

Tracking Element Visibility

This could be used for lazy loading (perhaps on the Node Tree View)

The react docs have an article on [useEffect](#) has an example in the **Examples of connecting to an external system section** that has a code sample of how to detect if an element is on screen. I imagine being able to use this if we needed roll our own lazy loading solution for that would not require that we have a specific height, or if we wanted to have lists nested in other lists (like in the tree view for example where each node has it's own children that we want to load in batches).

The screenshot shows the React Docs interface for the 'Examples of connecting to an external system' section. The navigation bar at the top includes links for 'Connecting to a chat server', '2. Listening to a global browser event', '3. Triggering an animation', '4. Controlling a modal dialog', and '5. Tracking element visibility', which is currently selected. Below the navigation bar, the title 'Example 5 of 5: Tracking element visibility' is displayed. The text explains that the external system is the browser DOM, and the `App` component displays a long list, then a `Box` component, and then another long list. It notes that when the `Box` component is fully visible, the background color changes to black. The implementation uses `useEffect` to manage an `IntersectionObserver`.

```
1 import { useRef, useEffect } from 'react';
2
3 export default function Box() {
4   const ref = useRef(null);
5
6   useEffect(() => {
7     const div = ref.current;
8     const observer = new IntersectionObserver(entries => {
9       const entry = entries[0];
10      if (entry.isIntersecting) {
11        document.body.style.backgroundColor = 'black';
12        document.body.style.color = 'white';
13      } else {
14        document.body.style.backgroundColor = 'white';
15        document.body.style.color = 'black';
16      }
17    });
18    observer.observe(div);
19  });
20 }
```

The right side of the screenshot shows a preview of the `Box` component, which is a white box containing a list of 20 items, each labeled 'Item #X (keep scrolling)'. The background of the preview is black, indicating that the `Box` component is currently visible in the viewport.

Revision #1

Created 24 June 2024 21:28:18 by Quinn Godfrey

Updated 24 June 2024 21:32:47 by Quinn Godfrey