ändra något själv) så är det svårt att förstå strukturen. Detta är dock säkert en vanesak och något man kommer in i om man sitter och jobbar med Ember under en längre period.

En av sakerna jag gillar med Ember är att det har mycket starkt inbyggt best practice. Även om det finns frihet att ändra hur saker görs, så är best practices inbyggda i ramverkets struktur. Detta gör också att det är lättare för utvecklare att hoppa in i redan existerande Ember-projekt, eftersom att strukturen och koden ser nästan exakt likadan ut. Det blir också lättare att byta mellan olika Ember-projekt, av samma anledning. (Pluralsight.com, 2015)



Jag tycker att Embers dokumentation är mer utförlig och lättförståelig än Reacts dokumentation. Jag upplever dock att även om det finns en hel del hjälp att hitta om Ember på diverse forum, i artiklar och på tutorials på Youtube så är det mesta flera år gammalt och inte längre applicerbart. Där har React en fördel eftersom att även om det inte finns lika mycket information att hitta, så är det som finns i alla fall uppdaterat.

Till dess att jag är mer erfaren utvecklare och har bättre förståelse för olika kodstrukturer så föredrar jag React som känns enklare att förstå, går snabbare att arbeta med och som det är enklare att göra små appar med.

Referenser

Kravchenko, J. (2017). *Angular, Ember and Vue: Is choosing a framework simply a matter of taste?* - JAXenter. [online]

Available at: <https://jaxenter.com/angular-ember-or-vue-choice-135987.html>

[Accessed 22 Feb. 2020].

Emberjs.com. (2020). *Ember.js - A framework for ambitious web developers*. [online]

Available at: <https://emberjs.com/>

[Accessed 22 Feb. 2020].

En.wikipedia.org. (2020). *Ember.js*. [online]

Available at: <https://en.wikipedia.org/wiki/Ember.js>

[Accessed 22 Feb. 2020].

Cli.emberjs.com. (2020). *Windows support - Appendix - Ember CLI Guides*. [online]

Available at: <https://cli.emberjs.com/release/appendix/windows/>

[Accessed 22 Feb. 2020].

Ranjan, A. (2020). *EmberJS vs ReactJS - GeeksforGeeks*. [online]

Available at: <https://www.geeksforgeeks.org/emberjs-vs-reactjs/>

[Accessed 22 Feb. 2020].

Peyrott, S. (2020). *React Virtual DOM vs Incremental DOM vs Ember's Glimmer: Fight*.

[online]

Available at: <https://auth0.com/blog/face-off-virtual-dom-vs-incremental-dom-vs-glimmer/>

[Accessed 22 Feb. 2020].

Pluralsight.com. (2015). *7 Reasons to Use Ember.js*. [online]

Available at: [https://www.pluralsight.com/blog/software-development/7-reasons-to-use-](https://www.pluralsight.com/blog/software-development/7-reasons-to-use-ember-)

[ember-
js?clickid=W9byP20AlxyOWKewUx0Mo3EUUknVA7wdSUp0R40&irgwc=1&mpid=269814&utm_source=impactradius&utm_medium=digital_affiliate&utm_campaign=269814&aid=7010a000001xAKZAA2](https://www.pluralsight.com/blog/software-development/7-reasons-to-use-ember-js?clickid=W9byP20AlxyOWKewUx0Mo3EUUknVA7wdSUp0R40&irgwc=1&mpid=269814&utm_source=impactradius&utm_medium=digital_affiliate&utm_campaign=269814&aid=7010a000001xAKZAA2)

[Accessed 22 Feb. 2020].