

Bundle your modules with
Webpack

Jake Peyser

November 2nd, 2016

“The module bundler. That also supports pre-processing, code splitting, request caching, automatic reloading...”

But... why?



- Front end development involves many repetitive tasks
- Production SPAs require a lot of code per page
- Scalable web apps require efficient static content delivery

The Basics

index.js

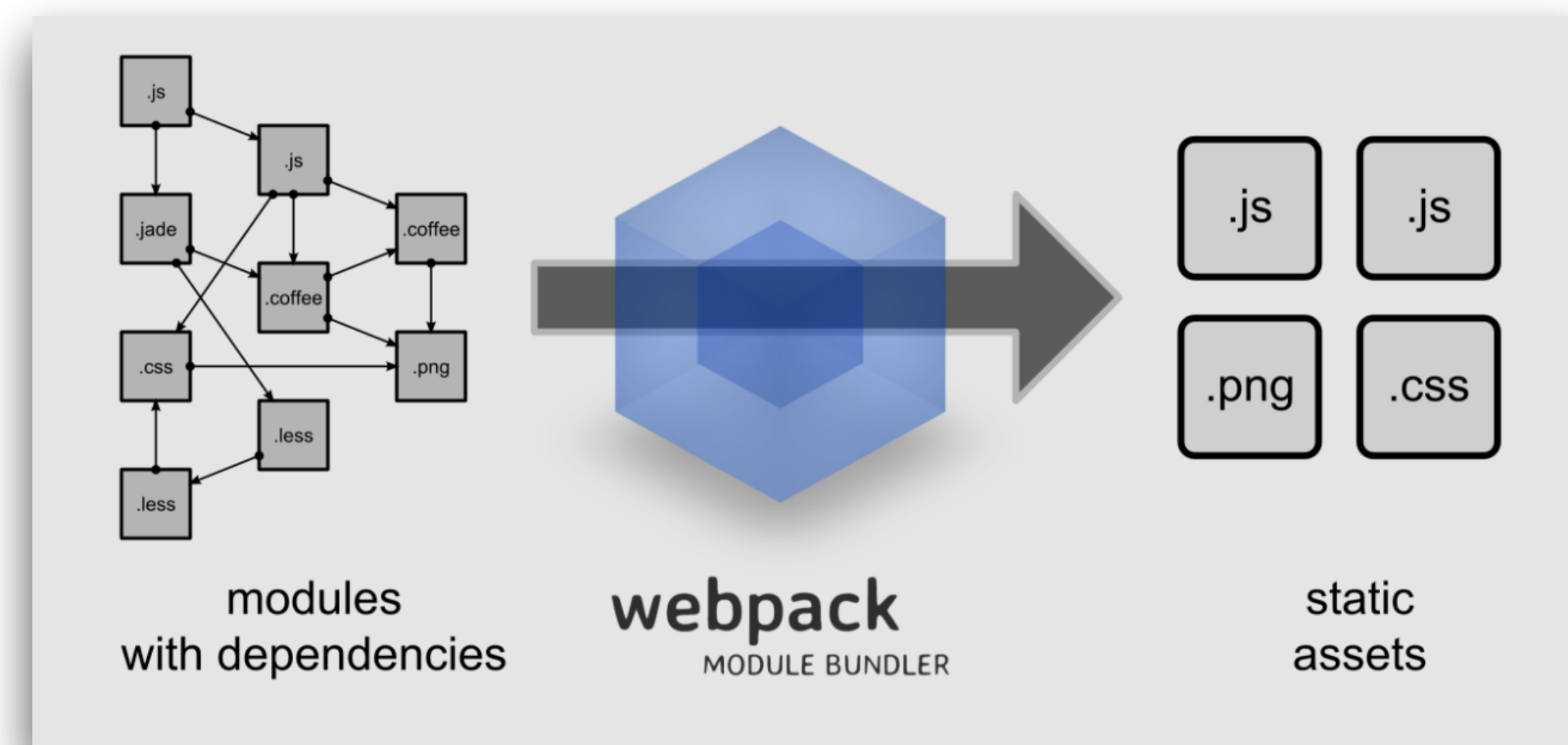
```
1 | console.log('Hello world!')
```

webpack.config.js

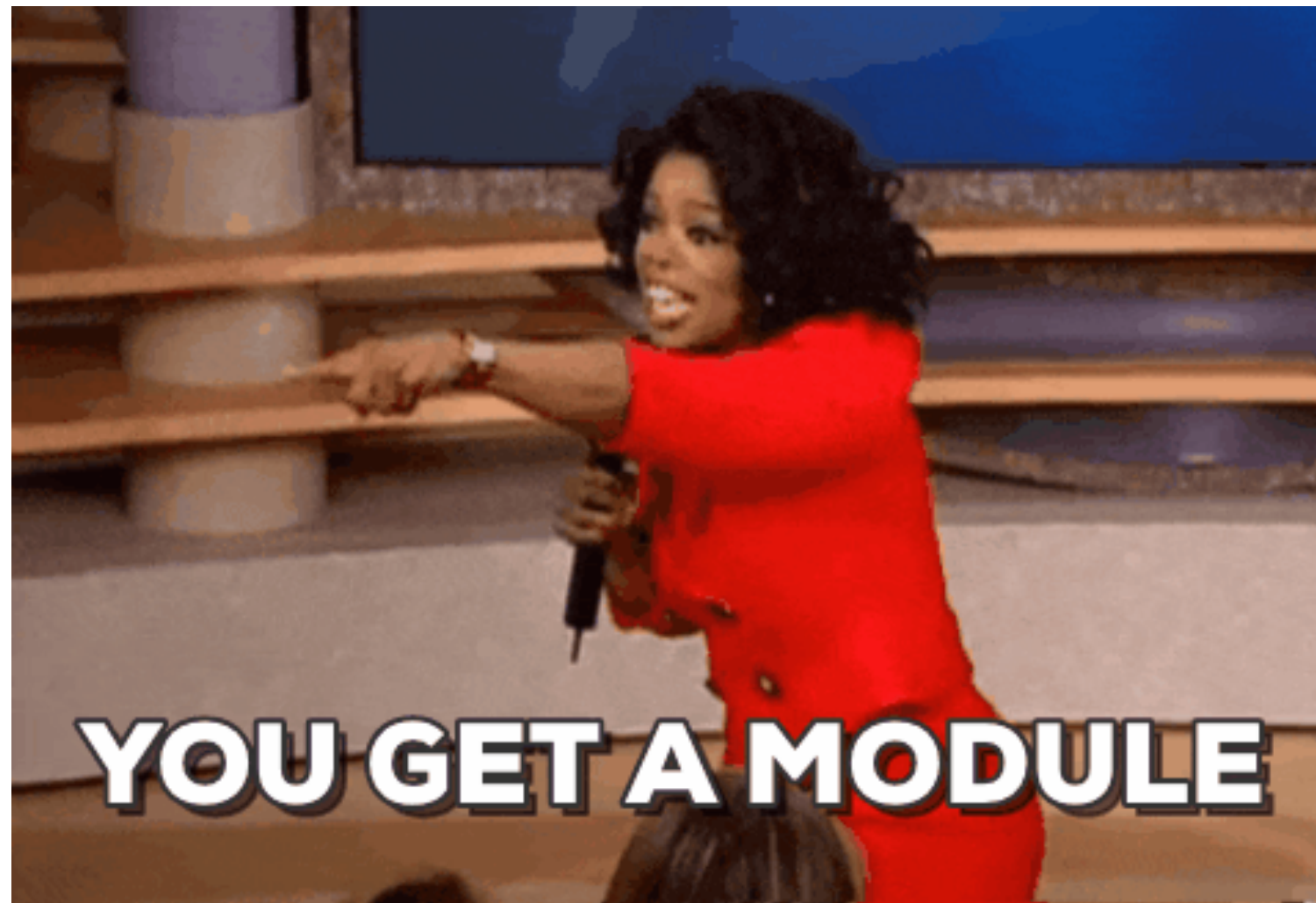
```
1 | module.exports = {  
2 |   entry: './src/index.js',  
3 |   output: {  
4 |     filename: 'bundle.js',  
5 |     path: __dirname + '/public'  
6 |   }  
7 | }
```

