LAZY LOADING AND SUSPENSE

In React Js

SWIPE NEXT

01

What is LAZY LOADING?

Lazy loading delays the loading of components until they are required, reducing the initial load time and improving performance. This is especially useful in large applications where not all components are needed right away.

SWIPE NEXT

EXAMPLE:

```jsx
import React, { lazy, Suspense } from 'react';

const MyComponent = lazy(() => import('./MyComponent'));

function App() {
  return (
    <div>
      <Suspense fallback={<div>Loading...</div>}>
        <MyComponent />
      </Suspense>
    </div>
  );
}

```

SWIPE NEXT

02

How it works:

- **MyComponent is only loaded when it's rendered.**
- **Suspense provides a fallback UI while the component is loading.**

SWIPE NEXT

03

Why Use SUSPENSE?

Suspense allows you to handle loading states more gracefully. While waiting for a component to load, you can show a spinner, a loading message, or any placeholder you choose.

SWIPE NEXT

04

Benefits of Lazy LOADING AND SUSPENSE

- **Improved Performance: By splitting code and loading components on demand, your application becomes more responsive.**
- **Better User Experience: Users don't have to wait for everything to load at once.**
- **Scalability: As your app grows, lazy loading ensures it remains efficient and fast.**

SWIPE NEXT

FINAL THOUGHTS

Incorporating lazy loading and Suspense into your React applications can make a significant difference in performance and user experience. As your app scales, these techniques help maintain speed and efficiency, ensuring a smooth and seamless experience for your users.

SWIPE NEXT

DON'T FORGET TO

Share

ABHIJEET CHAKANE

SWIPE BACK