**React query**

React query is a library for fetching data in a react application.

**Why we need**

* Since react is a UI library, there is no specific pattern for data fetching
* useEffect hook for data fetching and useState hook to maintain component state like loading, error or resulting data
* if the data is needed throughout the app, we tend to use state management libraries
* most of the state management libraries are good for working with client state. Theme for application / whether a modal is open
* state management libraries are not great for working with asynchronous / server state

**client state vs server state**

**client state**

persisted in our app memory and accessing / updating it is synchronous

**server state**

persisted remotely and requires asynchronous api for fetching / updating

has shared ownership

data can be updated by someone else without our knowledge

ui data may not be in sync with the remote data

challenging when we have to deal with caching, deduping multiple requests for the same data, updating state data in the background, performance optimization etc.

**react query setup**

step 1 new react project using CRA

step 2 setup api endpoint that serves mock data for use in our application

step 3 setup react router and a few routes in the application

step 4 install depency of react query and start fetching

>yarn add react query

step 5 wrap whole app with **QueryClientProvider**

step 6 query client initialize and provide it to provider

import "./App.css";

import { QueryClientProvider, QueryClient } from "react-query";

import AppRoutes from "./routes/AppRoutes";

const queryClient = new QueryClient();

function App() {

  return (

    <QueryClientProvider client={queryClient}>

      <AppRoutes />

    </QueryClientProvider>

  );

}

export default App;