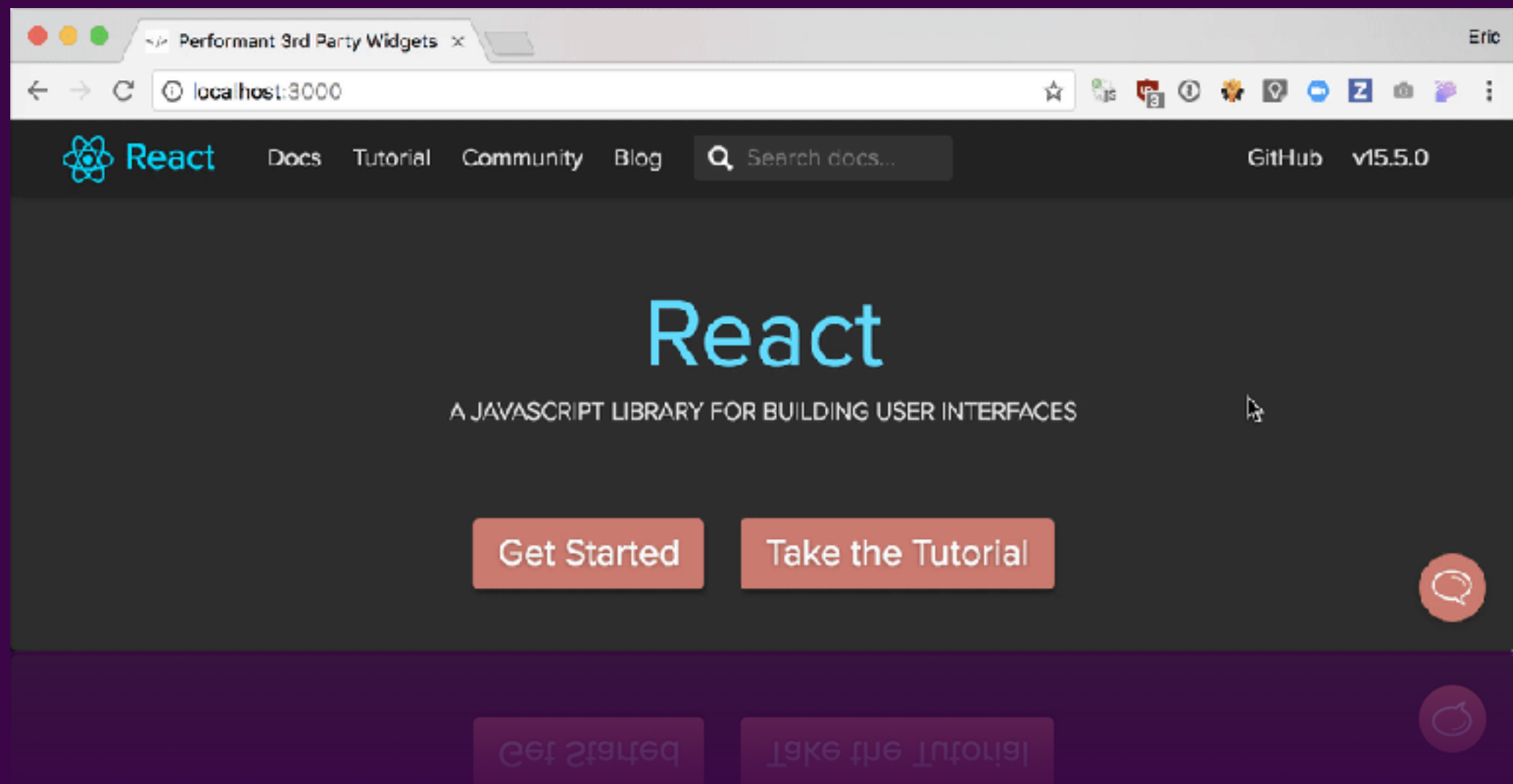# Shipping Performant 3rd Party JS Widgets

@chaddonohue
@ericclemmons
@poeticninja

Example

```html
<div
    data-my-widget="chat"
    data-accent-color="#61dafb"
    data-button-color="#cc7a6f"
><script async defer src="https://path.to/widgets.js"></script></div>
```