Output

```
1 | > webpack-basics@ build /basics  
2 | > webpack  
3 |  
4 | Hash: 167fc2428e7601add9ff  
5 | Version: webpack 1.13.3  
6 | Time: 62ms  
7 |    Asset      Size  Chunks             Chunk Names  
8 | bundle.js  1.42 kB       0  [emitted]  main  
9 |    [0] ./src/index.js 28 bytes {0} [built]
```



- Specify your source code **entry** point
- Designate your bundled **output** destination



Module Bundling

index.js

```
1 | const message = require('./message');
2 | console.log(message.hello());
```

message.js

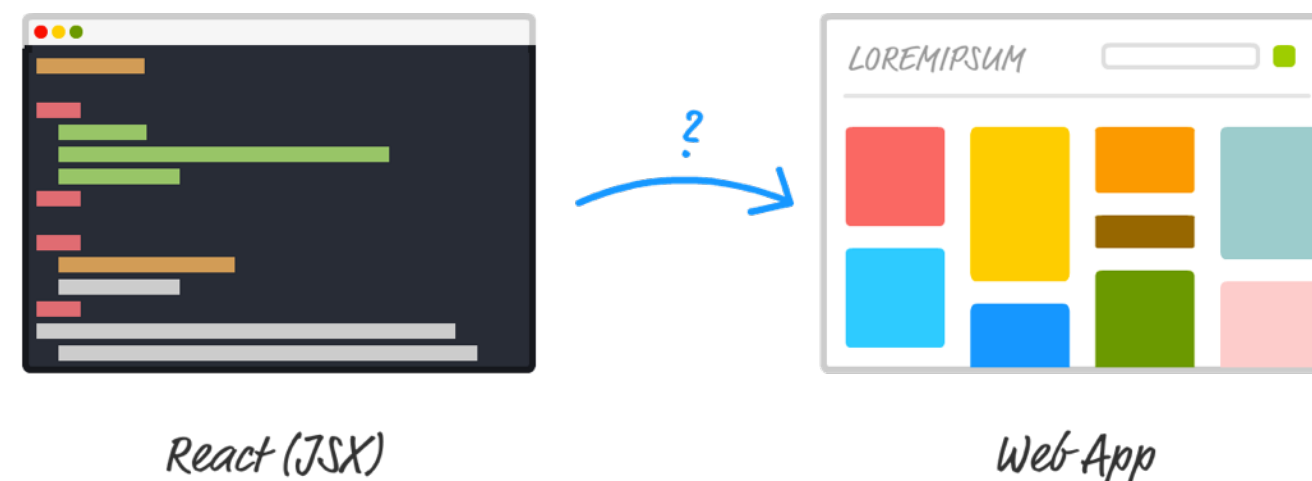
```
1 | module.exports = {
2 |   hello: () => {
3 |     return 'Hello world!'
4 |   }
5 | };
```

Output

```
1 | > webpack-basics@ build /module_bundling
2 | > webpack
3 |
4 | Hash: 1885378d3bfd437a29c9
5 | Version: webpack 1.13.3
6 | Time: 77ms
7 |      Asset      Size  Chunks             Chunk Names
8 | bundle.js  1.61 kB      0  [emitted]  main
9 |    [0] ./src/index.js 69 bytes {0} [built]
10 |    [1] ./src/message.js 69 bytes {0} [built]
```

- Webpack will crawl the ***dependency graph***, starting at your entry
- Creates a single JS file for each designated ***bundle***

