

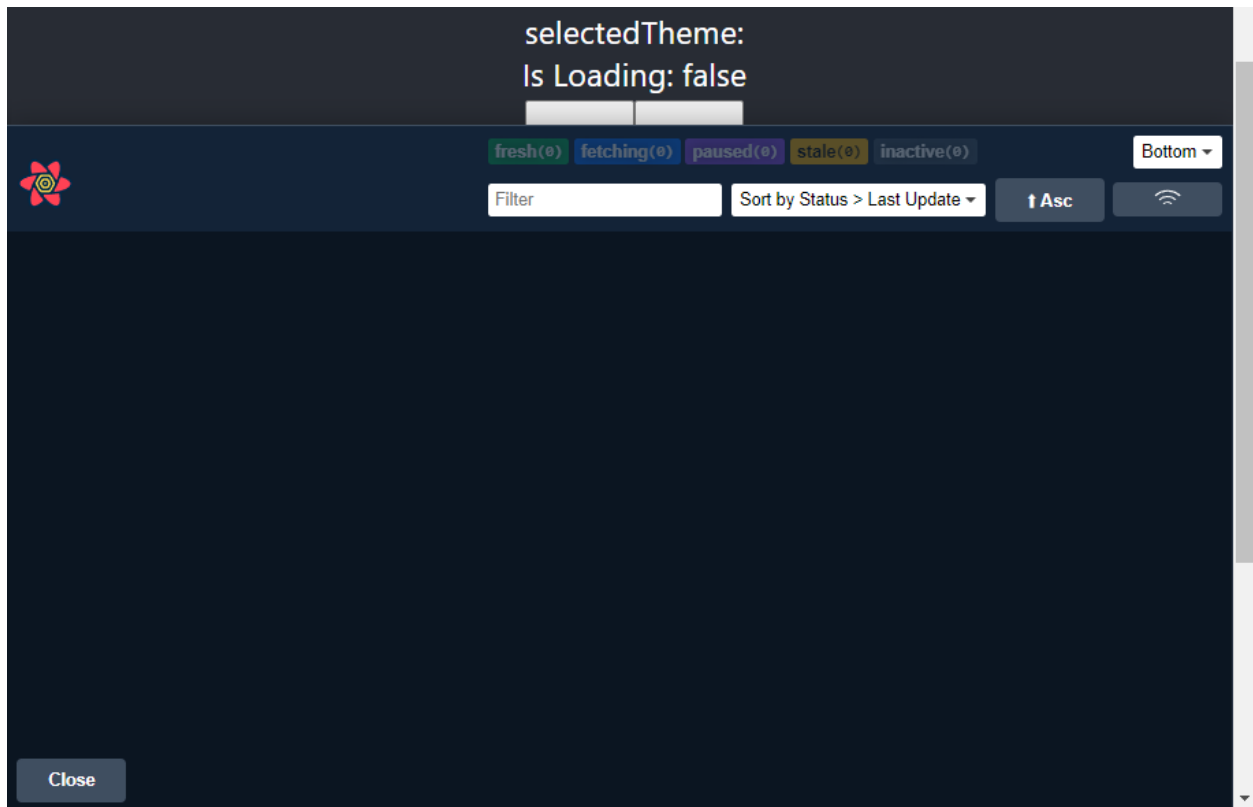
Chapter 1: What Is State and How Do We Manage It?

//No images...

