Making API Call with Hooks | Cheat Sheet

Concepts in Focus

1. Making API Call while using React Hooks

In Function Component, we can use

`Effect Hook` to make an API Call after the component render

API Call using the

`fetch()` method works asynchronously, so we need to write the API Call logic in an **async function**

1.1 Effects should be Synchronous

It's not a good practice to make an effect asynchronous because it may lead to unexpected results

JSX

```jsx
useEffect(async() => {
  await fetch(url, options)
  ...
})
```

If you need an asynchronous function, define and call the async function inside the Effect

JSX

```jsx
useEffect(() => {
  const funName = async() => {
    await fetch(url, options)
    ...
  }
  funName()
})
```

2. Displaying different views

While making an API call, we need to display different views like Loading View, Success View, and Failure View

For displaying these views, we need to maintain the following values in the state

- Status
- Data
- Error Message

We use a single state variable with an **object** as an initial value to store these values as they tend to change together

File: src/components/Leaderboard/index.js

JSX

```jsx
import { useState , useEffect } from "react"
...
const apiStatusConstants = {
  initial: "INITIAL",
  inProgress: "IN_PROGRESS",
  success: "SUCCESS",
  failure: "FAILURE"
}

const Leaderboard = () => {
  const [apiResponse, setApiResponse] = useState({
    status: apiStatusConstants.initial,
    data: null,
    errorMsg: null
  })
  ...
  return (...)
}

export default Leaderboard
```

Collapse

Based on the API Status, we display different views

File: src/components/Leaderboard/index.js

JSX

```
import { useState , useEffect } from "react"
...
const Leaderboard = () => {
  const [apiResponse, setApiResponse] = useState({
    status: apiStatusConstants.initial,
    data: null,
    errorMsg: null
  })

  useEffect(()=>{...})
  ...
  const renderLeaderboard = () => {
    const { status } = apiResponse
    switch (status) {
      case apiStatusConstants.inProgress:
        return renderLoadingView()
      case apiStatusConstants.success:
        return renderSuccessView()
      case apiStatusConstants.failure:
        return renderFailureView()
      default:
        return null;
    }
  }
  ...
  return(...)
}

export default Leaderboard
```

Collapse

3. Leaderboard Application Code

Use the below command to get the Final Code for the Leaderboard Application

SHELL

```
ccbp start RHSIVO39C2
```

MARKED AS COMPLETE

Submit Feedback