

1. Börja med att skapa ett nytt projekt (applikation) genom vite. Gå sedan in i mappen som skapades och fortsätt därifrån.
2. Installera det nödvändiga npm-paketet: react-router-dom
3. Nu behöver du beskriva för din applikation att den skall använda en router. Detta kan göras på olika sätt i olika filer, men för att hålla det enkelt rekommenderar jag att göra detta i index.tsx. I denna fil behöver du ersätta `<App />` med `<RouterProvider>`. Glöm inte bort att importera RouterProvider från react-router-dom.
4. Nu skall vi skapa din router. Skapa en ny fil som heter Router.tsx, vars syfte är att definiera alla sökvägar och komponenter som skall kunna användas av vår router.
5. Inuti Router.tsx, skapa en exporterad const som heter t.ex. router. Denna variabel skall vara resultatet av createBrowserRouter() från react-router-dom.
6. När du anropar createBrowserRouter behöver du skicka med en parameter som är en lista med routes. Varje objekt i listan behöver innehålla minst två egenskaper, path och element. Syftet med path är att beskriva vilket innehåll som [url:n](#) i webbläsaren skall ha för att visa komponenten som är värde till element.
7. Inuti din [] skall du nu skriva vilka routes du vill ha i din applikation. T.ex. Home, About och Contact brukar vara vanliga routes att ha. För varje route behöver du skapa upp ett objekt. Detta objekt skall ha attributet path som beskriver värdet efter ditt domännamn. Sedan kommer attributet element att beskriva vilken komponent som skall visas för den aktuella sökvägen.
8. Skapa nu upp routes för Home, About och Contact i din applikation. Eller om du vill ha andra så går det också bra såklart, dessa är bara exempel. Ibland är det viktigt att kunna skicka information till en route. Ett vanligt exempel på detta är när vi vill visa detaljer kring en produkt. Då behöver vi skicka med id:t för produkten till routen så att den komponenten vet vilken produkt som skall beskrivas. Detta löser vi genom att skicka med information som en url parameter.
9. Skapa upp ytterligare en route som skall ta in en url-parameter. Detta görs genom att du i din path för route-objektet anger dels vilken url som skall gälla, men också vilken parameter som skall användas. T.ex.

```
{ path="/product/:id" element=<Product /> }
```

Nu kommer url:en till denna route att bli t.ex. /product/4 och då är tanken att det är produkt med id 4 som skall visas.

10. På mottagarsidan, alltså den komponent som skall visas på den route som har en urlparameter så behöver den komponenten ta reda på vilken parameter som har skickats. Detta görs genom att använda den hook som heter useParams(). Men för att använda denna hook behöver du veta exakt vilken parameter som skickats. Var noga här! Tänk på vilken typ

av komponent du behöver göra för att kunna använda en hook.

11. Använd värdet från föregående exempel för att skriva ut det i konsolen. Eller på skärmen med html. Eller varför inte hämta ett objekt från ett api baserat på url-parametern? Finns inga gränser för vad ni kan göra här.