# Start up React

Att starta upp ett React projekt kan göras på flera olika sätt. Detta är något som också har förändrats med tiden. Allt eftersom React har fortsatt finnas har det också vidareutvecklats vilket har lett till att funktioner och smidigare lösningar uppkommit och fortfarande fortsätter att utvecklas.

Denna rapport är en guide till hur man enkelt kommer igång med sitt React projekt.

## Node.js - [1]

React kommer behöva använda sig utav Node. Node är en slags JavaScript miljö som möjliggör att kunna skriva JavaScript-kod utanför en webbläsare. [2] I detta projekt skapar den en slags server som JavaScriptet som skrivs i React kan köras och testas på.

* Så se till att ha det installerat innan du börjar. För att se så att Node är installerat på datorn och se vilken version man just nu har kan man ange kommandot i konsolen; *node -v.*

### NPM – [3]

Npm kommer med Node och är ett kommandoverktyg som man använder för att kalla på Nodes pakethanterare. Npm är en pakethanteraren för JavaScript med stor lista över återanvändbara kodmoduler och paket. [3]

* För att se så man har en fungerade eller kolla vilken version man har utav Npm skriver man *npm-v* i konsolen för att kolla vilken version. Det ska vara en version som är högre än 3 för att kunna köras med Node.
* Kommandot *npm start* används i konsolen när du befinner dig i en mapp där React projektet befinner sig. *npm start* startar då upp servern och gör det tillgängligt att köra sin kod där.

## Create React App – [4]

Funktionen Create React App hjälper dig konfigurera en modern webbapplikation genom att endast köra ett kommando. Funktionen hjälper dig så att du inte böhver lära eller konfigurera så många byggverktyg som tidigare. Konfigurationen hjälper dig att fokusera på utvecklingen istället. När det är dags att distribuera optimeras dina filer automatiskt. [5]

Under huven av funktionen används Webpack, Babel, ESLint och andra fantastiska projekt för att driva applikationen. [5]

För att utföra kommandot och skapa ett nytt Reactprojekt placerar dig du via konsolen i den mapp där projektet ska skapas. I bilaga 1 kan man se hur detta går till i konsolen. [b1] I konsolen skriver man sedan via npx;

* *npx create-react-app namn\_på\_app*

När installationen har lyckats behöver man bara gå in i den nya mapp som man precis har skapat via kommandot och har namnet som man har bestämt via konsolen. När du väl står inne i Reactprojektet skriver man kommandot *npm start* för att starta igång den miljö som man kommer köra applikationen på;

## ATOM IO – [6]

För att sen skriva kod och modifiera behövs det en kodeditor. Atom IO är en gratis open-source kodredigerare för operativsystemen Windows, Linux och macOS. Atom har stöd för plug-in program skriva i Node.js.

I detta projektet kommer formaten HTML, CSS, JS och JSX användas. Därför är en passande kodeditor ATOM IO. Atom innehåller allt som behövs och är användarvänlig. I Atom kan man smidigt bygga på önskade packet för att underlätta sitt arbete. Man kan enkelt skapa sina egna tangentbords-genvägar, där av indentering av kod och autofyllfunktioner, som gör kodskrivningen effektivare.

För att smidigt få upp sitt nyskapade projekt kan man i samma mapp som man tidigare ändvänt kommandot *npm start* för att starta projektet skriva kommandot *atom .* . Kommandot innehåller en punk efter ordet atom och vad det gör är att det öppnar upp projektkatalogen i Atoms grafiska gränssnitt i den på den plats som önskats.

# ”create-react-app”-kommandot – [4]

Som tidigare nämnt skapar kommandot create-react-app flera olika saker åt oss. En sak utav dessa är det filer och den mappstruktur som ingår i vårt projekt.

