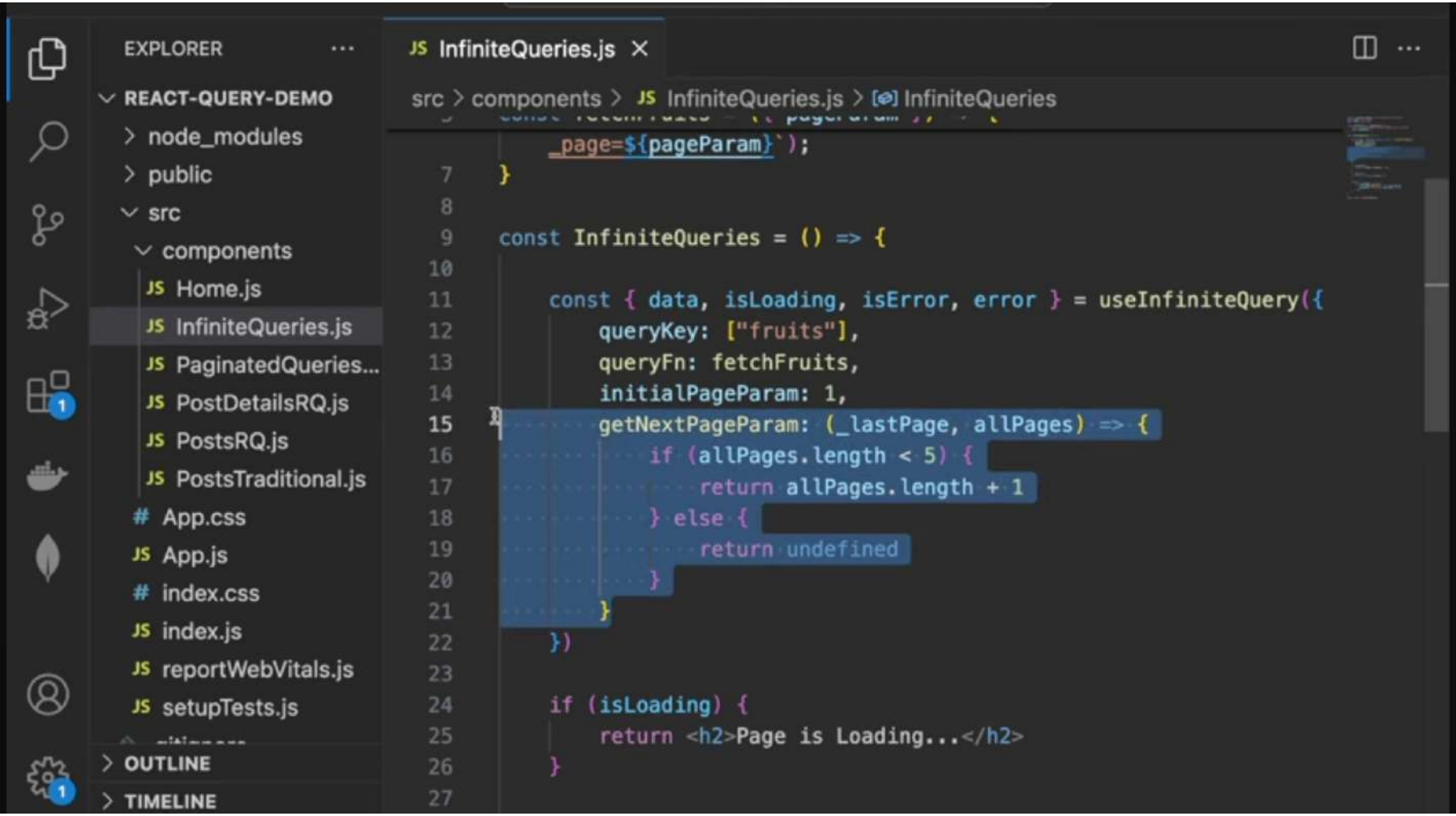
