

React lazy loading example

React lazyloading explain with example

Sure! Here's a simple explanation of **React Lazy Loading** along with an example.

What is React Lazy Loading?

React Lazy Loading is a technique to **load components only when they are needed**, instead of loading everything upfront. This helps improve the initial load time of your app and overall performance by splitting your code into smaller chunks that load on demand.

How React Lazy Loading works?

React provides a built-in function called `React.lazy()` which allows you to dynamically import components. Combined with the `<Suspense>` component, it can show a fallback (like a loading spinner) while the lazy-loaded component is being fetched.

Basic Example

Suppose you have two components:

- `Home.js`
- `About.js`

You want to load `About` only when the user navigates to the About page.

Step-by-step Example

1. Create the components

jsx

```
// Home.js
export default function Home() {
  return <h1>Home Page</h1>;
}
```

```
}

// About.js
export default function About() {
  return <h1>About Page</h1>;
}
```

2. Lazy load the `About` component in your main App

```
jsx

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

// Lazy load About component
const About = lazy(() => import('./About'));
import Home from './Home';

function App() {
  const [page, setPage] = React.useState('home');

  return (
    <div>
      <nav>
        <button onClick={() => setPage('home')}>Home</button>
        <button onClick={() => setPage('about')}>About</button>
      </nav>

      <Suspense fallback=<div>Loading...</div>>
        {page === 'home' && <Home />}
        {page === 'about' && <About />}
      </Suspense>
    </div>
  );
}

export default App;
```

What happens here?

- When the app loads, only `Home` component code is loaded immediately.
 - When the user clicks the **About** button, the `About` component is dynamically imported (lazy-loaded).
 - While loading the `About` component, the fallback UI (`Loading...`) is shown.
 - After loading, the `About` component is rendered.
-

Benefits of React Lazy Loading

- **Reduced initial bundle size:** Only essential code loads upfront.
 - **Better user experience:** Loading indicators can be shown while waiting.
 - **Improved performance:** Faster page loads, especially in large apps.
-

If you want, I can provide an example using React Router lazy loading too! Want me to?

if i have routes, then how can i handle lazyloading

Great question! When you have **routes** in React (typically using **React Router**), you can lazy load route components so that each route's component loads only when you navigate to that route.

How to Handle Lazy Loading with Routes in React Router

Example using React Router v6 and `React.lazy()`

Step 1: Install React Router (if not installed)

```
bash
```

```
npm install react-router-dom@6
```

Step 2: Create your route components (like `Home.js`, `About.js`, `Contact.js`)

jsx

```
// Home.js
export default function Home() {
  return <h1>Home Page</h1>;
}

// About.js
export default function About() {
  return <h1>About Page</h1>;
}

// Contact.js
export default function Contact() {
  return <h1>Contact Page</h1>;
}
```

Step 3: Set up lazy loading for route components inside your main `App.js`

jsx

```
import React, { Suspense, lazy } from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';

// Lazy load route components
const Home = lazy(() => import('./Home'));
const About = lazy(() => import('./About'));
const Contact = lazy(() => import('./Contact'));

function App() {
  return (
    <Router>
      <nav>
        <Link to="/">Home</Link> |{' '}
        <Link to="/about">About</Link> |{' '}
        <Link to="/contact">Contact</Link>
      </nav>
    </Router>
  );
}
```

```

</nav>

{/* Suspense shows fallback UI while loading lazy components */}
<Suspense fallback={<div>Loading page...</div>}>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about" element={<About />} />
    <Route path="/contact" element={<Contact />} />
  </Routes>
</Suspense>
</Router>
);
}

export default App;

```

Explanation:

- Each route component (`Home` , `About` , `Contact`) is imported lazily with `React.lazy` .
- The `<Suspense>` component wraps the `<Routes>` , so **while any route component is being loaded, the fallback UI (`Loading page...`) is shown**.
- This way, the code for each route loads only when the user navigates to that route.

Summary

- Use `React.lazy(() => import('./YourComponent'))` to lazy load route components.
- Wrap your routes inside `<Suspense fallback={...}>` .
- Use React Router `<Routes>` and `<Route>` as usual.

If you want, I can also show how to split lazy loading for nested routes or error boundaries with lazy loading! Just ask.