The stale property in React Query is used to determine whether the data in a query result is stale or not.

When you make a query request using React Query, the library caches the data returned by the server. When you request the same data again, React Query checks if the cached data is still valid or if it has become stale.

If the data is stale, React Query can either return the stale data immediately or refetch the data from the server. The stale property is a boolean value that indicates whether the cached data is stale or not.

You can use the stale property in your application to determine whether to display cached data or to refetch the data from the server. For example, you can display cached data immediately if it's not stale, and then refetch the data in the background to update the cache if it is stale.

In summary, the stale property in React Query is used to check if the cached data is still valid or if it has become stale. You can use this property to decide whether to display cached data or to refetch the data from the server.

By Default Time is 0 sec.

import React from 'react'

import { useQuery } from 'react-query'

import axios from 'axios'

const fetchData=()=>{

  return axios.get('http://localhost:4000/superheroes')

}

export const RQSuperHeroesPage = () => {

  //Step 1 : Increase parameter to hold other valyer

  const{isLoading, data,isError, error}=useQuery('rqsuper-heroes',fetchData,

  {

    staleTime:5000,

  })

    if(isLoading){

      return <h2>Loading......</h2>

    }

// step 2 : Add condition to check error or not

    if(isError){

      return <h2>{error.message}</h2>

    }

      return (

        <div>

          RQSuperHero Page

          {data?.data.map((hero)=>{

            return <div key={hero.name}>{hero.name}</div>

          })}

        </div>

      )

    }