

Get lazy with React code splitting

Dillon Mulroy | Software Engineer @ Visionist inc

First things first: The Bundle

- Bundlers create one giant file of all your source code concatenated together that is then used to serve your app to clients.
- Your bundle will grow in size as your app grows (both from your own source and the additions of 3rd party code).
- Larger bundles mean longer load times.



What is code splitting and why should you care?

- Code splitting takes your bundle and breaks it into multiple smaller bundles.
- It enables you to “lazily load” your code.
- Faster load times :)
- Increased complexity :(

React.lazy()

- Released in React 16.6
- “React.lazy takes a function that must call a dynamic import(). This must return a Promise which resolves to a module with a default export containing a React component.” - reactjs.org
- Dynamic Import
 - Stage 3 in [TC39](#)
 - Supported by all major bundlers
 - Requires [babel-plugin-syntax-dynamic-import](#)
 - CRA supports it out of the box

React.lazy()

1. Import lazy and Suspense from React.
2. Import your component with lazy() and import()
3. Wrap your component with the <Suspense /> component
4. ???
5. Profit \$\$\$

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

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

const App = () => (
  <div>
    <Suspense fallback={<div>Loading...</div>}>
      <MyLazyComponent />
    </Suspense>
  </div>
);
```

A quick word on <Suspense />

- “Suspense lets components “wait” for something before rendering. Today, Suspense only supports one use case: loading components dynamically with `React.lazy`. In the future, it will support other use cases like data fetching.” - reactjs.org
- Check out [Dan Abramov's JSConf Iceland](#) talk to see what's in store for data fetching with Suspense.

Demo

<https://github.com/dmmulroy/react-md-lazy-talk>

To summarize

- React 16.6 introduced `lazy()` and `<Suspense />`
- Lazy loading is enabled through code splitting, dynamic imports (stage 3 in TC39), and `Suspense`.
- Common use cases could be lazily loading each page/view in your app, complex components, or components that may be infrequently used.
- Modern tooling + React enables lazy loading with little complexity and can drastically reduce your bundle sizes and increase load times

- <https://reactjs.org/docs/code-splitting.html>
- <https://babeljs.io/docs/en/babel-plugin-syntax-dynamic-import>
- <https://github.com/tc39/proposal-dynamic-import>
- <https://www.youtube.com/watch?v=nLF0n9SACd4>
- <https://github.com/FormidableLabs/webpack-dashboard>
- <https://carbon.now.sh>

Find Me @

- github: <https://github.com/dmmulroy>
- twitter: https://twitter.com/dillon_mulroy
- gmail: dillon.mulroy@gmail.com