Webpack Encore

Webpack your App
and Love it!

SymfonyCasts

CodeMash

Yo! I'm Ryan!

- Lead of the Symfony documentation team
- Writer for SymfonyCasts.com
- Symfony evangelist... Fanboy

SymfonyCasts

symfonycasts.com
twitter.com/weaverryan

Yo! I'm Ryan!

> Lead of the Symfony documentation team

> Writer for SymfonyCasts.com

> Symfony evangelist... Fanboy

> Husband of the much more talented @leannapelham

SymfonyCasts

symfonycasts.com
twitter.com/weaverryan

Yo! I'm Ryan!

- Lead of the Symfony documentation team
- Writer for SymfonyCasts.com
- Symfony evangelist... Fanboy
- Husband of the much more talented @leannapelham
- Father to my much more charming son, Beckett

SymfonyCasts

symfonycasts.com
twitter.com/weaverryan

PART 1

All of modern JavaScript in 45 minutes!

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015

@weaverryan

All of modern JavaScript in 45 minutes!

PART 1

ES6, ES2015 , ECMAScript

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015, ECMAScript, NodeJS

@weaverryan

All of modern JavaScript in 45 minutes!

PART 1

ES6, ES2015, ECMAScript, NodeJS

yarn

@weaverryan

All of modern JavaScript in 45 minutes!

PART 1

ES6, ES2015, ECMAScript, NodeJS

yarn, Babel

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015, ECMAScript, NodeJS

yarn, Babel, webpack

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015 , ECMAScript, NodeJS

yarn, Babel, webpack , ReactJS

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015 , ECMAScript, NodeJS

yarn, Babel, webpack , ReactJS, modules ...

@weaverryan

PART 1

All of modern JavaScript in 45 minutes!

ES6, ES2015 , ECMAScript, NodeJS

yarn, Babel, webpack , ReactJS, modules ...

... and of course ...

@weaverryan

All of modern JavaScript in 45 minutes!

PART 1

ES6, ES2015 , ECMAScript, NodeJS

yarn, Babel, webpack , ReactJS, modules ...

... and of course ...

the 12 new JS things they invent during this presentation

@weaverryan

Modern JavaScript is a lot like...

@weaverryan

Modern JavaScript is a lot like...

Game of Thrones

@weaverryan

GoT

JavaScript

@weaverryan

GoT

Countless characters and competing factions fighting for influence

JavaScript

@weaverryan

GoT

Countless characters and competing factions fighting for influence

JavaScript

Countless libraries and competing standards fighting for influence

@weaverryan

GoT

JavaScript

@weaverryan

GoT

That character you love and followed for 2 seasons, was just unceremoniously decapitated

JavaScript

@weaverryan

GoT

That character you love and
followed for 2 seasons, was
just unceremoniously decapitated

JavaScript

You spent 6 months building
your site in “Hipster.js” only
to read on Twitter that:
“no self-respecting dev
uses that crap anymore”

Got

Java

and
was
apitated

ding
only
t:
v

uses that crap anymore”

@weaverryan

```
// yay.js
var message = 'I like Java...Script';

console.log(message);
```

@weaverryan

```
// yay.js
var message = 'I like Java...Script';

console.log(message);
```

```
> node yay.js

I like Java...Script
```

```
// yay.js
var message = 'I like Java...Script';

console.log(message);
```

```
> node yay.js

I like Java...Script
```

NodeJS: server-side JavaScript engine

```
// yay.js
var message = 'I like Java...Script';

console.log(message);
```

```
> node yay.js

I like Java...Script
```

NodeJS: server-side JavaScript engine

yarn/npm: Package manager for NodeJS

```
class ProductCollection
{
    constructor(products) {
        this.products = products;
    }
}

let collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

what language is this?

```
class ProductCollection
{
    constructor(products) {
        this.products =
    }
}

let collection = n
    'Sheer She
    'Wool Hat
    'After-She
    'After-Shear
]);
let prods = c

let loopThroughPro
    for (let prod
        callback(
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

what language is this?