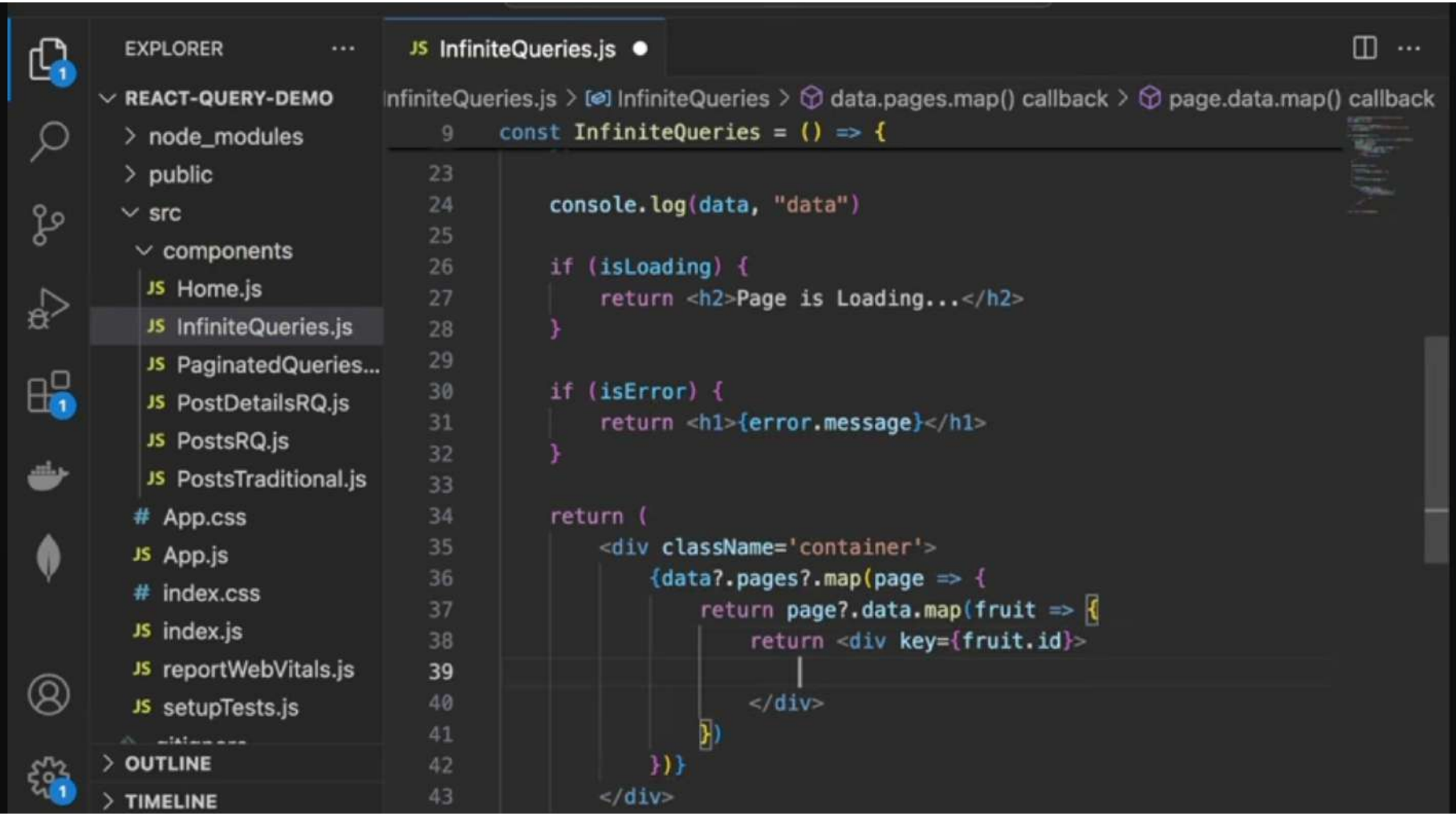