Addy Osmani ✔
@addyosmani

Following ∨

Replying to @slightlylate @AdamRackis and 2 others

Average SPA ships 420KB JS, spends 5s in startup (20-30% in parse), 10-15s to TTI. I wouldn't put a ratio on it (variance) but parse matters

RETWEETS   LIKES
9   27

6:47 PM - 26 Mar 2017

Tools

- babel

- webpack

- https://github.com/th0r/webpack-bundle-analyzer
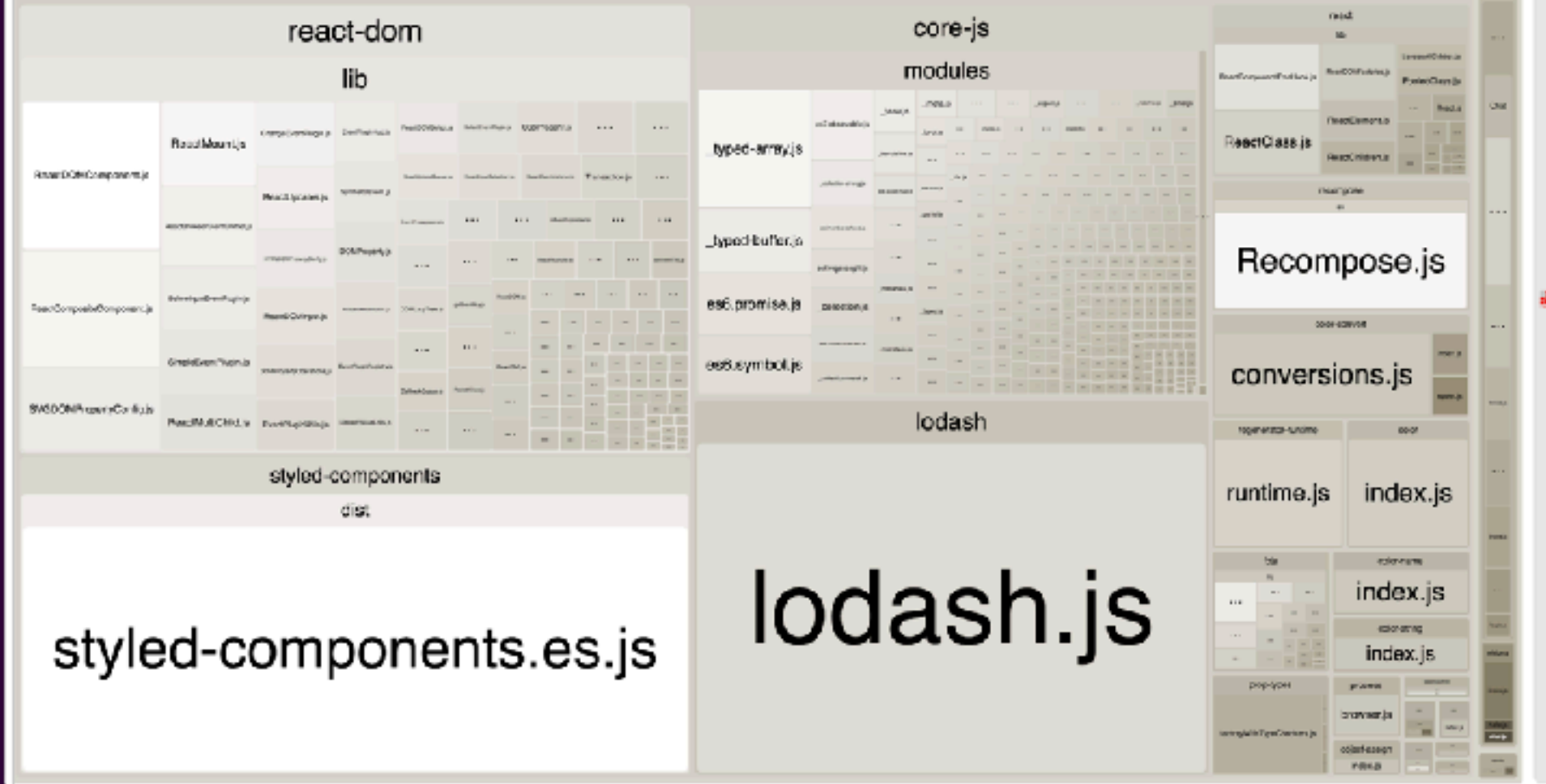
(2.19MB)

Single **widgets.js** file served to users with *everything*.

webpack -p
(427KB)

- Automatically minifies using UglifyJS.

