

Proji.se – Ett projekthanteringssystem

Kristoffer Lind (2014-05-26)

Abstrakt

Rapporten avhandlar tankar kring uppbyggnaden av projekthanteringssystemet proj.se. Detta utvecklades för att förenkla samarbete och effektivisera arbetsprocessen i ett projektarbete.

Applikationen är byggd med AngularJS och Firebase. Under utvecklingsprocessen användes även Node.js, Grunt, Bower, Yeoman, Protractor och Travis CI. Applikationen fungerar till det ämnade ändamålet, men vidare arbete kommer att utföras för att nå målbilden.

Innehållsförteckning

Abstrakt.....2

Bakgrund.....4

Positiva erfarenheter.....5

Negativa erfarenheter.....6

Sammanfattning.....7

Bakgrund

Utan att använda sig av något system för projekthantering är det omständligt att samarbeta och strukturera upp information överskådligt; det blir lätt rörigt och ineffektivt med uppgifter som behöver ändras på flera platser. Målet var därför att skapa en applikation som förenklar arbetsprocessen vid ett projektarbete. Kombinationen med AngularJS¹ och Firebase² valdes eftersom MEAN³ Stack skulle innebära ett för omfattande arbetsåtagande under den tänkta projekttiden.

Metod

Applikationen har byggts med AngularJS och Firebase. AngularJS är ett ramverk för klientsidan och Firebase är databas i form av en tjänst. Testning har utförts med Karma⁴ och Protractor⁵; Karma är en testmiljö för enhetstester och Protractor är en wrapper för Selenium⁶, vilket används för integrationstestning. Travis CI⁷ har använts för kontinuerlig integration; Travis CI är en tjänst för att köra uppgifter i samband med commit till versionshanteringen på Github. Vid varje commit provbyggs applikationens produktionskod, samt att enhetstester och integrationstester körs. Node.js⁸, Bower⁹, Yeoman¹⁰ och Grunt¹¹ har använts i utvecklingsprocessen. Node.js är en plattform för att köra JavaScript utanför webbläsare, dess pakethantering används i projektet för bibliotek till utvecklingsmiljön och behövs som plattform för att köra en del av utvecklingsverktygen. Bower är en pakethanterare och används för bibliotek på klientsidan. Yeoman är en "scaffolder" och har använts till att generera grunden för projektet. Grunt är en "uppgiftskörare" och har använts för att automatisera processer som körs ofta. Node.js har också använts för att underlätta installationsprocessen, dess pakethanterare används för att automatisera installationen. Kommandot npm install installerar normalt sett bara de beroenden som finns i sin pakethanterare, men har scriptats vidare så att det även installerar Bowers beroenden, samt hämtar och installerar Selenium.