JavaScript

ECMAScript

@weaverryan

ECMAScript

The official name of standard JavaScript

@weaverryan

ECMAScript

The official name of standard JavaScript

ES6/ES2015/Harmony

@weaverryan

ECMAScript

The official name of standard JavaScript

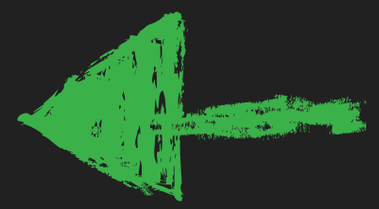
ES6/ES2015/Harmony

The 6th accepted (so official) version of ECMAScript (ES2018 is latest)

@weaverryan

```
class ProductCollection
{
    constructor(products) {
        this.products = products;
    }
}

let collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

```
class ProductCollection
{
    constructor(products) {
        this.products = products;
    }
}

let collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```



Proper class and inheritance syntax

```
class ProductCollection
{
    constructor(products) {
        this.products = products;
    }
}

let collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

let: similar to var, but more hipster

```
class ProductCollection
{
    constructor(products) {
        this.products = products;
    }
}

let collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

```
function (product) {
  console.log(product);
}
```

```
class ProductCollection
{
    constructor(products) {
        th
    }
}

let collec
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);
let prods = collection.getProducts();

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

But will it run in a browser???

```
class ProductCollection
{
    constructor(products) {
        th
    }
}

let collec
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morni
]);
let prods = collection.

let loopThroughProducts = callback => {
    for (let prod of prods) {
        callback(prods);
    }
};

loopThroughProducts(product => console.log('Product: '+product));
```

But will it run in a browser???

Maybe!

Now we just need to wait 5 years for the worst browsers

(ahem, IE)

to support this

… or do we?

Babel

A JS *transpiler!*

Babel is a NodeJS binary...

```
> yarn add --dev @babel/cli
```

... package.json

```
{
  "devDependencies": {
    "@babel/cli": "^7.0.0"
  }
}
```

Babel can transpile anything

* you configure what you want to transpile

Babel can transpile anything

ES6 JS --> ES5 JS

* you configure what you want to transpile

Babel can transpile anything

ES6 JS --> ES5 JS

Draft ES features --> ES5 JS

* you configure what you want to transpile

Babel can transpile anything

ES6 JS --> ES5 JS

Draft ES features --> ES5 JS

CoffeeScript --> JavaScript

* you configure what you want to transpile

Babel can transpile anything

ES6 JS --> ES5 JS

Draft ES features --> ES5 JS

CoffeeScript --> JavaScript

Coffee --> Tea

* you configure what you want to transpile

package.json

```
+  "browserslist": [
+    "> 0.5%",
+    "last 2 versions",
+    "Firefox ESR",
+    "not dead"
+  ]
```

* you configure what you want to transpile

```
> ./node_modules/.bin/babel \
        public/js/productApp.js \
    -o public/build/productApp.js
```

source:

```
loopThroughProducts(
    product => console.log('Product: '+product)
);
```

```
> ./node_modules/.bin/babel \
        public/js/productApp.js \
    -o public/build/productApp.js
```

source:

```
loopThroughProducts(
    product => console.log('Product: '+product)
);
```

built:

```
loopThroughProducts(function (product) {
    return console.log('Product: ' + product);
});
```

Big Takeaway #1:

Modern JavaScript has a build step

But we can use new (or experimental) features now

New to ES6:

JavaScript Modules!

@weaverryan

The Classic Problem:

If you want to organize your JS into multiple files, you need to manually include all those script tags!

```
// public/js/ProductCollection.js

class ProductCollection
{
    constructor(products) {
        this.products = products;
    }

    getProducts() {
        return this.products;
    }

    getProduct(i) {
        return this.products[i];
    }
}

export default ProductCollection;
```

```
// public/js/productApp.js

import ProductCollection from './ProductCollection';

var collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);

// ...
```

(.js extension is optional)

```
// public/js/productApp.js

import ProductCollection from './ProductCollection';

var collection = new ProductCollection([
    'Sheer Shears',
    'Wool Hauling Basket',
    'After-Shear (Fresh Cut Grass)',
    'After-Shear (Morning Dew)',
]);

// ...
```

(.js extension is optional)

```
// public/js/productApp.js

import ProductCollection from './ProductCollection';
```

```
> ./node_modules/.bin/babel \
        public/js/productApp.js \
    -o public/builds/productApp.js

  // ...
```

```
<script src="/builds/productApp.js"></script>
```

Elements Console Sources Network Timeline

top Preserve log

▶ Uncaught ReferenceError: require is not defined

```
3 var _ProductCollection = require('./ProductCollection');
```

Module loading in a browser is hard to do

@weaverryan

Introducing...

@weaverryan

Webpack!

- bundler
- module loader
- all-around nice guy

Install webpack

```
> yarn add --dev webpack
```

```
// public/js/ProductCollection.js

class ProductCollection
{
    // ...
}
 export default ProductCollection;
```

```
// public/js/productApp.js

import ProductCollection from './ProductCollection';

// …
```

Go webpack Go!

