

Installing Packages to Use React with Apollo GraphQL

Finally, in this lesson, we will install packages that required to use React with Apollo GraphQL

For this application, no elaborate React setup is needed. Simply use [create-react-app](#) to create your React application. If you want to have an elaborate React setup instead, see this [setup guide for using Webpack with React](#). To get started, the following steps have to be performed:

- Create a new React application with create-react-app
- Create a folder/file structure for your project (recommendation below)

You can create your own folder and file structure for your own components in the *src/* folder; the following top-level structure is only a recommendation. If you adjust it to your own needs, keep in mind that the JavaScript import statements with their paths will need to be adjusted to match. If you don't want to create everything, you can clone this [GitHub repository](#) instead and follow its installation instructions.

- App/
 - index.js
- Button/
- Error/
- FetchMore/
- Input/
- Issue/
 - IssueList/
 - IssueItem/
 - index.js
- Link/
- Loading/
- Organization/

- Profile/
- Repository/
 - RepositoryList/
 - RepositoryItem/
 - index.js
- TextArea/
- constants/
 - routes.js
- index.js
- serviceWorker.js
- style.css

The Apollo Client setup can be completed in the top-level `src/index.js` file, where the React to HTML entry point exists as well. First, install the Apollo Client in your project folder using the command line:

```
npm install apollo-client --save
```



Secondly, install the `apollo-cache-inmemory` and the `apollo-link-http` for mandatory configurations which are used to create the Apollo Client.

```
npm install apollo-cache-inmemory apollo-link-http -- save
```



We further need to install `graphql` and `graphql-tag` for Apollo Client to work with GraphQL.

```
npm install graphql graphql-tag --save
```



Next, we install the library `react-apollo` to connect the Apollo Client with React.

```
npm install react-apollo --save
```



To implement error handling on an application level, we will install another Apollo package, namely `apollo-link-error`.

```
npm install apollo-link-error --save
```



To combine multiple links in your application for using with the Apollo Client instance, we can download the `apollo-link` package which makes link composition possible.

```
npm install apollo-link --save
```



Install `react-router-dom` to show two separate pages for your application.

```
npm install react-router-dom --save
```



Install the package `recompose` which implements many higher-order components and we will use it for server-side filtering.

```
npm install recompose --save
```

