# Luodaan React-sovellus ja julkaistaan se Heroku-palvelussa

Web-sovellus ei ole sama asia kuin JavaScriptillä ohjelmoitu koodinpätkä. Web-sovellusta ajetaan selaimella, joten sovelluksessa on html-kielellä rakennettu rakenne, css-tiedostolla tehdyt tyylit. Lisäksi Web-sovellusta varten tulee olla asennettuna web-palvelin ja ohjelmaan konfiguroitu tieto mitä yhteyksiä palvelimen ja sovelluksen välille tarvitaan.

Tässä harjoituksessa teemme React-kirjastoa hyväksi käyttäen Web-Sovelluksen ja ajamme sitä Heroku palvelimessa.

Ennen tehtävän aloittamista

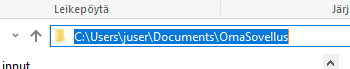
1. Rekisteröi itsellesi Heroku-tili osoitteessa heroku.com ja lataa sieltä heroku-cli.
2. Rekisteröi itsellesi GitHub-tili
3. Asenna Node.js paketti Windowsille

## Luodaan React-sovellus

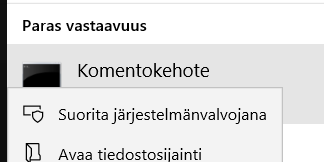
React-sovellus on siis ohjelma, jota voi ajaa selaimella. Voisimme rakentaa sovelluksen alusta asti itse, mutta se vaatisi paljon konfigurointia ja tietämystä ympäristöstä. Helpompi tapa on käyttää valmista ”Pohjaratkaisua” käyttämällä create-react-app -pohjaa. Pohja on yleinen näkemys sovelluksesta ja pyrkii toimimaan jonkinlaisena konfiguraatiostandardina.

Tämän käyttöä varten tarvitsemme vielä pakettienhallintatyökalun nimeltä NPM. Tämän saa helpoiten käyttöön lataamalla ja asentamalla netistä Node.js asennuspaketin, joka siis on virtuaalinen ympäristö JavaScriptin ajamiseen, mutta pitää sisällään NPM-ohjelman, joka osaa siis asentaa ohjelmia avoimen lähdekoodin kirjastosta.

1. Luo kansio ja anna sille nimeksi OmaSovellus
2. Kannattaa kopioida otsikkoriviltä osoite.



1. Avaa cmd-sovellus järjestelmänvalvoja-oikeuksilla   
   (Painia Win-näppäin kirjoita cmd 🡪 Hiiren kakkospainike🡪 Suorita järjestelmän valvojana)



1. Mene luomaasi kansioon cd (liitä leikepöydältä osoite)

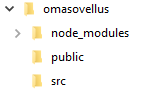


1. Anna komento:

npx install -g create-react-app

create-react-app omasovellus

cd omasovellus

Mitä siis juuri teimme? Käytimme haimme ”pohjan” create-react-app ja loimme sen avulla perussovelluksen. Käytännössä siis websovelluksen esittämistä varten kansiorakenteen ja webbipalvelimen (joka käynnistyessään näyttää sovelluksen selaimessa osoitteessa localhost:3000).

Webbisivun aloitussivu eli **index.html** löytyy kansiosta **public**.

index.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    -->

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>

1. Käynnistä sovellus komennolla:

npm start

Mutta eihän tuo sivu vastaa yhtään tuota koodia?! Homman juoni on Reactin käyttämä jsx-kieli, jossa yhdistetään html-rakenne ja JavaScript yhdeksi kokonaisuudeksi. Kaipaamalle sisältö on sijoitettu lohkoon **<div id=”root”>.** Sisältö sijoitetaan lohkoon, jonka id=”root”.

