

useEffect for Service Calls debate

A topic of much debate in 2022

- You CAN use useEffect to load service calls
 - Many did/do
 - It ensures data for component will load
- 2022 raised debate that you mostly SHOULDN'T

Why NOT to use useEffect for service calls?

- Lots of edge cases to consider
- If component unloads while fetching?
 - State update is error
- Always triggers a render of empty component
 - With follow up rerender with data.
- What if you want to retry?
 - Have to write event trigger anyway

Debate generated a lot of heat

This debate was frustrating for many

- What they had worked
 - If they ignored edge cases
 - worked "most of the time" :(
- Little clear guidance in what SHOULD do
 - Concerns React was requiring more libraries
 - Concerns React was demanding more server-side work

For this course

- We CAN use useEffect to call services
- But will only do so when "appropriate"

"Appropriate" means

- rendering IS the trigger
- an event is NOT the trigger

Most service calls will be done with events

When to use useEffect for service calls

Example: On page load

- There is no event trigger
 - Technically a page "load" event
 - But same result
- Root component (Ex: App.jsx) won't "unmount"

An on root component load service call effect

```
useEffect(  
  () => {  
    fetchSession()  
    .then( session => {  
      // use session data to set state  
    })  
    .catch( err => {  
      // use error data to set state  
    });  
  },  
  [] // Only run on first render  
);
```

Will run twice in development!

Key Concepts

- `useEffect` because no user action event trigger
- When response changes state
 - There WILL be a rerender
- Dependency array
 - Don't want it EVERY render
 - Or even more than once
- Response may be slow
 - What to show in meantime?
 - Don't have option to trigger state conflicts
- Should handle errors
 - Even if "silently"

Summary - Most Service calls not done with useEffect

- You CAN
 - But alternatives usually better
- Problems with useEffect service calls include
 - Unmounted components
 - Retrying

Instead most service calls triggered by events

Summary - Initial Page Loads

"Initial Page Load" is an exception

- We will use `useEffect` for these
- No risk of unmounting
- Run only once

Summary - Things to Consider

- Be sure to handle errors
 - Even "silently" for non-problem errors
- Show something while waiting
- Avoid events that would cause state confusion