```
> ./node_modules/.bin/webpack \
    public/js/productApp.js \
    public/build/productApp.js
```

Go webpack Go!

```
> ./node_modules/.bin/webpack \
    public/js/productApp.js \
    public/build/productApp.js
```

The one built file contains the code from both source files

```
Hash: b37f59a4e511e17cbadf
Version: webpack 1.13.1
Time: 60ms
        Asset     Size  Chunks             Chunk Names
productApp.js  2.24 kB       0  [emitted]  main
   [0]public/js/productApp.js 512 bytes {0} [built]
   [1]public/js/ProductCollection.js 250 bytes {0} [built]
```

Optional config to make it easier to use:

```
// webpack.config.js
module.exports = {
    entry: {
        product: './public/js/productApp.js'
    },
    output: {
        path: ‘./public/build',
        filename: '[name].js',
        publicPath: '/build/'
    }
};
```

Optional config to make it easier to use:

```
// webpack.config.js
module.exports = {
    entry: {
        product: './public/js/productApp.js'
    },
    output: {
        path: ‘./public/build',
        filename: '[name].js',
        publicPath: '/build/'
    }
};
```

Optional config to make it easier to use:

```
// webpack.config.js
module.exports = {
    entry: {
        product: './public/js/productApp.js'
    },
    output: {
        path: ‘./public/build',
        filename: '[name].js',
        publicPath: '/build/'
    }
};
```

public/build/product.js

```
<script src="/build/product.js"></script>
```

```
> ./node_modules/.bin/webpack
```

```
> yarn run webpack
```

Great!

Now let’s add more features to Webpack!

@weaverryan

Features, features, features, features, features

- babel transpiling
- dev server
- production optimizations
- CSS handling
- Sass/LESS
- PostCSS
- React
- Vue
- versioning
- source maps
- image handling
- extract css
- code splitting
- jQuery providing
- TypeScript
- friendly errors

No Problem

@weaverryan

```
var path = require('path');
var process = require('process');
var webpack    = require('webpack');
var production = process.env.NODE_ENV === 'production';

// helps load CSS to their own file
var ExtractPlugin = require('extract-text-webpack-plugin');
var CleanPlugin = require('clean-webpack-plugin');
var ManifestPlugin = require('webpack-manifest-plugin');

var plugins = [
    new ExtractPlugin('[name]-[contenthash].css'), // <=== where should content be piped

    // put vendor_react stuff into its own file
    // new webpack.optimize.CommonsChunkPlugin({
    //     name: 'vendor_react',
    //     chunks:    ['vendor_react'],
    //     minChunks: Infinity, // avoid anything else going in here
    // }),
    new webpack.optimize.CommonsChunkPlugin({
        name:      'main', // Move dependencies to the "main" entry
        minChunks: Infinity, // How many times a dependency must come up before being extracted
    }),
    new ManifestPlugin({
        filename: 'manifest.json',
        // prefixes all keys with builds/, which allows us to refer to
        // the paths as builds/main.css in Twig, instead of just main.css
        basePath: 'builds/'
    }),
];

if (production) {
    plugins = plugins.concat([
        // This plugin looks for similar chunks and files
        // and merges them for better caching by the user
        new webpack.optimize.DedupePlugin(),

        // This plugins optimizes chunks and modules by
        // how much they are used in your app
        new webpack.optimize.OccurenceOrderPlugin(),

        // This plugin prevents Webpack from creating chunks
        // that would be too small to be worth loading separately
        new webpack.optimize.MinChunkSizePlugin({
            minChunkSize: 51200, // ~50kb
        }),

        // This plugin minifies all the Javascript code of the final bundle
        new webpack.optimize.UglifyJsPlugin({
            mangle: true,
            compress: {
                warnings: false, // Suppress uglification warnings
            },
            sourceMap: false
        }),