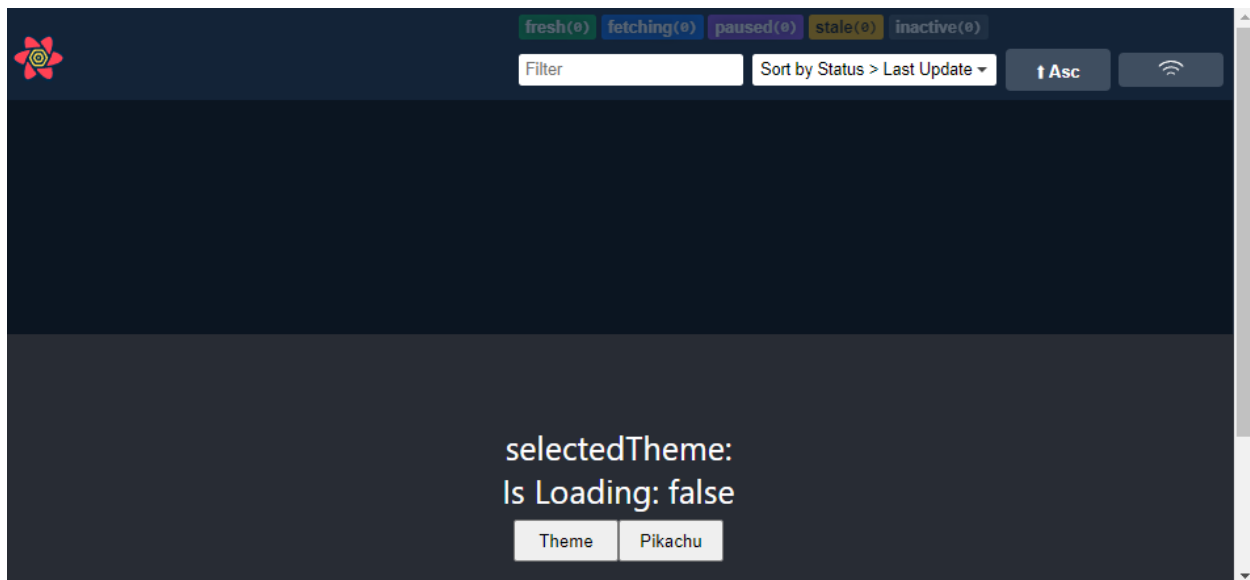
Chapter 2: Server State versus Client State

//No images...

Chapter 3: React Query - Introducing, Installing, and Configuring it



```
▼ <div id="root" style="padding: 0px 0px 500px;">
...  ▶ <aside class="ReactQueryDevtools" aria-label="React Query Devtools">...</aside> == $0
      ▶ <div class="App">...</div> flex
    </div>
```

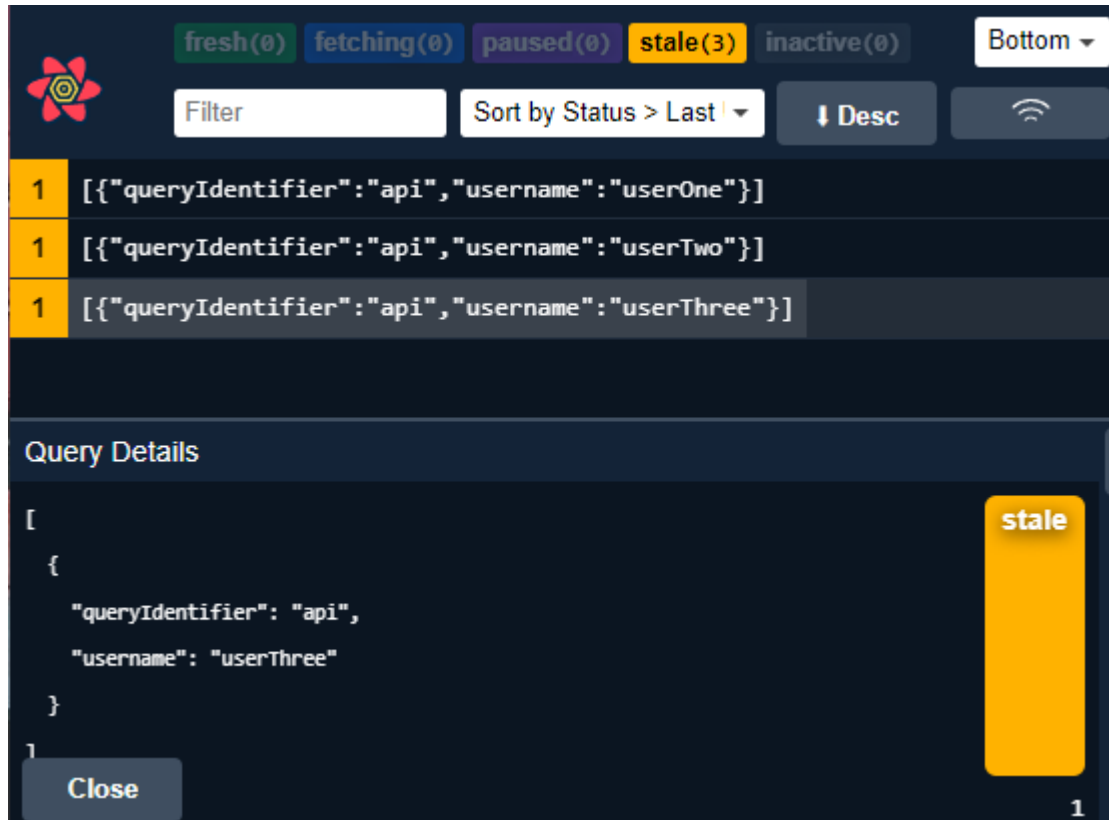


```
▼ <div id="root">
  ▶ <div class="ReactQueryDevtoolsPanel" aria-label="React Query Devtools Panel" id="ReactQuery
    DevtoolsPanel" style="font-size: 0.9em; font-family: sans-serif; display: flex; background-
    color: rgb(11, 21, 33); color: white; height: 500px; position: relative; flex-direction: co
    lumn;">...</div> flex == $0
  ▶ <div class="App">...</div> flex
</div>
```