1 <https://angularjs.org/>

2 <https://www.firebase.com/>

3 MongoDB, Express.js, Angular.js, Node.js

4 <http://karma-runner.github.io/>

5 <https://github.com/angular/protractor>

6 <http://docs.seleniumhq.org/>

7 <https://travis-ci.org/>

8 <http://nodejs.org/>

9 <http://bower.io/>

10 <http://yeoman.io/>

11 <http://gruntjs.com/>

Positiva erfarenheter

Möjligheten att få välja tekniker själv har varit bra, det har gett mig möjlighet att undersöka tekniker jag trodde skulle vara intressanta. Nu är jag nästan helt övertygad om att detta är vad jag vill arbeta med efter utbildningen. Kunskapen om alla verktyg jag använt är något jag definitivt kan ta med mig, framförallt till framtida projekt med mycket javascript, men även mer generella kunskaper. Testningen ger mig en bra grund för att närma mig TDD, Karma kör enhetstesterna på <1s när något sparas, integrationstesterna kör jag on demand med Protractor och Travis CI provbygger hela applikationen och kör både enhetstester och integrationstester vid varje commit. Efter att ha skrivit om integrationstesterna till att följa page object model, tycker jag att jag fått till en väldigt bra testmiljö, det skulle nu vara ganska enkelt att lägga till mycket fler tester, testningen har dock redan fått tre veckor och jag valde därför att istället lägga fokus på applikationen. Bower och npm sköter pakethantering väldigt väl, med dessa har jag också lyckats scripta upp framförallt installation, som annars är en del steg att beskriva. Angularjs har varit väldigt trevligt att arbeta med och Firebase har hanterat enklare databasförfrågningar väl. Vad gäller dokumentationen tycker jag att det fungerat bra med det format jag haft på sprint backlog. Även kommunikation har fungerat bra, hörde av mig till John Häggerud som ställt upp och haft mycket handledningstid med mig. Jag har också diskuterat kod och verktyg i stort sett dagligen med Anton K Andersson. Vi är också en hel del studenter som använder Slack som en kommunikationskanal där vi hjälps åt att hålla oss uppdaterade med ny teknik och problemlösning. Dessa kommunikationskanaler är ovärderliga, både för egna funderingar och den kunskapsfördjupning det innebär att försöka svara på frågor. Jag är mycket stolt över vad jag lyckats åstadkomma under kursen och kommer definitivt att arbeta vidare med projektet.

Negativa erfarenheter

Product backlog har inte fungerat, det blir en del dubbeljobb; varje avslutad uppgift ska markeras som klar på två platser. Detta har jag gjort veckovis och vid något tillfälle har jag misslyckats med synkroniseringen. Lyckades inte heller riktigt få något bra format för punkter som inte är färdiga idéer och det har blivit lite rörigt då jag haft product backlog, sprint backlog, funderingar.md samt en del uppgifter efter workshops och testsessioner som bara funnits på papper. Dessa har jag valt att väva in i sprint backlog som grupperingar.

Jag misslyckades till viss del med bedömningen av arbestmängden, jag trodde inte att jag skulle uppnå målbilden, men min bedömning var att jag skulle vara närmare den.

Firebase har i ett par fall snarare varit ett hinder än hjälp, det har varit otroligt smidigt till enklare uppslag, men väldigt omständligt vid mer komplicerade anrop. Under ett par timmar var det problem med Firebase och då kunde jag inte göra någonting alls eftersom jag även i utvecklingsmiljön arbetat mot Firebase. Fortsättningsvis ska jag vara försiktig med att göra mig beroende av yttre tjänster.

Källkoden har stundvis varit rörig eftersom jag fått göra stora förändringar efterhand som jag lärt mig. Inlärningsprocessen har resulterat i ganska mycket dubbeljobb. Statistiken är just nu 34,4k++ 22.8k--, ungefär 1/3 behållning av kod i versionshanteringen. Dessa stora förändringar har resulterat i att jag troligen har en del källkod som inte längre används(död kod).

Fortsättningsvis bör jag nog börja med ett mindre projekt för första fasen av inläring, det är inte särskilt mycket av det som blir användbart ändå.

Sammanfattning

Jag är nöjd med resultatet av denna period, men det är mycket arbete som kvarstår innan målbilden är uppnådd. Applikationen har potential att bli bra och jag kommer därför fortsätta arbetet efter kurslut. Första steget blir att byta ut Firebase mot MongoDB, Nodejs och Express. Detta var mitt ursprungliga val av tekniker, jag valde dock att under kursen använda Firebase för att minska arbetsbördan. Att sätta mig in i Firebase har gett mycket idéer för hur en egen lösning skulle kunna fungera och jag kommer definitivt titta mer på Omnibinder (Firebase använder detta för sin synkronisering). Däremot hade det blivit mindre dubbelt arbete om jag satt mig in i MEAN Stack från början. Min målsättning är att bli väldigt duktig på MEAN Stack, eftersom jag tror att det är det här jag vill arbeta med efter utbildningen. Under projektiden lade jag tre veckor på testning, detta har tagit mig ett steg närmare TDD, vilket jag varit nyfiken på sedan experimenterande med Busterjs i Websteknik I.