        // This plugins defines various variables that we can set to false
        // in production to avoid code related to them from being compiled
        // in our final bundle
        new webpack.DefinePlugin({
            __SERVER__:      !production,
            __DEVELOPMENT__: !production,
            __DEVTOOLS__:    !production,
            'process.env': {
                BABEL_ENV: JSON.stringify(process.env.NODE_ENV),
                'NODE_ENV': JSON.stringify('production')
            },
        }),
        new CleanPlugin('web/builds', {
            root: path.resolve(__dirname , '..')
        }),
    ]);
}

module.exports = {
    entry: {
        main: './main',
        video: './video',
        checkout_login_registration: './checkout_login_registration',
        team_pricing: './team_pricing',
        credit_card: './credit_card',
        team_subscription: './team_subscription',
        track_organization: './track_organization',
        challenge: './challenge',
        workflow: './workflow',
        code_block_styles: './code_block_styles',
        content_release: './content_release',
        script_editor: './script_editor',
        sweetalert2_legacy: './sweetalert2_legacy',
        admin: './admin',
        admin_user_refund: './admin_user_refund',

        // vendor entry points to be extracted into their own file
        // we do this to keep main.js smaller... but it's a pain
        // because now we need to manually add the script tag for
        // this file is we use react or react-dom
        // vendor_react: ['react', 'react-dom'],
    },
    output: {
        path: path.resolve(__dirname, '../web/builds'),
        filename: '[name]-[hash].js',
        chunkFilename: '[name]-[chunkhash].js',
        // in dev, make all URLs go through the webpack-dev-server
        // things "mostly" work without this, but AJAX calls for chunks
        // are made to the local, Symfony server without this
        publicPath: production ? '/builds/' : 'http://localhost:8090/builds/'
    },
    plugins: plugins,
    module: {
        loaders: [
            {
                test: /\.js$/,
                exclude: /node_modules/,
                loader: "babel-loader"
            },
            {
                test:   /\.scss/,
                loader: ExtractPlugin.extract('style', 'css!sass'),
            },
            {
                test:   /\.css/,
                loader: ExtractPlugin.extract('style', 'css'),
            },
            {
                test:   /\.(png|gif|jpe?g|svg)?(\?v=[0-9].[0-9].[0-9])?$/i,
                loader: 'url?limit=10000',
            },
            {
                // the ?(\?v=[0-9].[0-9].[0-9])? is for ?v=1.1.1 format
                test: /\.woff(2)?(\?v=[0-9].[0-9].[0-9])?$/,
                // Inline small woff files and output them below font/.
                // Set mimetype just in case.
                loader: 'url',
                query: {
                    prefix: 'font/',
                    limit: 5000,
                    mimetype: 'application/font-woff'
                },
                //include: PATHS.fonts
            },
            {
                test: /\.ttf?(\?v=[0-9].[0-9].[0-9])?$\.eot?(\?v=[0-9].[0-9].[0-9])?$/,
                loader: 'file',
                query: {
                    name: '[name]-[hash].[ext]',
                    // does this do anything?
                    prefix: 'font/',
                },
                //include: PATHS.fonts
            },
            {
                test: /\.json$/,
                loader: "json-loader"
            },
        ]
    },
    node: {
        fs: 'empty'
    },
    debug:   !production,
    devtool: production ? false : 'eval',
    devServer: {
        hot: true,
        port: 8090,
        // tells webpack-dev-server where to serve public files from
        contentBase: '../web/',
        headers: { "Access-Control-Allow-Origin": "*" }
    },
};
```

@weaverryan

```
var path = require('path');
var process = require('process');
var webpack    = require('webpack');
var production = process.env.NODE_ENV === 'production';

// helps load CSS to their own file
var ExtractPlugin = require('extract-text-webpack-plugin');
var CleanPlugin = require('clean-webpack-plugin');
var ManifestPlugin = require('webpack-manifest-plugin');

var plugins = [
    new ExtractPlugin('[name]-[contenthash].css'), // <=== where should content be piped

    // put vendor_react stuff into its own file
    // new webpack.optimize.CommonsChunkPlugin({
    //     name: 'vendor_react',
    //     chunks:    ['vendor_react'],
    //     minChunks: Infinity, // avoid anything else going in here
    // }),
    new webpack.optimize.CommonsChunkPlugin({
        name:      'main', // Move dependencies to the "main" entry
        minChunks: Infinity, // How many times a dependency must come up before being extracted
    }),
    new ManifestPlugin({
        filename: 'manifest.json',
        // prefixes all keys with builds/, which allows us to refer to
        // the paths as builds/main.css in Twig, instead of just main.css
        basePath: 'builds/'
    }),
];

if (production) {
    plugins = plugins.concat([
        // This plugin looks for similar chunks and files
        // and merges them for better caching by the user
        new webpack.optimize.DedupePlugin(),