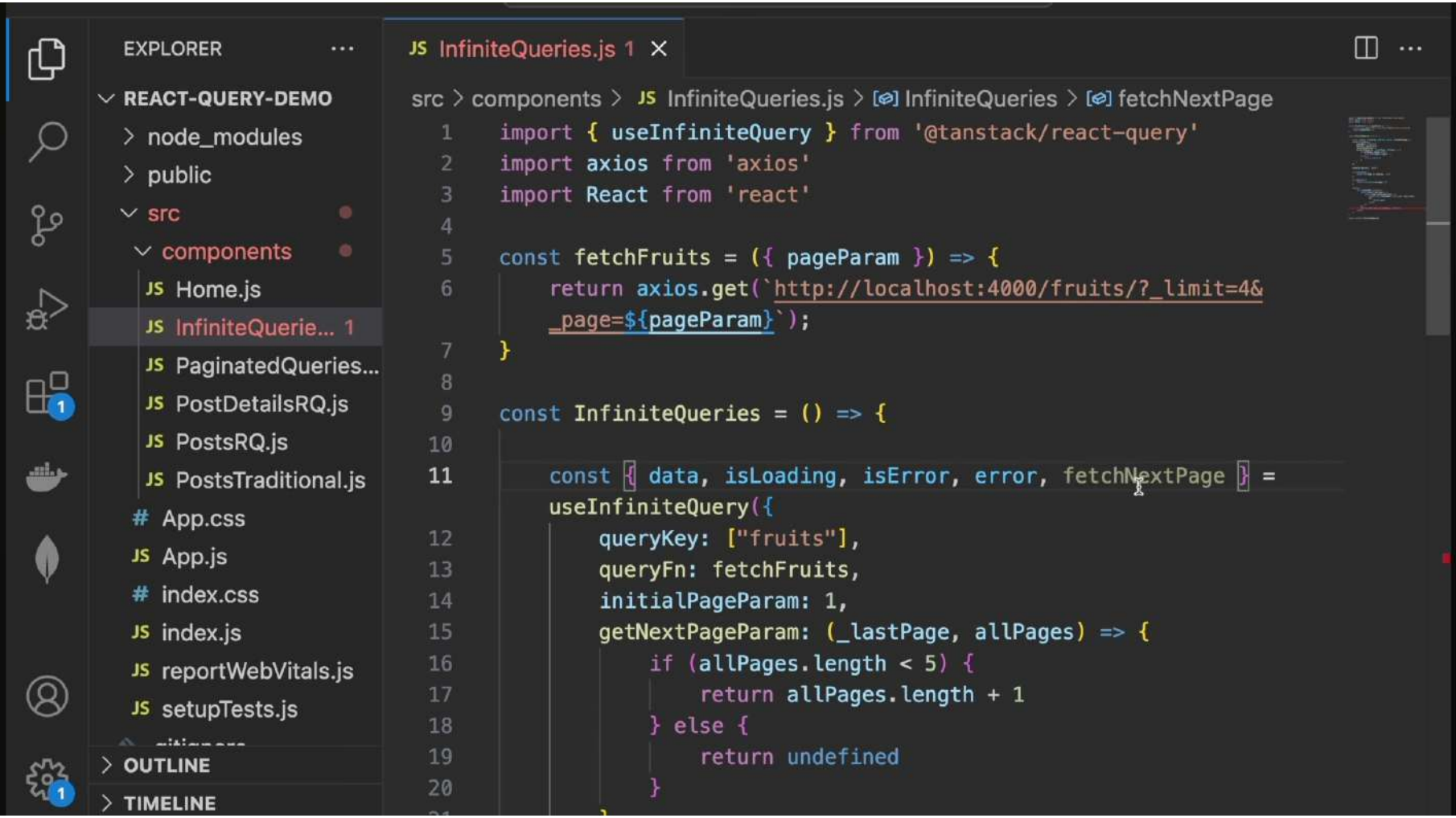