Task Running & Plugins



Output

```

1 > webpack-basics@ build /task_running
2 > webpack
3
4 Hash: 4cb07f1d22a9a69fbd87
5 Version: webpack 1.13.3
6 Time: 6392ms
7     Asset      Size  Chunks             Chunk Names
8 bundle.js  220 kB          0  [emitted]  main
9     + 173 hidden modules

```

App.js

```

1 import React from 'react';
2
3 export default () => (
4   <span>Hello world!</span>
5 );

```

index.js

```

1 import React from 'react'
2 import { render } from 'react-dom'
3 import App from './App'
4
5 render(
6   <App />,
7   document.getElementById('main')
8 )

```

webpack.config.js

```

1 const webpack = require('webpack');
2 module.exports = {
3   entry: './src/index.js',
4   output: {
5     filename: 'bundle.js',
6     path: __dirname + '/public'
7   },
8   module: {
9     loaders: [
10       {
11         test: /\.jsx?$/,
12         exclude: /node_modules/,
13         loader: 'babel',
14         query: { presets: ['react', 'es2015'] }
15       }
16     ]
17   },
18   plugins: [
19     new webpack.optimize.UglifyJsPlugin()
20   ]
21 }

```

- Transpile source code using **loaders** (eg. jsx -> js)
- Chain and **automate** the order in which loaders run
- *Common* functionality is provided by webpack and others through **plugins**

webpack.config.js

```
1  const webpack = require('webpack');
2  const HtmlWebpackPlugin = require('html-webpack-plugin');
3  module.exports = {
4    entry: {
5      app: './src/index.js',
6      vendor: [ 'react', 'react-dom' ]
7    },
8    output: {
9      filename: '[name].[chunkhash].js',
10     chunkFilename: '[chunkhash].js',
11     path: __dirname + '/public'
12   },
13   module: {
14     loaders: [
15       {
16         test: /\.jsx?$/,
17         exclude: /node_modules/,
18         loader: 'babel',
19         query: { presets: ['react', 'es2015'] }
20       }
21     ]
22   },
23   plugins: [
24     new webpack.optimize.UglifyJsPlugin(),
25     new HtmlWebpackPlugin({
26       template: './src/index.html'
27     }),
28     new webpack.optimize.CommonsChunkPlugin({
29       names: ['vendor', 'manifest']
30     })
31   ]
32 }
```