        // This plugins optimizes chunks and modules by
        // how much they are used in your app
        new webpack.optimize.OccurenceOrderPlugin(),

        // This plugin prevents Webpack from creating chunks
        // that would be too small to be worth loading separately
        new webpack.optimize.MinChunkSizePlugin({
            minChunkSize: 51200, // ~50kb
        }),

        // This plugin minifies all the Javascript code of the final bundle
        new webpack.optimize.UglifyJsPlugin({
            mangle: true,
            compress: {
                warnings: false, // Suppress uglification warnings
            },
            sourceMap: false
        }),

        // This plugins defines various variables that we can set to false
        // in production to avoid code related to them from being compiled
        // in our final bundle
        new webpack.DefinePlugin({
            __SERVER__: !production,
            __DEVELOPMENT__: !production,
            __DEVTOOLS__: !production,
            'process.env': {
                BABEL_ENV: JSON.stringify(process.env.NODE_ENV),
                'NODE_ENV': JSON.stringify('production')
            },
        }),
        new CleanPlugin('web/builds', {
            root: path.resolve(__dirname , '.' )
        }),
    ]);
}

module.exports = {
    entry: {
        main: './main',
        video: './video',
        checkout_login_registration: './checkout_login_registration',
        team_pricing: './team_pricing',
        credit_card: './credit_card',
        team_subscription: './team_subscription',
        track_organization: './track_organization',
        challenge: './challenge',
        workflow: './workflow',
        code_block_styles: './code_block_styles',
        content_release: './content_release',
        script_editor: './script_editor',
        sweetalert2_legacy: './sweetalert2_legacy',
        admin: './admin',
        admin_user_refund: './admin_user_refund',

