Friday, June 11, 2021 4:02 PM

- **npm i react react-router react-dom --save**

  - *react* -> React is a JavaScript library for creating user interfaces.
    The react package contains only the functionality necessary to define React components. It is typically used together with a React renderer like react-dom for the web, or react-native for the native environments.
    https://www.npmjs.com/package/react

  - *react-router* -> Declarative routing for React.
    https://www.npmjs.com/package/react-router

  - *react-dom* -> This package serves as the entry point to the DOM and server renderers for React. It is intended to be paired with the generic React package, which is shipped as react to npm.
    https://www.npmjs.com/package/react-dom

- **npm i @startupjs/init @startupjs/orm @startupjs/react-sharedb**

  - *Startupjs/init* -> Init sharedb connection on client web, native or server side
    https://www.npmjs.com/package/@startupjs/init

  - *Startupjs/orm* -> ORM system for Racer.js and ShareDB
    https://www.npmjs.com/package/@startupjs/orm

  - *Startupjs/react-sharedb* -> Run sharedb in react
    https://www.npmjs.com/package/@startupjs/react-sharedb

- **npm i sharedb sharedb-access sharedb-hooks sharedb-mongo sharedb-redis-pubsub sharedb-wsbus-pubsub --save**

  - *ShareDB* -> ShareDB is a realtime database backend based on Operational Transformation (OT) of JSON documents. It is the realtime backend for the DerbyJS web application framework.
    https://www.npmjs.com/package/sharedb

  - *ShareDB-access* -> Using sharedb-access you can control create, read, update, and delete database operation for every collection. You can use two types of rules: allow and deny. By default all the operations are denied. So, you should add some rules to allow them. If at least one allow-rule allows the write, and no deny-rules deny the write, then the write is allowed to proceed.
    If you use nodejs that doesn't support async/await you need sharedb-access@3.0.0
    https://www.npmjs.com/package/sharedb-access

  - *ShareDB-hooks* -> The way to hook db-interactions on the serverFrom
    https://www.npmjs.com/package/sharedb-hooks

  - *ShareDB-mongo* -> MongoDB database adapter for sharedb. This driver can be used both as a snapshot store and oplog.
    https://www.npmjs.com/package/sharedb-mongo

  - *ShareDB-redis-pubsub* -> Redis pub/sub adapter adapter for ShareDB.
    This ShareDB add-on gives you horizontal scalability; the ability to have a cluster of multiple server nodes rather than just a single server. Using this adapter, clients can connect to any machine in your cluster, and the ops they submit will be forwarded clients connected through other nodes, and there will be no race conditions with regard to persistence.
    https://www.npmjs.com/package/sharedb-redis-pubsub

  - *ShareDb-wsbus-pubsub* -> ShareDb PubSub module based on ws-bus
    https://www.npmjs.com/package/sharedb-wsbus-pubsub

- **npm i racer racer-highway --save**

  - *Racer* -> Racer is a realtime model synchronization engine for Node.js. By leveraging ShareDB, multiple users can interact with the same data in realtime via Operational Transformation, a sophisticated conflict resolution algorithm that works in realtime and with offline clients.

ShareDB also supports PubSub across multiple servers for horizontal scaling. Clients can express data subscriptions and fetch es in terms of queries and specific documents, so different clients can be subscribed to different overlapping sets of data. On top of this sophisticated backend, Racer provides a simple model and event interface for writing application logic.
https://www.npmjs.com/package/racer

  - *Racer-highway* -> Transport plugin for Racer. It uses pure websockets and fallbacks to browserchannel in the case of old browsers and proxy errors. If you only need to use websockets try racer-ws.
    https://www.npmjs.com/package/racer-highway

- **npm i express axios bluebird --save**

  - *Express* -> The Express philosophy is to provide small, robust tooling for HTTP servers, making it a great solution for single page applications, web sites, hybrids, or public HTTP APIs..
    https://www.npmjs.com/package/express

  - *Bluebird* -> Bluebird is a fully featured promise library with focus on innovative features and performance.
    Promises in Node.js 10 are significantly faster than before. Bluebird still includes a lot of features like cancellation, ite ration methods and warnings that native promises don't. If you are using Bluebird for performance rather than for those - please consider giving native promises a shot and running the benchmarks yourself.
    https://www.npmjs.com/package/bluebird

  - *Axios* -> Promise based HTTP client for the browser and node.js
    https://www.npmjs.com/package/axios

- **npm i mingo --save**

  - *Mingo* -> MongoDB query language for in-memory objects
    https://www.npmjs.com/package/mingo

- **npm i webpack webpack-cli webpack-dev-server --save-dev**

  - *Webpack* -> Webpack is a module bundler. Its main purpose is to bundle JavaScript files for usage in a browser, yet it is also capable of transforming, bundling, or packaging just about any resource or asset.
    https://www.npmjs.com/package/webpack