Sijoittaminen tapahtuu tiedostossa **index.js**, joka löytyy kansiosta **SRC.**

index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import \* as serviceWorker from './serviceWorker';

Import-komennolla tuodaan tiedostoja sovelluksen käyttöön:

* React-tiedossa löytyy reactin komennot
* ReactDOM tiedosto pitää sisällään Document Object Modellin käsittelyyn liittyvät komennot. Voit ajatella DOMin html-tagien muodostamana rakenteena (<body><div> </div></body>)
* index.css pitää sisällään sivun tyylit
* App on nyt se tiedosto joka pitää sisällään ”koodin”, vielä tarkemmin funktiot ja luokat.
* serviceworker liittyy ohjelman sijoittamiseen muistiin, joka mahdollistaa ohjelman käytön huonolla nettiyhteydellä tai offline-tilassa.

ReactDOM.render(<App />, document.getElementById('root'));

* Webbisivun sisältö sijoitetaan sovellukseen käskyllä render (joka löytyy ReactDOM-kirjastosta).
* Rect käyttää jsx-kieltä, joka yhdistää html-merkkausta ja JavaScriptiä. <App /> on itseasiassa funktiokutsu, joka toteuttaa App-funktion. voit mielessäsi muuttaa lauseen muotoon App(). Varsinainen sivun sisältö on siis SRC-kansion App.js -tiedoston App-funktiossa.
* document.getElementById(’root’) Sijoittaa App()-funktion tuoman sisällön html-elementtiin, jonka id=”root”.

App.js

import React from 'react';

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <img src={logo} className="App-logo" alt="logo" />

        <p>

          Edit <code>src/App.js</code> and save to reload.

        </p>

        <a

          className="App-link"

          href="https://reactjs.org"

          target="\_blank"

          rel="noopener noreferrer"

        >

          Learn React

        </a>

      </header>

    </div>

  );

}

export default App;

# Tehtävä – oma sovellus

1. Avaa App.js ja tallenna se nimellä Oma.js
2. Muuta index.js tiedostoa niin että se hakee import-komennolla äsken tekemäsi tiedoston ” Oma.js”.
3. Muut Render-elementtiä niin, että se hakee App-tiedoston sijasta Oma-funktiota.
4. Muuta Funktiota niin, että se tulostaa ”Minun hieno sovellus”

# Sovelluksen julkaiseminen

1. Avaa cmd ja jatka työskentelyä Omasovellus-kansiossa.
2. kirjaudu sisään Herokuun. Komento avaa selaimen, jossa sinun tulee painaa log in -painiketta.:

heroku login

1. Lisää valitse Githubiin siirrettävä koodi kirjoittamalla:

git init

1. Kerro Heroku-palvelulle mitä pakettia verkkosivujen luomiseksi aiomme käyttää.

heroku create -b https://github.com/mars/create-react-app-buildpack.git

1. Siirrä sovelluksen koodi GitHubiin komennolla:

git Add .

1. Kommentoi siirretty koodi

git commit -m "react-create-app on Heroku"

1. Vie sovellus GitHubista Herokun palvelimille

git push heroku master

1. Avaa sovellus

heroku open

Eli koko homma lyhyesti

npm install -g create-react-app

create-react-app my-app

cd my-app

git init

heroku create -b https://github.com/mars/create-react-app-buildpack.git

git add .

git commit -m "react-create-app on Heroku"

git push heroku master

heroku open

<https://dev.to/smithmanny/deploy-your-react-app-to-heroku-2b6f>

<https://medium.com/better-programming/how-to-deploy-your-react-app-to-heroku-aedc28b218ae>

<https://blog.heroku.com/deploying-react-with-zero-configuration>

import React from 'react';

import logo from './logo.svg';

import './App.css';

function Oma() {

  return (

    <div className="Oma">

        <h1>Minun hieno sovellus!</h1>

    </div>

  );

}

export default Oma;

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import Oma from './Oma';

import \* as serviceWorker from './serviceWorker';

ReactDOM.render(<Oma/>, document.getElementById('root'));

serviceWorker.unregister();