## Mappar

Vid konfigurering av projektet skapar tre mappar som kan vara bra att hålla reda på. Node\_modules är en utav dessa och innehåller alla de delar av biblioteken vi kommer att använda och de vi kommer installera i projektet. Där hittar man alla Reacts-funktioner och alla andra funktioner som man använder sig utav som ingår i de olika biblioteken.

I mappen som heter src är i den vi skriver all Reactkod i och lägger alla React-komponenter. Vilket vanligast sker med JavaScript eller JavaScript Extension. Men här kan man även hitta filer som CSS. Är läggs all den kod som ska visas för användaren och man bygger här komponenter som man sätter ihop något att visa.

I mappen pubic ligger bland annat den html-fil som först kommer köras på webbapplikationen. Det att hålla reda på i den filen är att man ska referera till det id i den div-tagg där man vill rendera den Reactkod som man har skrivit.

## Filer

### package-lock.json – [7]

package-lock.json håller koll på alla versioner utav paketen och vilka det är som krävs för att köra projektet, med alla våra beroenden. När man installerar paket kan ett kommando vara värt att lägga på minnet som innehåller växelspaken –save. Den används tillsammans med npm och ser till så att packen som läggs till eller installeras i projektet sparas i package-lock.json filen. Detta gör så att man kan ta bort node\_modules mappen och skriva npm install för att installera samtliga av projektets packet på nytt.

### Package JSON – [8]

Detta är en fil som skapas då man börjar installera packet till ditt projekt. Själva filen innehåller mycket metadata om projektet. Mestadels används den för två saker; att hantera alla packet, beroenden och moduler som används av koden, men även olika skript som hjälper till att hantera byggnader och exekveringar med mera. [6]

# Källförteckning

[1] - Node JS - <https://nodejs.org>

[2] – Node JS Foundation - <https://foundation.nodejs.org/>

[3] - A Beginner’s Guide to npm — the Node Package Manager, Michael Wanyoike & Peter Dierx, 2017 - <https://www.sitepoint.com/beginners-guide-node-package-manager/>

[4] – Create React App sourcecode - <https://github.com/facebook/create-react-app>

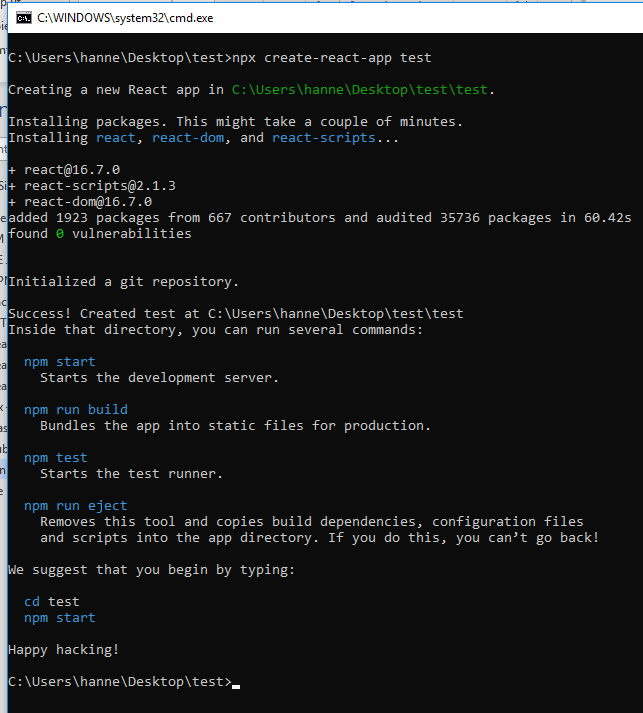
[5] – Facebook Create React App. Docs - <https://facebook.github.io/create-react-app/>

[6] – Atom Flight Manual - <https://flight-manual.atom.io/#the-native-web>

[7] – npm -package-lock.json - https://docs.npmjs.com/files/package-lock.json

[8] - Specifics of npm's package.json handling, 2018 - https://docs.npmjs.com/files/package.json

# Bilagor

[b1] -