



PWA and REACT



Add to home screen



Progressive Web Apps (PWAs) are web apps that use service workers, manifests, and other web-platform features in combination with progressive enhancement to give users an experience on par with native apps.

Prepare react for service worker 1

We will use a template when we create a new project

```
npx create-react-app my-app --template cra-template-pwa
```



Prepare react for service worker 2

In the src/index.js file you change
"unregister()" to "register()";

```
// If you want your app to work offline and load faster, you can change  
// unregister() to register() below. Note this comes with some pitfalls.  
// Learn more about service workers: https://cra.link/PWA  
serviceWorkerRegistration.unregister();
```

Using the REACT PWA template

The service-worker is integrated into production configuration, and it will take care of compiling a service worker file that will automatically precache all of your assets **by default**.

The service worker will use a **cache-first strategy** for handling all requests for webpack-generated assets, including **navigation requests** for your HTML, ensuring that your web app is consistently fast, even on a slow or unreliable network

Building for Relative Paths

By default, Create React App produces a build assuming your app is hosted at the server root.

To override this, specify the `homepage` in your `package.json`, for example:

```
"homepage": "http://mywebsite.com/relativepath",
```

This will let Create React App correctly infer the root path to use in the generated HTML file.

Example:

<https://mortenbonderup.github.io/test/>



1. Run the npm build command

```
npm run build
```

2. Create github repo – like test

Example:

<https://mortenbonderup.github.io/test/>

3. Upload all files in "build" section to the github repo

4. Test your solution on mobile/desktop - online/offline



Assignment

- 1) Create a new REACT app with the *cra-template-pwa*
- 2) Think of a good github repo name, create the github repo
- 3) Make the changes that register the service worker and add homepage key/value to the package.json
- 4) Optional: Design two new icons (192x192) / (512x512) and include them in the project public folder (replace the existing)
- 5) Create at least one REACT page with some content of your own choice. If you want to have multiple pages, think of using router extension. You can copy from one of your RACE projects.
- 6) Run *npm run build* command and upload files/folders from *build* folder.
- 7) Test if app installable on mobile / desktop.

Time estimation: 90 minutes

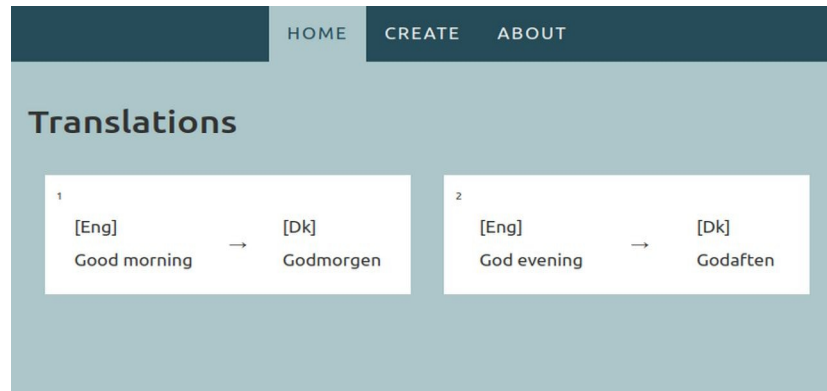
CRUD

CRUD (Create, Read, Update, Delete) is an acronym for ways one can operate on stored data. It is a mnemonic for the four basic functions of persistent storage. CRUD typically refers to operations performed in a database or datastore, but it can also apply to higher level functions of an application such as soft deletes where data is not actually deleted but marked as deleted via a status.

<https://translations-70a25-default-rtdb.europe-west1.firebaseio.com>

`https://translations-70a25-default-rtdb.europe-west1.firebaseio.com/`





- 1) Create new project folder
- 2) From VS code – clone Mortens solution from:
<https://github.com/MortenBonderup/react-translation.git>
- 3) In terminal run "npm install" to install REACT environment
- 4) In terminal, run the app with "npm start".

- 1) Morten explains how the solution works
- 2) Come up with an idea to your own app

Suggestions:

- 1) Home inventory list (item, value etc. (for insurance))
 - 2) Group member app (which person belongs to which group)
 - 3) Home tool overview (screwdriver, drill, grinding machine etc.)
 - 4) Whatever comes to your mind ;-)
- 3) Create a Firebase real time database project.
 - 4) Change Mortens solution so that it fits your idea

Time estimation: 60 – 90 minutes