        // vendor entry points to be extracted into their own file
        // we do this to keep main.js smaller... but it's a pain
        // because now we need to manually add the script tag for
        // this file is we use react or react-dom
        // vendor_react: ['react', 'react-dom'],
    },
    output: {
        path: path.resolve(__dirname, '../web/builds'),
        filename: '[name]-[hash].js',
        chunkFilename: '[name]-[chunkhash].js',
        // in dev, make all URLs go through the webpack-dev-server
        // things *mostly* work without this, but AJAX calls for chunks
        // are made to the local, Symfony server without this
        publicPath: production ? '/builds/' : 'http://localhost:8090/builds/'
    },
    plugins: plugins,
    module: {
        loaders: [
            {
                test: /\.js$/,
                exclude: /node_modules/,
                loader: "babel-loader"
            },
            {
                test:   /\.scss/,
                loader: ExtractPlugin.extract('style', 'css!sass'),
            },
            {
                test:   /\.css/,
                loader: ExtractPlugin.extract('style', 'css'),
            },
            {
                test:   /\.(png|gif|jpe?g|svg)?(\?v=[0-9].[0-9].[0-9])?$/i,
                loader: 'url?limit=10000',
            },
            {
                // the ?(\?v=[0-9].[0-9].[0-9])? is for ?v=1.1.1 format
                test: /\.woff(2)?(\?v=[0-9].[0-9].[0-9])?$/,
                // Inline small woff files and output them below font/.
                // Set mimetype just in case.
                loader: 'url',
                query: {
                    prefix: 'font/',
                    limit: 5000,
                    mimetype: 'application/font-woff'
                },
                //include: PATHS.fonts
            },
            {
                test: /\.ttf?(\?v=[0-9].[0-9].[0-9])?$\.eot?(\?v=[0-9].[0-9].[0-9])?$/,
                loader: 'file',
                query: {
                    name: '[name]-[hash].[ext]',
                    // does this do anything?
                    prefix: 'font/',
                },
                //include: PATHS.fonts
            },
            {
                test: /\.json$/,
                loader: "json-loader"
            },
        ]
    },
    node: {
        fs: 'empty'
    },
    debug:   !production,
    devtool: production ? false : 'eval',
    devServer: {
        hot: true,
        port: 8090,
        // tells webpack-dev-server where to serve public files from
        contentBase: '../web/',
        headers: { "Access-Control-Allow-Origin": "*" }
    },
};
```

*** not visible here:

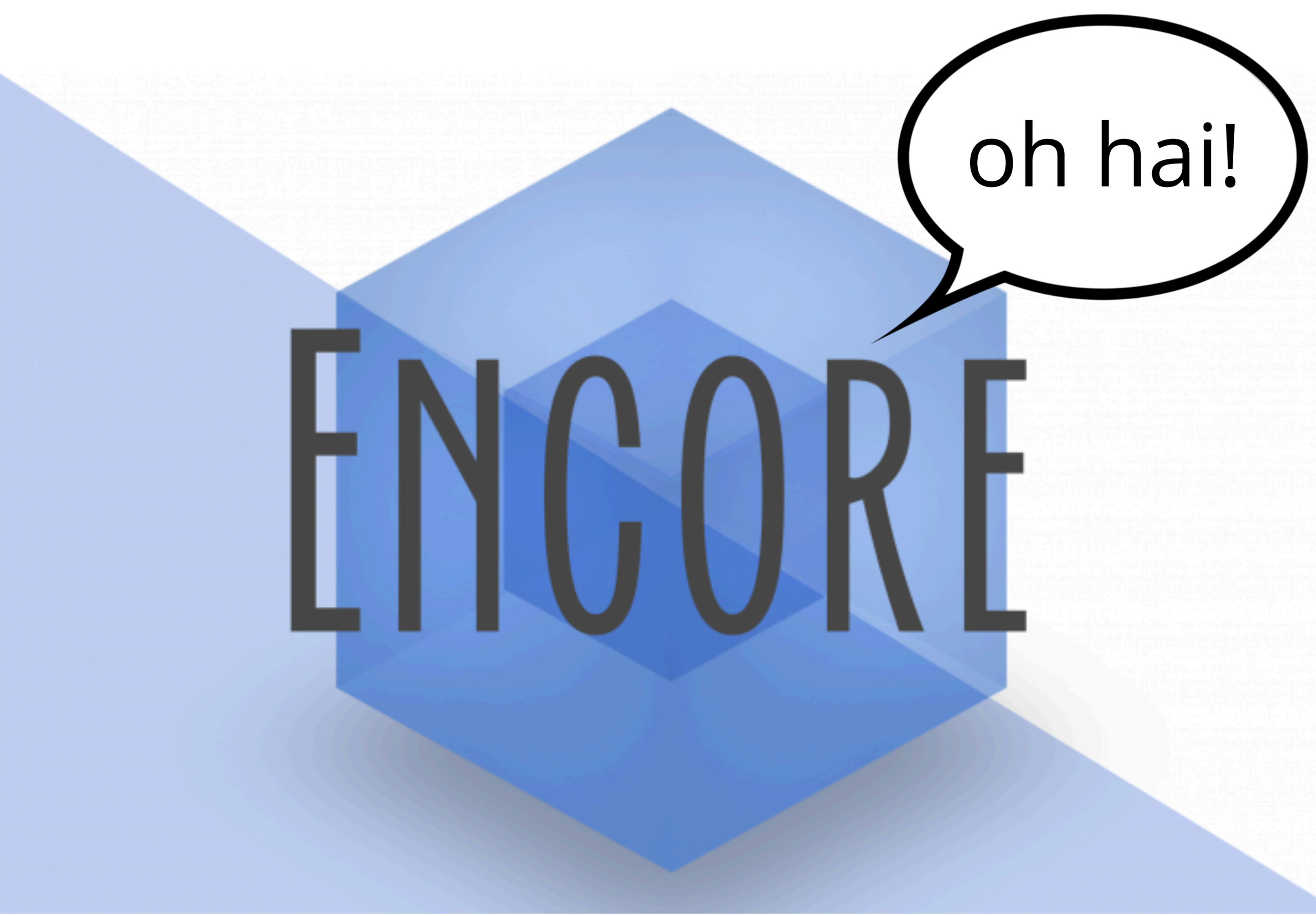
the 1000 ways you can ruin your day.

@weaverryan

PART 2

Hello Webpack Encore

@weaverryan


oh hai!
ENCORE

Let’s start over

@weaverryan

Let’s start over

```
> rm -rf package.json yarn.lock node_modules/

> cowsay “Make mooooore room please”
 ____________________________
< Make mooooore room please >
 ----------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