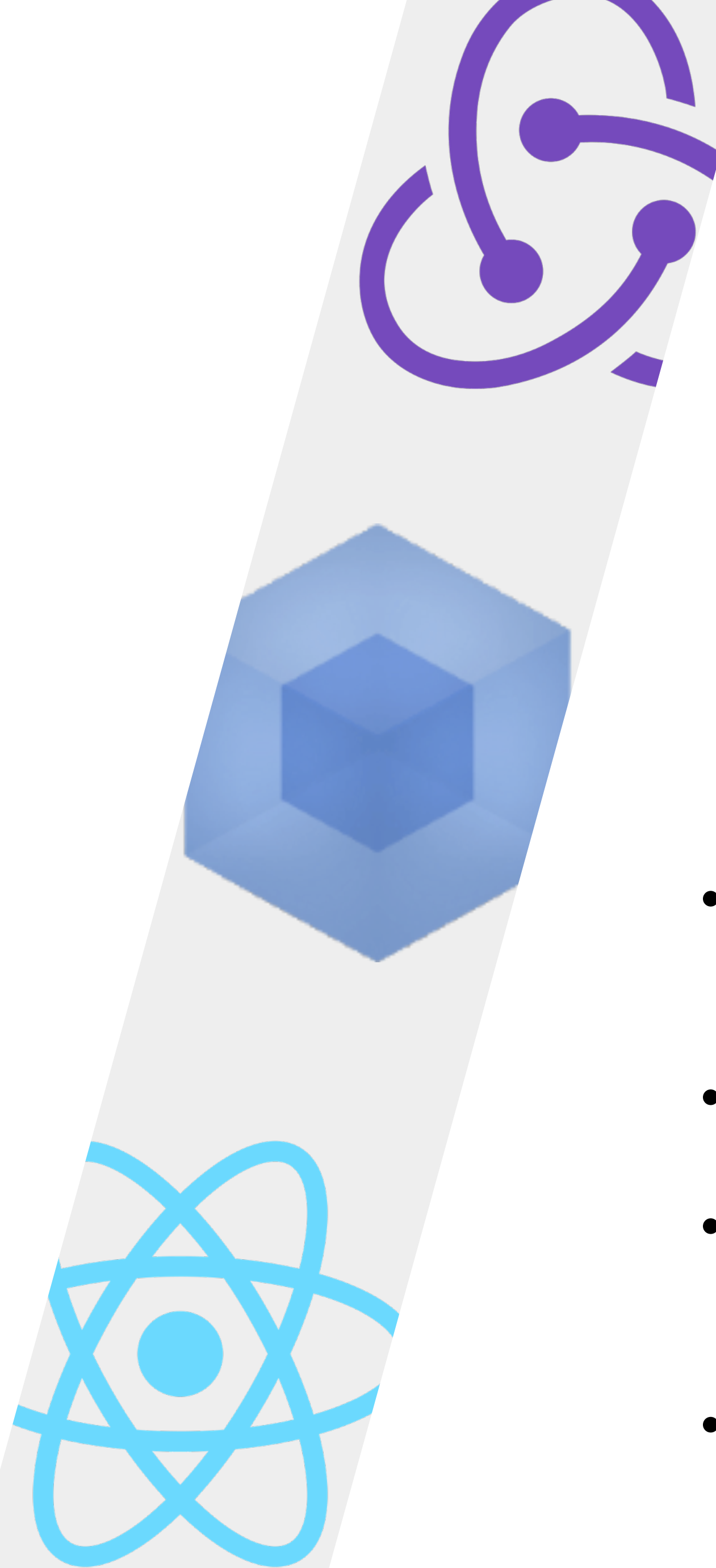
Code Splitting

- **Split your code** into main and vendor bundles
- Creates a single JS file for each designated ***bundle***
- ***Cache*** code that does not change often to decrease initial page load time

Output

```
1  > webpack-basics@ build /code_splitting
2  > webpack
3
4  Hash: d56114b2f69d3ad1d558
5  Version: webpack 1.13.3
6  Time: 6530ms
7
8      Asset      Size  Chunks             Chunk Names
9  app.0165d97c77bf7d9471c8.js  466 bytes    0, 2  [emitted]  app
10 vendor.1b17c05a0b0d6775510a.js    219 kB    1, 2  [emitted]  vendor
11 manifest.47be8c57411e1d9a48fb.js   757 bytes      2  [emitted]  manifest
12 index.html  412 bytes             [emitted]
13
14 [0] multi vendor 40 bytes {1} [built]
15 + 173 hidden modules
```

Key Points



- Most popular build tool adopted by the React community
- Powerful and very heavy on configuration
- [create-react-app](#) provides a boilerplate with nice abstraction
- Most advanced features are not necessary outside of production

Helpful Resources

- [Webpack: When to Use and Why](#)
- [Gulp vs Grunt: Why use task runners and the elegance of plugins](#)
- [Create React Apps with No Configuration](#)
- [Utilizing Webpack and Babel to build a React.js App](#)
- [SurviveJS: Webpack from Apprentice to Master](#)
- [Webpack Docs](#)
- [Webpack 2.0 Docs \(work in progress\)](#)
- [jakepeyser/webpack-intro](#)



Useful Plugins



- [DedupePlugin](#) - Deduplicate equal or similar files in the output
- [UglifyJSPlugin](#) - Minimize all JavaScript output of chunks
- [CommonsChunkPlugin](#) - Split code into chunks
- [OfflinePlugin](#) - Makes your app offline ready by caching all (or some) output assets
- [DefinePlugin](#) - Create global constants which can be configured at compile time
- [HotModuleReplacementPlugin](#) - Enables Hot Module Replacement
- [HtmlWebpackPlugin](#) - Simplifies creation of HTML files to serve your webpack bundles