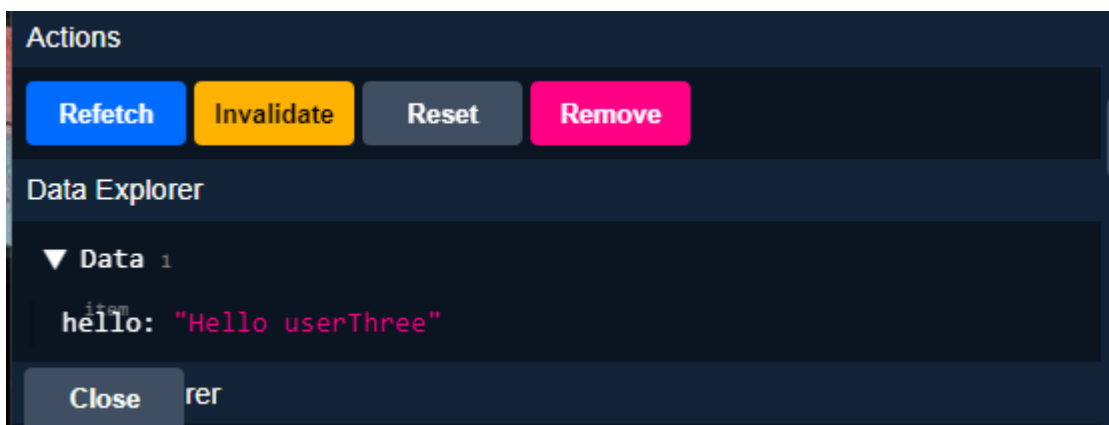
Chapter 4: Fetching Data with React Query

//No Images...

Chapter 5: More Data-Fetching Challenges



The screenshot shows the Redux DevTools component inspector for a component. At the top, there's a status bar with buttons: `fresh(0)`, `fetching(0)`, `paused(0)`, `stale(3)`, and `inactive(0)`. A dropdown menu shows `Bottom`. Below this is a search bar labeled `Filter`, a sort dropdown labeled `Sort by Status > Last`, and a button labeled `Desc`. The main list shows three items, each with a status of `1` and a query object: `[{"queryIdentifier": "api", "username": "userOne"}]`, `[{"queryIdentifier": "api", "username": "userTwo"}]`, and `[{"queryIdentifier": "api", "username": "userThree"}]`. Below the list is a `Query Details` section showing a JSON object: `{ "queryIdentifier": "api", "username": "userThree" }`. A yellow `stale` badge is visible on the right side of the details view. A `Close` button is at the bottom left, and a `1` badge is at the bottom right.



The screenshot shows the Redux DevTools component inspector for a component. At the top, there's an `Actions` section with buttons: `Refetch`, `Invalidate`, `Reset`, and `Remove`. Below this is a `Data Explorer` section showing a tree view with a `Data` node containing a `hello` property with the value `"Hello userThree"`. A `Close` button is at the bottom left, and a `rer` button is at the bottom right.

