Without React Query Handling Error :

Step 1 : Create prop to store error

  //Step 1: Error Handling- Create prooperty

  const [error,setError]=useState('')

Step 2 : Add catch exception

 useEffect(()=>{

  axios.get('http://localhost:4000/superheroes1').then((res)=>{

    setData(res.data)

    setIsLoading(false)

     //Step 2: Error Handling- Add catch

  }).catch(error=>{

    setError(error.message)

    setIsLoading(false)

  })

},[])

Step 3 : Add Condition

//step 3 : Add condition

  if(error){

    return <h4>{error}</h4>

  }

Whole Code :

import React from 'react'

import { useState,useEffect } from 'react'

import axios from 'axios'

export const SuperHeroesPage = () => {

  const [isLoading,setIsLoading]=useState(true)

  const [data,setData]=useState([])

  //Step 1: Error Handling- Create prooperty

  const [error,setError]=useState('')

  useEffect(()=>{

  axios.get('http://localhost:4000/superheroes1').then((res)=>{

    setData(res.data)

    setIsLoading(false)

     //Step 2: Error Handling- Add catch

  }).catch(error=>{

    setError(error.message)

    setIsLoading(false)

  })

},[])

  if(isLoading){

    return <h2>Loading......</h2>

  }

  //step 3 : Add condition

  if(error){

    return <h4>{error}</h4>

  }

    return (

      <div>

        SuperHero Page

        {data.map((hero)=>{

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

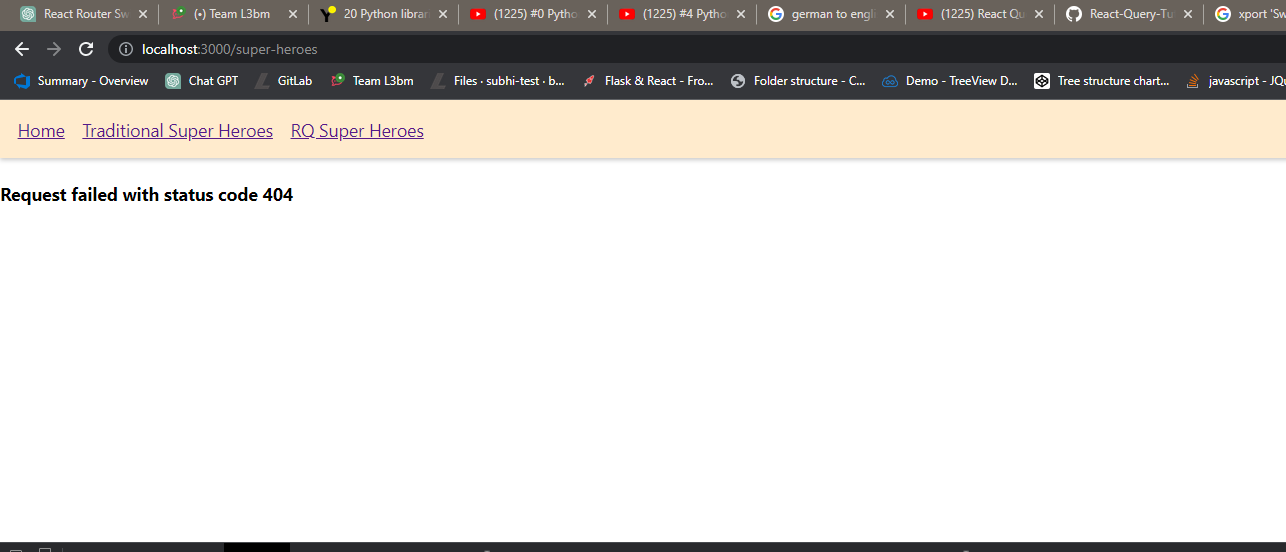
        })}

      </div>

    )

  }

Output :



Error Handling with react query :

  //Step 1 : Increase parameter to hold other valyer

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

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

    if(isError){

      return <h2>{error.message}</h2>

    }

Whole Code :

import React from 'react'

import { useQuery } from 'react-query'

import axios from 'axios'

const fetchData=()=>{

  return axios.get('http://localhost:4000/superheroes1')

}

export const RQSuperHeroesPage = () => {

  //Step 1 : Increase parameter to hold other valyer

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

    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>

      )

    }