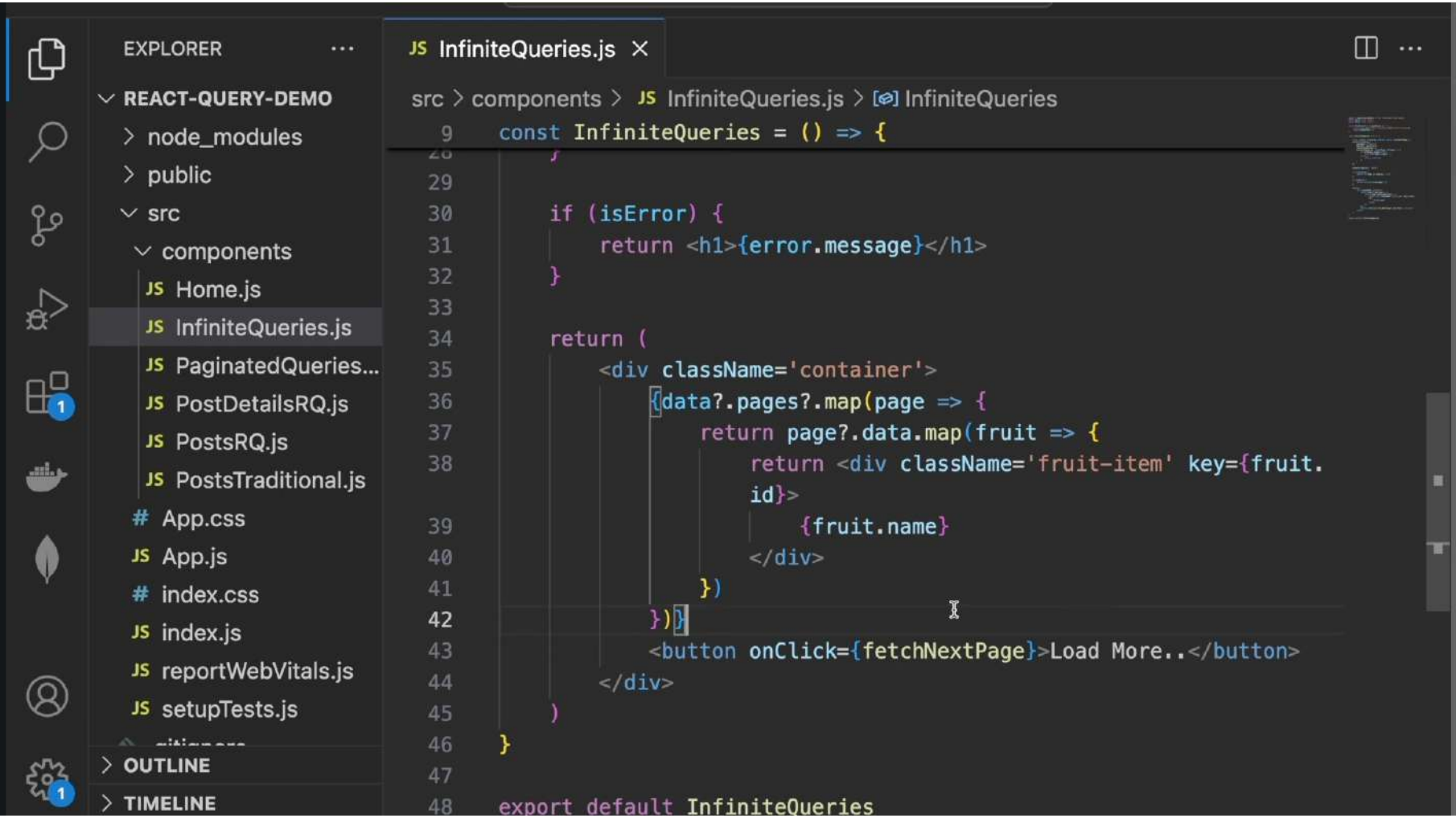