  - *Webpack-cli* -> Webpack CLI provides a flexible set of commands for developers to increase speed when setting up a custom webpack project. As of webpack v4, webpack is not expecting a configuration file, but often developers want to create a more custom w ebpack configuration based on their use-cases and needs. webpack CLI addresses these needs by providing a set of tools to improve the setup of custom webpack configuration.
    https://www.npmjs.com/package/webpack-cli

  - *Webpack-dev-server* -> Use webpack with a development server that provides live reloading. This should be used for development only.
    It uses webpack-dev-middleware under the hood, which provides fast in-memory access to the webpack assets.
    https://www.npmjs.com/package/webpack-dev-server

- **npm i webpack-merge webpack-plugin-serve clean-webpack-plugin assets-webpack-plugin css-minimizer-webpack-plugin --save-dev**

  - *webpack-merge* -> webpack-merge provides a merge function that concatenates arrays and merges objects creating a new object. If functions are encountered, it will execute them, run the results through the algorithm, and then wrap the returned values wit hin a function again.
    This behavior is particularly useful in configuring webpack although it has uses beyond it. Whenever you need to merge config uration objects, webpack-merge can come in handy.
    https://www.npmjs.com/package/webpack-merge

  - *webpack-plugin-serve* -> A Webpack development server in a plugin.
    https://www.npmjs.com/package/webpack-plugin-serve

  - *clean-webpack-plugin* - > A webpack plugin to remove/clean your build folder(s).
    By default, this plugin will remove all files inside webpack's output.path directory, as well as all unused webpack assets after every successful rebuild.
    https://www.npmjs.com/package/clean-webpack-plugin

  - *assets-webpack-plugin* -> Webpack plugin that emits a json file with assets paths.

https://www.npmjs.com/package/assets-webpack-plugin

  - *css-minimizer-webpack-plugin*(Older plugin:optimize-css-assets-webpack-plugin) -> This plugin uses cssnano to optimize and minify your CSS. Just like optimize-css-assets-webpack-plugin but more accurate with source maps and assets using query string, allows to cache and works in parallel mode.
    https://www.npmjs.com/package/css-minimizer-webpack-plugin

- **npm i @pmmmwh/react-refresh-webpack-plugin react-refresh eslint-plugin-react --save-dev**

  - *@pmmmwh/react-refresh-webpack-plugin* -> An EXPERIMENTAL Webpack plugin to enable "Fast Refresh" (also previously known as Hot Reloading) for React components.
    https://www.npmjs.com/package/@pmmmwh/react-refresh-webpack-plugin

  - *react-refresh* -> This package implements the wiring necessary to integrate Fast Refresh into bundlers. Fast Refresh is a feature that lets you edit React components in a running application without losing their state. It is similar to an old feature known as "hot reloading", but Fast Refresh is more reliable and officially supported by React.
    This package is primarily aimed at developers of bundler plugins. If you’re working on one, here is a rough guide for Fast Refresh integration using this package.
    https://www.npmjs.com/package/react-refresh

  - *eslint-plugin-react* -> React specific linting rules for ESLint
    https://www.npmjs.com/package/eslint-plugin-react

- **npm i @babel/core @babel/register @babel/polyfill @babel/preset-env @babel/preset-react babel-eslint babel-loader --save-dev**

  - *@babel/core* -> Babel compiler core.
    https://www.npmjs.com/package/@babel/core

  - @babel/register -> Babel require hook
    https://www.npmjs.com/package/@babel/register

  - *@babel/polyfill* (Deprecated)-> Babel includes a polyfill that includes a custom regenerator runtime and core-js.
    This will emulate a full ES2015+ environment (no < Stage 4 proposals) and is intended to be used in an application rather tha n a library/tool. (this polyfill is automatically loaded when using babel-node).
    This means you can use new built-ins like Promise or WeakMap, static methods like Array.from or Object.assign, instance methods like Array.prototype.includes, and generator functions (provided you use the regenerator plugin). The polyfill adds to the global scope as well as native prototypes like String in order to do this.
    https://www.npmjs.com/package/@babel/polyfill

  - *@babel/preset-env* -> A Babel preset for each environment.
    https://www.npmjs.com/package/@babel/preset-env

  - *@babel/preset-env* -> A Babel preset for all react plugins.
    https://www.npmjs.com/package/@babel/preset-react

  - *babel-eslint* (Deprecated)-> You only need to use babel-eslint if you are using types (Flow) or experimental features not supported in ESLint itself yet. Otherwise try the default parser (you don't have to use it just because you are using Babel).
    https://www.npmjs.com/package/babel-eslint

  - *babel-loader* -> This package allows transpiling JavaScript files using Babel and webpack.
    https://www.npmjs.com/package/babel-loader