# Testrapport for gruppe 42

Lever en rapport som dokumenterer hvilken gruppe dere har testet, hva som er utviklet og resultatene fra testingen

## Innledning

Gruppe 42 har utviklet en webapp for å opprette og holde oversikt over eventer, lage lister av gjøremål og lage notater. Gruppene ble enige om å bruke Jest med Enzyme som rammeverk, og har stort sett brukt shallow rendering.

Gruppene møttes på forhånd for å sette opp et rammeverk sammen og bli enige om hva som bør testes. I etterkant har vi hatt kontinuerlig kommunikasjon over chat hvor vi ga tilbakemelding om resultatene våre.

## Resultater fra testene

Vi startet med å teste at alle komponentene ble rendret som de skulle. Da oppdaget vi at alle komponentene som importerte Datepicker-komponenten, eller importerte en komponent som importerte Datepicker, utløste en feilmelding. Feilmeldingen kom fra en intern kode i Datepicker, som hadde en syntaksfeil. De berørte komponentene var:

* EventContainer
* Home
* App

Andre feilmeldinger fikk vi fordi Clock- og Todo-komponentene ikke ble eksportert (export default <komponent>) på slutten av filen.

Note komponenten var den eneste av de atomiske komponentene som fikk inn funksjonene sine som parametre fra en forelder-komponent. Dette gjorde den mer egnet for testing enn de andre. Vi fikk derfor testet funksjonaliteten som endrer tittel og tekst, og denne virket som den skulle. Note hadde ingen slette-funksjon som kunne testes.

Sletting av en event skulle også testes, men på grunn av feilmeldingen som skjer når vi importerer EventContainer er ikke dette mulig fordi alle eventene lagres i en state i EventContainer.

For å teste Note-komponenten ble vi nødt til å legge til en unik id på input-elementene for å kunne teste dem hver for seg. Inputelementene med name= ‘tittel’ fikk id=’tittelid’, mens inputelementet med name=’text’, fikk id=’tekstid’. Dette var fordi vi måtte ha mulighet til å hente ut et spesifikt input-element , men siden de er likestilt i komponenthierarkiet og har samme klassenavn så fikk vi ut alle. Derfor lagde vi unike id’er som vi kunne bruke for å hente de ut og teste dem hver for seg.

Vi testet ikke Calendar-komponenten, i og med at den var importert fra et eksternt bibliotek, og det ga dermed ikke mening å teste denne. Det var ingen interaksjon med CalendarContainer, annet enn at den hentet events fra LocalStorage. Derfor har vi ikke skrevet tester til denne komponenten, med unntak av en render-test.

Clock-komponenten har ingen brukerinteraksjon, og viser kun tiden. Dette er trivielt å teste ved å kjøre den og deretter sammenligne resultatet med hva klokken er. Det er derimot veldig komplisert å teste ved å skrive komponenttester. Dette er fordi man må sørge for at klokken ikke endrer state fra du setter den til du sjekker den, noe den gjør kontinuerlig med en tick() funksjon. Vi anser dette som litt utenfor scope, og har derfor ikke skrevet tester. Vi kan derimot melde at klokkekomponenten viser riktig tid når den kjøres.

Vi fant en måte å teste localStorage på ved å bruke en mockup (test av denne er i Event.snapshot.test), men vi klarte ikke å få dette til å virke i praksis med testene våre. Dette er delvis fordi funksjonene som interagerte med localStorage ikke kunne bli kalt ved å simulere brukerinteraksjon med komponentene, og delvis fordi dersom vi eksplisitt kalte på disse funksjonene, så kunne ikke resultatet testes ved å sjekke komponentens state-verdier siden disse ofte var ikke-eksisterende. Dette hadde vært lettere dersom kodens struktur hadde vært annerledes, slik at funksjoner ble sendt nedover i hierarkiet og f. eks ble definert som onClick på knapper ol, og at komponentene i seg selv hadde en state. Slik som det er nå virker det som at nesten alle funksjonene er interne funksjoner i komponentene, og det er få funksjoner som kalles direkte ved brukerinteraksjon.