EXPLORER
REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQueries.js
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
InfiniteQueries.js
src > components > InfiniteQueries.js > InfiniteQueries
_page=${pageParam}`);
}
const InfiniteQueries = () => {
const { data, isLoading, isError, error } = useInfiniteQuery({
queryKey: ["fruits"],
queryFn: fetchFruits,
initialPageParam: 1,
getNextPageParam: (_lastPage, allPages) => {
if (allPages.length < 5) {
return allPages.length + 1
} else {
return undefined
}
}
})
if (isLoading) {
return <h2>Page is Loading...</h2>
}


EXPLORER
REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQueries.js
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
InfiniteQueries.js
InfiniteQueries.js > InfiniteQueries > data.pages.map() callback > page.data.map() callback
const InfiniteQueries = () => {
console.log(data, "data")
if (isLoading) {
return <h2>Page is Loading...</h2>
}
if (isError) {
return <h1>{error.message}</h1>
}
return (

{data?.pages?.map(page => {
return page?.data.map(fruit => {
return

})
})}

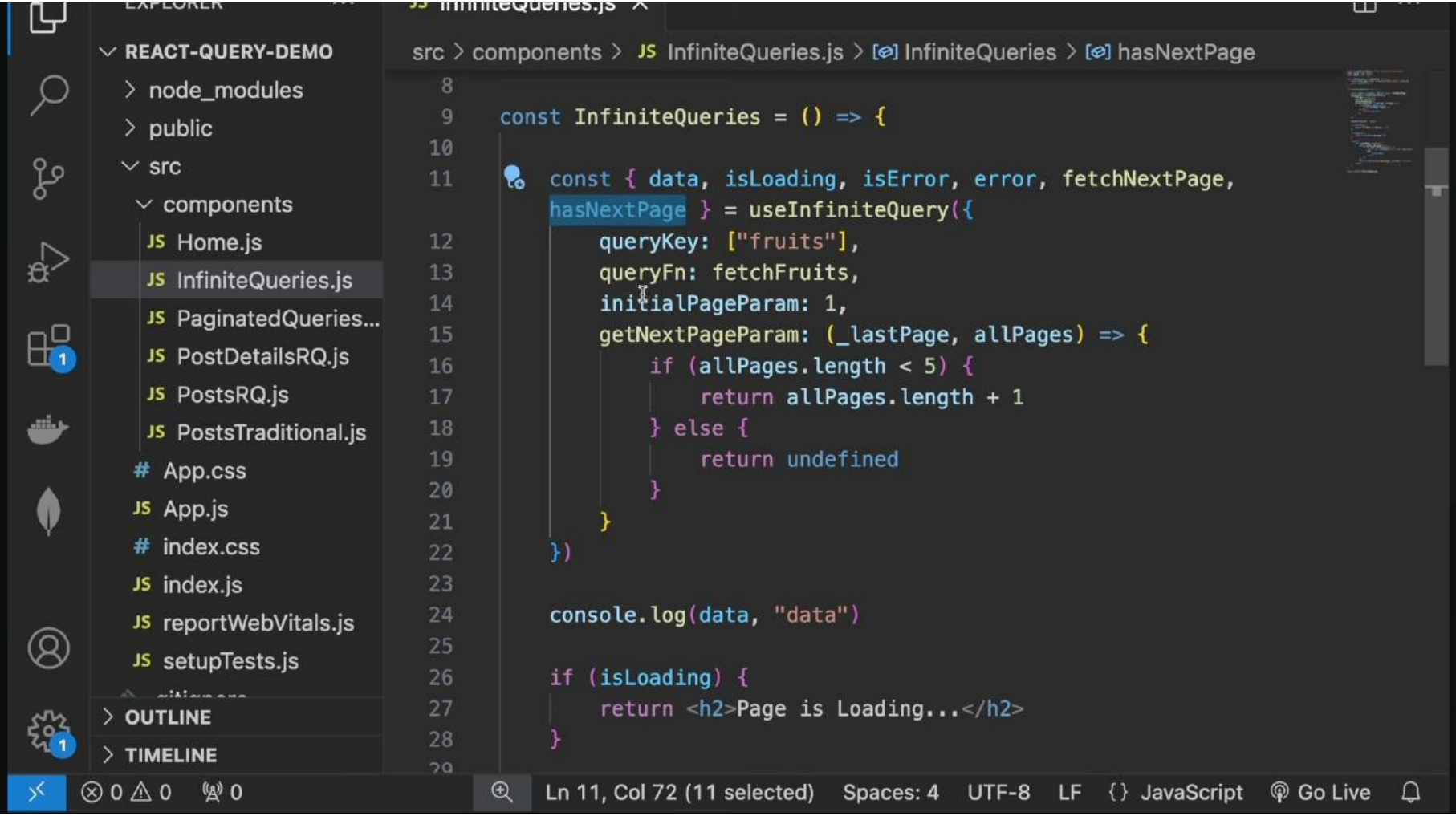

EXPLORER
REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQuerie... 1
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
InfiniteQueries.js 1
src > components > InfiniteQueries.js > InfiniteQueries > fetchNextPage
import { useInfiniteQuery } from '@tanstack/react-query'
import axios from 'axios'
import React from 'react'
const fetchFruits = ({ pageParam }) => {
return axios.get(`http://localhost:4000/fruits/?_limit=4&
_page=${pageParam}`);
}
const InfiniteQueries = () => {
const { data, isLoading, isError, error, fetchNextPage } =
useInfiniteQuery({
queryKey: ["fruits"],
queryFn: fetchFruits,
initialPageParam: 1,
getNextPageParam: (_lastPage, allPages) => {
if (allPages.length < 5) {
return allPages.length + 1
} else {
return undefined
}

