

LAZY


Loading...

in

React



What is Lazy Loading?



Lazy loading or code splitting is the ability to defer the loading of a chunk of code.

That means, instead of bundling and loading all the code when the page loads, just the needed code will be loaded, and some parts will be loaded whenever it's needed.

When do we need Lazy Loading?

Code splitting is useful for medium size to larger React apps.

Even in some medium-size apps, we don't need lazy loading.

If you have some parts of your app that most of your users won't visit, it makes sense to load it lazily instead of increasing your bundle size and hence, decreasing the performance of your app.

How to lazy load a component?

Import lazy and Suspense from the React library.

Instead of importing the Users component, we use lazy to be able to defer it and import the needed component dynamically when needed.

With that change and the help of Suspense, we can now use the render props provided by React Router to display the component once imported.

if it takes time to load, it will use the fallback to show a loading message.

You're not limited to a simple message you can use a spinner component, loading indicator, etc.

Example:

```
import React, { lazy, Suspense } from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom'

// Normal Components
import NavBar from './components/NavBar';
import Posts from './components/Posts/Posts';

// Code Splitted Component (Added Lazy Loading)
const Users = lazy(() => import('./components/Users/Users'))

function App() {

  return (
    <BrowserRouter>
      <main>
        <NavBar />
        <Switch>
          <Route exact path="/" component={Posts} />
          <Route path="/users" render={() =>
            <Suspense fallback={<h1>Loading....</h1>}>
              <Users />
            </Suspense>
          }/>
        </Switch>
      </main>
    </BrowserRouter>
  );
}

export default App;
```

Benefits of lazy loading React component

- **Faster initial loading**
- **Better User Experience**
- **Less bandwidth consumption**
- **Preserving system resources**
- **Reduced work for the browser**