- Shims *process.env.NODE_ENV = "production" for dead-code removal.*

- https://webpack.js.org/guides/production-build/ #the-automatic-way

compression-webpack-plugin
(134KB)

- Configurable, compressed bundles with Gzip.

```
49    plugins: [
50      new CompressionPlugin({
51        test: /\.(html|js)$/
52      })
53    ]
```

- https://github.com/webpack-contrib/compression-webpack-plugin

Tree Shaking (109kB)

- **babel-plugin-lodash**

```
1    {
2      "plugins": [
3  +      ["lodash", {
4  +        "id": ["lodash", "recompose"]
5  +      }],
```

- https://github.com/lodash/babel-plugin-lodash

- "Tree shaking" bundles the named imports vs. an entire CommonJS module.

  - https://webpack.js.org/guides/tree-shaking/

Preact
(71.2kB)

- **preact-compat** is a drop-in replacement for **react**

- Use **webpack**'s **resolve.alias:**

```
20  +  resolve: {
21  +    alias: {
22  +      react: "preact-compat",
23  +      "react-dom": "preact-compat"
24  +    }
```

Code Splitting
(31KB for widgets.js)

- Automatically splits dynamic **import**s

```
25    -   const render = widgets[myWidget];
   26 +     // Attempt to import widget init script, or throw exception
   27 +     const render = await (async () => {
   28 +       try {
   29 +         const imported = await import(`./${myWidget}.js`);
   30 +
   31 +         // `import`ed code-split modules contain all exports
   32 +         return imported.default;
   33 +       } catch (e) {
   34 +         // Friendly error message for invalid widget names
   35 +         if (e.message.match("Cannot find module")) {
   36 +           throw new Error(`Unknown data-my-widget: ${myWidget}`);
   37 +         } else {
   38 +           throw e;
   39 +         }
   40 +       }
   41 +     })();
```

Asset	Size	Chunks		Chunk Names
widgets.js	230 kB	0	[emitted]	widgets
widgets.js.map	92 bytes	0	[emitted]	widgets
widgets.js.gz	72.2 kB		[emitted]	
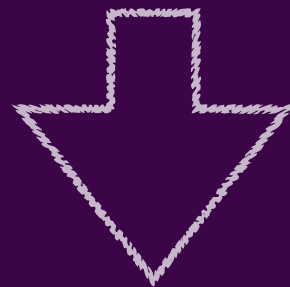
Asset	Size	Chunks		Chunk Names
0.f7f14-546e3d2.js	136 kB	0	[emitted]	
1.f7f14-642abe5.js	20.9 kB	1	[emitted]	
widgets.js	96.6 kB	2	[emitted]	widgets
0.f7f14-546e3d2.js.map	108 bytes	0	[emitted]	
1.f7f14-642abe5.js.map	108 bytes	1	[emitted]	
widgets.js.map	92 bytes	2	[emitted]	widgets
1.f7f14-642abe5.js.gz	7.82 kB		[emitted]	
widgets.js.gz	31.7 kB		[emitted]	
0.f7f14-546e3d2.js.gz	40.9 kB		[emitted]	

Targeted Polyfills
(6.54kB)

- Bundle & ship code to majority of your users based on their browser statistics.

- Lazy-load additional polyfills for uncommon browser targets.

