Management of **React** app files

When creating a **React** app with the **create-react-app** tool, the resulting app – which is completely defined by the files in the app folder – will consist of about 30,000 files, using up 200+ Mb of disk space. This is due to the huge amount of packages being included. Fortunately, we do not need to include all these files when distri­buting **React** projects. The two files **package.json** and **package-lock.json** contain the complete infor­ma­tion about package dependencies. This makes it possible to create a very small “stripped” version of a **React** project.

We consider two ways of distributing and obtaining **React** applications:

* Using a GitHub repository
* Using a **.zip** file

# React apps and GitHub

When a **React** app is created with **create-react-app**, the resulting app will also contain a **.gitignore** file. This means that the creator of the app can simply push the app folder to GitHub; the **.gitignore** file ensures that only the essential (i.e. non-recreatable) elements of the app are actually pushed to GitHub. This turns out to be the following elements in the app folder:

* The **src** and **public** subfolders (i.e. the elements added manually to the project)
* The files **package.json** and **package-lock.json**, which contain the app dependencies
* The files **.gitignore** and **README.md**, which are GitHub-specific

The “consumer” of the app – e.g. a student who needs to work with the app – will then simply obtain the app from GitHub, either through an IDE like Visual Studio Code or through another GitHub client. Once the app has been retrieved, the consumer then needs to perform one additional step in order to use the app, as compared with the usual step for starting the development server:

1. Open a terminal window in the app folder
2. Run **npm install** without any parameters. This will prompt **NPM** to retrieve all the packages on which the app depends, by using the information in the **package…** files.
3. When the installation has finished, the full version of the app has now been recreated, and the development server can again be started as usual by running **npm start**

# React apps and .zip files

The procedure is fairly similar when using a **.zip** file for distribution instead. The creator of an app can e.g. make a copy of the complete app folder, and then in the copied folder strip away everything except:

* The **src** and **public** subfolders (i.e. the elements added manually to the project)
* The files **package.json** and **package-lock.json**, which contain the app dependencies

This stripped version of the app folder can then be zipped into a **.zip** file, and then be made available as e.g. a download from a website. The consumer of the app will then need to retrieve the **.zip** file, unzip it, and perform the exact same steps as described above: In the app folder, first run **npm install** without any para­meters, and then run **npm start**.