Query Explorer

▼ Query 17

- items
abortSignalConsumed: false
- defaultOptions: null
- ▶ options
- 7 items
meta: null
- cacheTime: 300000
- ▶
- observers
cache 5
- 1 item
items
▶ logger
- 25 items
▼
- queryKey 1
▼ 0 2 items
item
queryIdentifier: "api"



fresh(0)

fetching(1)

paused(0)

stale(2)


inactive(0)

Bottom ▼

Filter

Sort by Status > Last ▼

↓ Desc



1	[{"queryIdentifier": "api", "username": "userOne"}]
1	[{"queryIdentifier": "api", "username": "userTwo"}]
1	[{"queryIdentifier": "api", "username": "userThree"}]

Query Details

Actions

Refetch

Invalidate

Reset

Remove

Data Explorer

▼ Data 1

Close

"Hello userTwo"

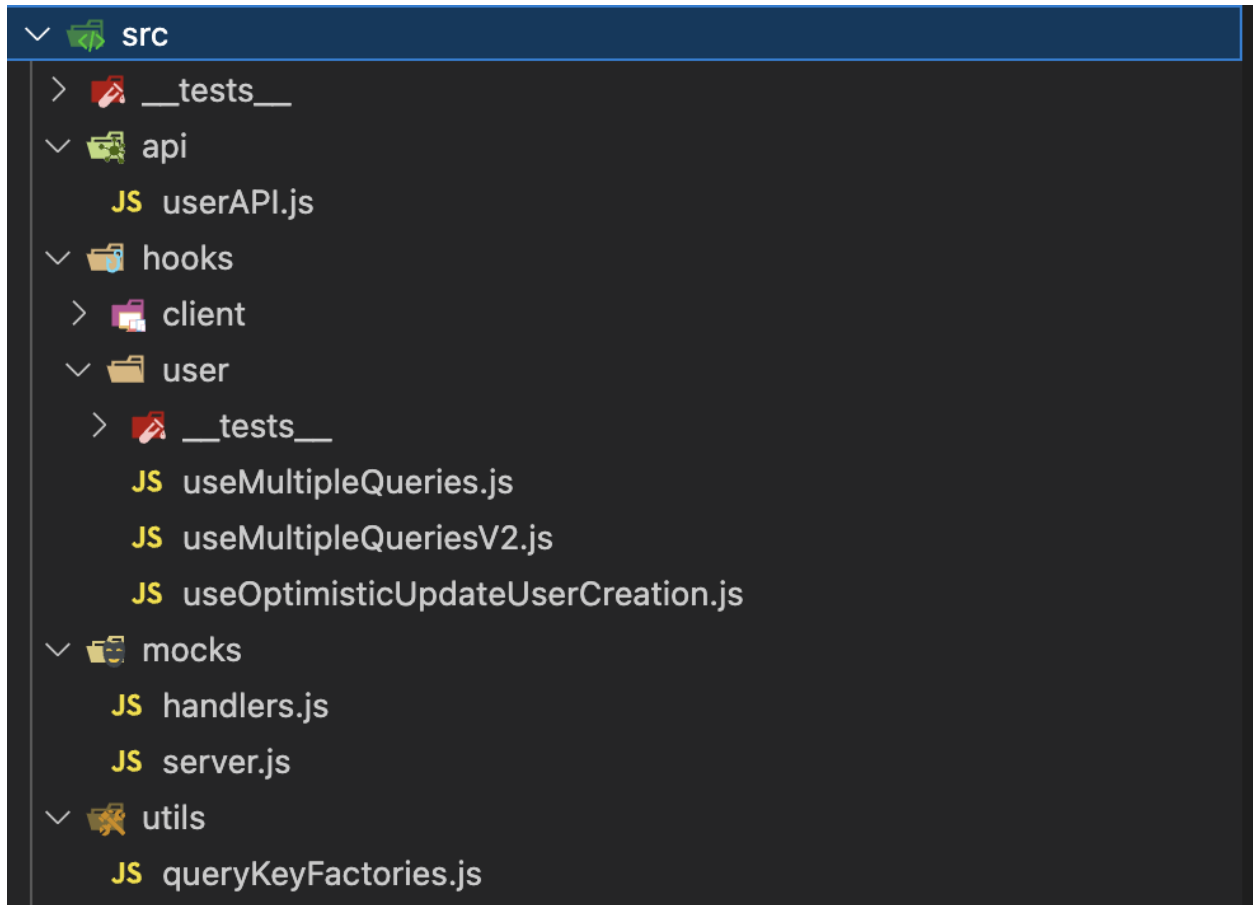
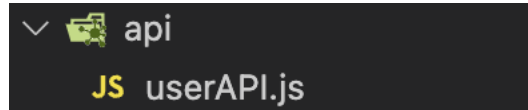
Chapter 6: Performing Data Mutations with React Query

//No images...

Chapter 7: Server-Side Rendering with Next.js or Remix

//No images...

Chapter 8: Testing React Query Hooks and Components



Chapter 9: What Changes in React Query v5?

//No images...