

Inlämning: The proof concept

Appen är än fortsatt utveckling från förgående inlämning: Rätt förutsättningar. Jag har inför denna inlämning skapat en ny branch V2 där react är implementerad. Jag har även i denna modul valt att installera två javapaketer, react-router-dom och axios. När båda är installerade har en commit gjorts, vilket gör att det går att återställa med git revert.

Jag har inte lyckats hinna med att få till 8 komponenter.

Jag har använt state i följande komponenter: ProfilePage, Cards, Inputfields.

Conditional rendering, finns i StartPage. Detta styrs via state.loggedIn. Tanken är att en inloggad användare inte ska se inloggningsformuläret. Jag har inte riktigt implementerat den så det fungerar som jag vill ha det då jag saknar en databas för att lagra min användare.

Detta gör att appen inte har någonstans att verifiera inloggningen. Mina komponenter Cards, InputsField och Profilepage uppdatera state när användaren skriver i formuläret för Cards. Från ProfilePage skickas en funktion ner i träd och körs från Inputfields.

Förklara med egna ord vad som är viktigt att tänka på vid skapandet av webbapplikationer med React (ungefär tre paragrafer).

Man bör planera i förväg och ha en struktur för sitt projekt. Det är bra att titta igenom vilka komponenter som behövs och hur dessa integreras med varandra.

Undvik att använda state om det går, det är bättre att använda props istället och låta en komponent styra så mycket som möjligt.

Genom att göra så kan man avgöra vart state behövs och använda dessa om de bara påverkar den givna komponenten.

Beskriv och förklara användningsområdet av det ramverk/bibliotek utöver React som du använt i din applikation.

Jag lade till react-router paketet. Jag använde detta paket eftersom det hjälper mig med routing i min app. Jag tänker mig ha minst 3 sidor till att börja med, registration, start och profil. Paketet gör det lätt för mig att lägga till fler routes i takt med att appen byggs ut.

Ett annat paket jag lade till är axios. Detta hjälper mig att fetcha hundbilder till profilsidan.

Axios kommer förmodligen ersättas eller användas till något helt annat när en databas är upplagd för att hantera användardata.

Förklara med egna ord vad som är viktigt att tänka på vid användningen av lokal data i React (ungefär en paragraf).

Vid lokalt sparande av data eller hämta av sådan data bruka man spara dessa i komponenternas tillståndshantering, dvs state. Det är viktigt när man uppdatera state se till så anropet är klart innan man ritar gränssnittet. Detta beror på att funktioner sker asynkront.

Vilka åtgärder har du genomfört i din källkodshantering för att kunna rulla tillbaka och vidareutveckla projektet i framtiden? Varför är dessa lämpliga val?

Mitt tankesätt med källkodshantering är att införa en ny branch när det blir en större ändring och commits för mindre.

Jag har därför en ny branch V2 där react lades till. Detta är eftersom det ändra helt hur app.html fungerar.

Alla ändringar som jag sedan gör för att bygga ut app.html är gjorda med commits.

Jag har t.ex en commit där jag lade till **react-router**, vilket gör att jag enkelt kan rulla tillbaka med revert.

Jag tycker att detta sätt att jobba gör att jag har god kontroll på hur projektet går. Externa som använder mitt projekt blir inte heller drabbad när större ändringar införs eftersom dessa görs på ny branch. Samma sak det lätt att rulla tillbaka mindre uppdateringar om det inte fungerar så som det skall med revert.

Gällande det bibliotek/ramverk utöver React som du lagt till i ditt projekt - vad tycker du detta paket tillför för styrkor till ditt projekt och vilka begränsningar introducerar det för framtida designval?

React-router hjälper mig med routing. Fördelarna till skillnad från vanliga routing med <a> är att sidorna ej laddas om. Detta är perfekt för spa. En annan fördel är att det finns en funktion för dynamiska sidor vilket jag kan använda när appen expanderar.

Nackdelarna är en av det som kommer med spa, dvs laddning av startsidan kan ta längre tid eftersom alla filer som behövs för appen laddas ner direkt. Med react-router kommer också dependency till andra paket vilket kräver mer koder och är beroende av andra paket till skillnad från traditionell <a>.

Detta sättet att jobba (client-side routing) gör också att search-engine-optimizer blir mindre optimerad.

Analysera och reflektera kring den eventuella byggprocessen du använt för att kompilera JSX koden i ditt projekt - vad var bra och dåligt med att välja denna metod?

Jag lade till React till mitt projekt mha webpack. Standard transpiler som kommer med webpack klarar inte per automatik att översätta JSX-koden till javascript. För att få det att fungera installerades babel-loader.

Fördelarna att göra så är att det är flexibel och man kan konfigurera lite som man vill. I detta projekt ville vi att react startar från app.html vilket inte är lätt att göra med t.ex create-react-app. Nackdelarna är att man får konfigurera själv och sätta upp systemet från grunden för det skall fungera. Vilket tar tid och är inte direkt lätt att få rätt på allt direkt.

Reflektion

Har bara lyckats få Profilsidan att fungera någorlunda. Det jag saknar för att vidareutveckla appen är någon typ av databas där jag kan lagra användare och dess data. Jag behöver förmodligen också kunna skicka props till mina sidor. Detta är för jag vill hålla reda på om en användare är inloggad samt kunna låta användaren skapa forumtråd eller skicka meddelanden.

Här har jag inte riktigt klurat ut hur data bör skickas då jag använder react-router-dom och kör egentligen bara Link till mina sidor. Jag tänker mig jag måste få en tydlig struktur innan jag går vidare med att vidareutveckla appen.

Av: Fon To

Github: https://github.com/fonto1/dog_app , branch V2