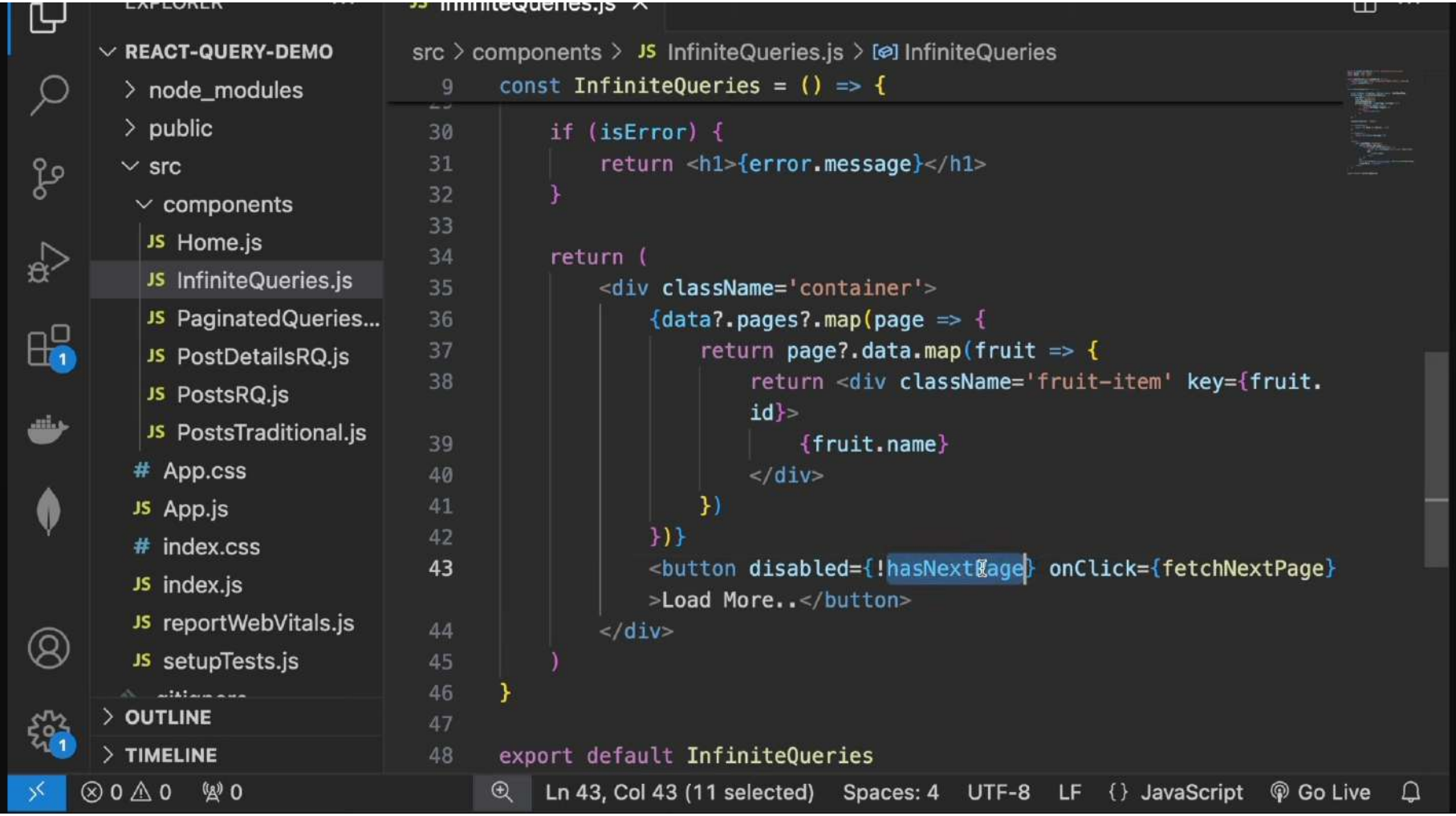

EXPLORER
REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQueries.js
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
InfiniteQueries.js
src > components > InfiniteQueries.js > InfiniteQueries
const InfiniteQueries = () => {
if (isError) {
return <h1>{error.message}</h1>
}
return (

{data?.pages?.map(page => {
return page?.data.map(fruit => {
return <div className='fruit-item' key={fruit.
id}>
{fruit.name}

})
})}
<button onClick={fetchNextPage}>Load More..</button>

)
}
export default InfiniteQueries


REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQueries.js
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
src > components > InfiniteQueries.js > InfiniteQueries > hasNextPage
const InfiniteQueries = () => {
const { data, isLoading, isError, error, fetchNextPage,
hasNextPage } = useInfiniteQuery({
queryKey: ["fruits"],
queryFn: fetchFruits,
initialPageParam: 1,
getNextPageParam: (_lastPage, allPages) => {
if (allPages.length < 5) {
return allPages.length + 1
} else {
return undefined
}
}
})
console.log(data, "data")
if (isLoading) {
return <h2>Page is Loading...</h2>
}
Ln 11, Col 72 (11 selected)
Spaces: 4
UTF-8
LF
JavaScript
Go Live


REACT-QUERY-DEMO
node_modules
public
src
components
Home.js
InfiniteQueries.js
PaginatedQueries...
PostDetailsRQ.js
PostsRQ.js
PostsTraditional.js
App.css
App.js
index.css
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
src > components > InfiniteQueries.js > InfiniteQueries
const InfiniteQueries = () => {
if (isError) {
return <h1>{error.message}</h1>
}
return (

{data?.pages?.map(page => {
return page?.data.map(fruit => {
return <div className='fruit-item' key={fruit.
id}>
{fruit.name}

})
})}
<button disabled={!hasNextPage} onClick={fetchNextPage}
>Load More..</button>

)
}
export default InfiniteQueries
Ln 43, Col 43 (11 selected)
Spaces: 4
UTF-8
LF
JavaScript
Go Live