- https://github.com/zloirock/core-js
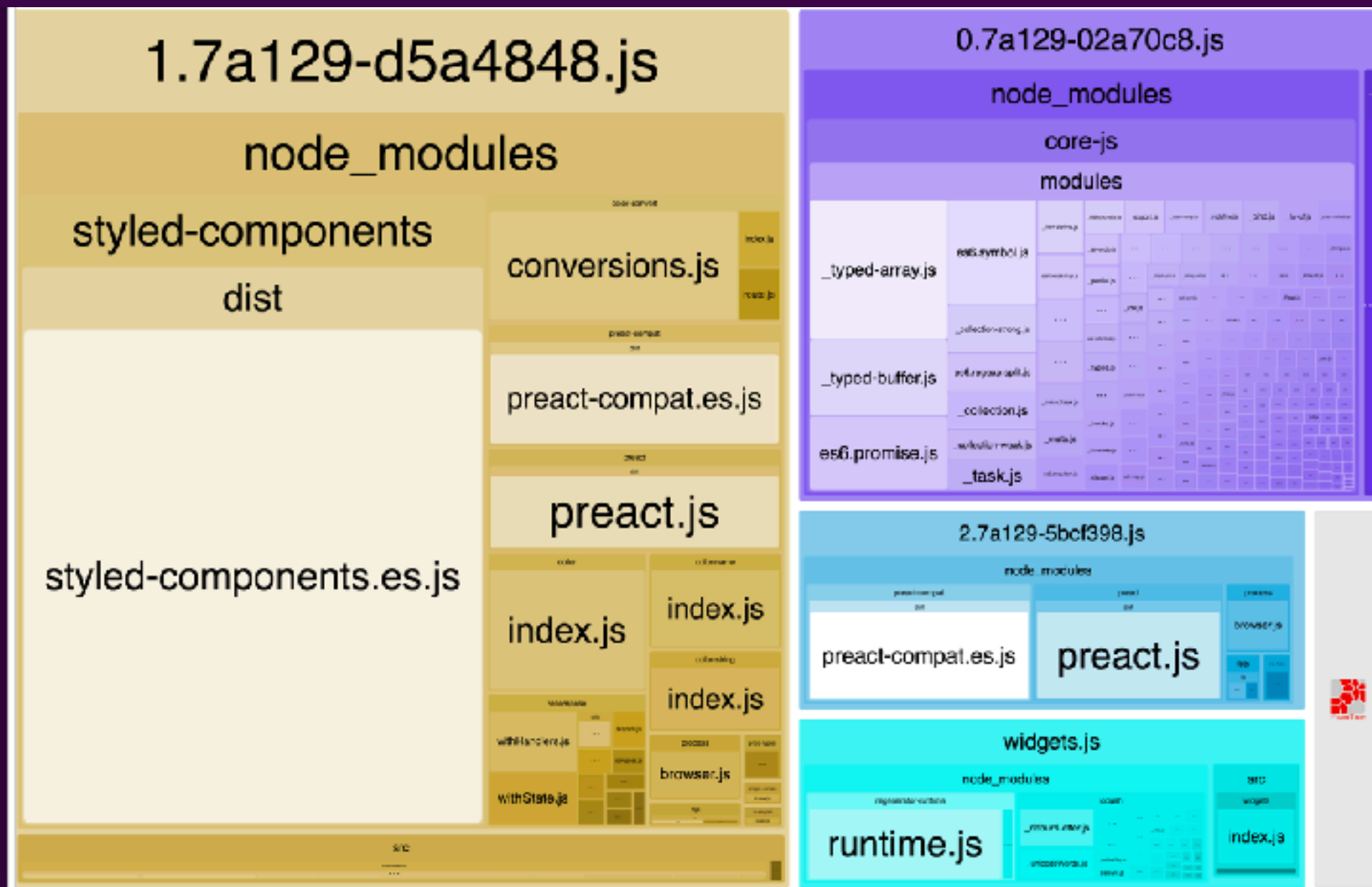
chat.js (40.8kB)
hello.js (7.82kB)
polyfills.js (17kB)
widgets.js (6.54kB)

How to draw an owl

1.

2.

1. Draw some circles   2. Draw the rest of the fucking owl

styled-components

- Popular CSS-in-JS solution for dynamic styles

- **21kB**

- Significant effort to convert to alternatives.

- (11kB in savings with **styled-jsx**)

color

- Popular color manipulation library.

- **8kB**

- **polychrome** (2kB) is nearly a drop-in replacement.

- (6kB in savings)

setState

- State management libraries can add bloat.

- React's **setState** can solve for most isolated state management needs.

- (3kB in savings)

chat.js (20.03kB)
hello.js (7.82kB)
polyfills.js (17kB)
widgets.js (6.54kB)

Thanks!

https://github.com/ericclemmons/performant-3rd-party-widgets