Høyne din kodekvalitet med statisk typing og hissig linting

Hans Ole Gjerdrum
Systek AS

14. januar 2018

1 Introduction

Alt for ofte, når jeg bytter prosjekt eller åpner et nytt repo, slår det meg i ansiktet at koden mest ser ut som et tenåringsrom. Og hvem bryr seg egentlig om hvordan kodekvaliteten? Du kan jo hevde at det bare er du selv som sitter i din kode-borg og regjerer allmektig, men da glemmer du de gangene du trenger hjelp, eller skal par-programere deg gjennom et problem, og den dagen kommer da du skal forlate prosjektet og gå videre. Da er det koden du etterlater deg som er ditt *ettermæle*

I nesten alle prosjektene jeg har jobbet som front-end-utvikler har jeg sittet alene med ansvaret for JavaScript-koden. Mens Javautviklerene lenge har hatt prosesser og verktøy for å sikre kvaliteten på deres kode, har ikke vi frontendere jobbet etter slike metoder. Men moderne frontend-kode trenger også slike metoder.

Og det var kanskje ikke så farlig så lenge man jobbet i *global.js* på < 500 linjer kode med validering av inputfelter. Men moderne applikasjoner skal løse uhyre mer kompliserte oppgaver, de

$=>$ Lesbar, forståelig og forvaltbar kode

2 Lint

2.1 Hva er Linting

Fra **Oxford Dictionary** har vi at lint er:

> Short, fine fibres which separate from the surface of cloth or yarn during processing.

... det vi på norsk kaller *lo*. Innen informatikken er linting brukt om verktøy som skal hjelpe til med å oppdage og å fjerne kildekodens hybelkaniner, eller sagt på en annen måte *Statisk analyse av kildekode for å detektere brudd på definerte regler*. Lint, som verktøy, ble først utviklet av Stephen C. Johnson hos Bell Labs for sjekk av C-kode. Senere har det ingått som del av Unix OS.

2.2 Linting i JavaScript

Det første verktøyet for å *linte* JavaScript-kode kom i 2002. Det var online kodesjekkeren **JSLint**, utviklet av Douglas Crockford. Ved lime js-koden sin inn i textarea på siden http://jslint.com/ fikk man feilmeldinger og advarsler om hvor koden fravekt et sett med forhåndsdefinerte regler for hva god kode skulle innebære. For bedre å kunne tilpasse disse reglene etter egne behov, forket Anton Kovalyov i 2010 JSLint ut til prosjektet **JSHint**. JSHint ble også loevert som kommandolinje klient distribuert som mode-module, at linting kunne bli et steg av kodebyggingen. Dagens "industristandard" må sies å være Nicholas C. Zakas' **ESLint** fra 2013. Dette verktøyet har tatt brukertilpassingen ennå et steg videre, og har også muliggjort å anngi hvilke EcmaSript-versjn koden er skrevet i. For utivklere på TypeScript finnes det også en linte-verktøy kalt **TSLint** utviklet av Palantir Technologies.

3 Type

Outline

4 Results

In this section we describe the results.

5 Conclusions

We worked hard, and achieved very little.