… and let's start simple

```
// assets/js/cow_say.js
export default function (msg) {
   return `The cow says: ${msg}`;
}
```

… and let's start simple

```
// assets/js/cow_say.js
export default function (msg) {
   return `The cow says: ${msg}`;
}
```

```
// assets/js/app.js
import cow_say from './cow_say';

console.log(cow_say('Moooo'));
```

Step 1: Install Encore

```
> yarn add @symfony/webpack-encore --dev
```

Step 2: webpack.config.js

```
var Encore = require('@symfony/webpack-encore');

Encore
    .setOutputPath('public/build/')
    .setPublicPath('/build')

    // will output as public/build/app.js
    .addEntry('app', './assets/js/app.js')

    .enableSourceMaps(!Encore.isProduction())
;

module.exports = Encore.getWebpackConfig();
```

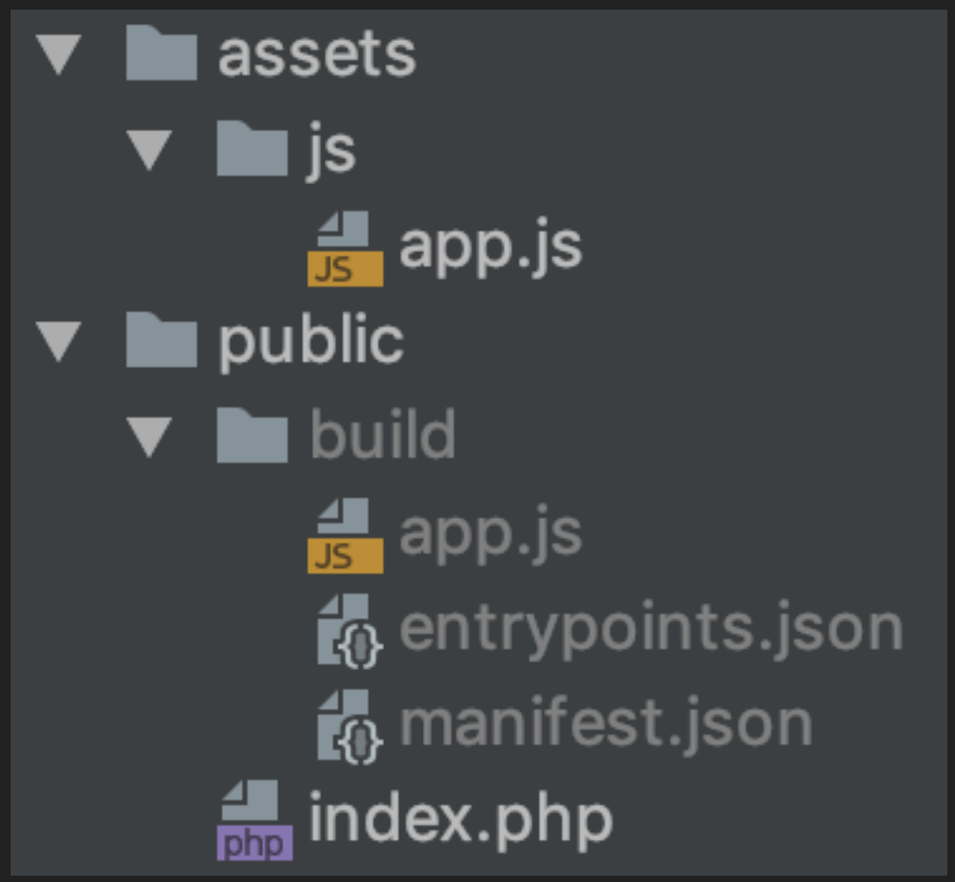
Step 2: webpack.config.js

```
var Encore = require('@symfony/webpack-encore');

Encore
    .setOutputPath('public/build/')
    .setPublicPath('/build')

    // will output as public/build/app.js
    .addEntry('app', './assets/app.js')

    // will output as public/build/global.css
    .addStyleEntry('global', './assets/global.scss')

    // allow sass/scss files to be processed
    .enableSassLoader()

    // allow legacy applications to use $/jQuery as a global variable
    .autoProvidejQuery()

    // enable source maps during development
    .enableSourceMaps(!Encore.isProduction())

    // empty the outputPath dir before each build
    .cleanupOutputBeforeBuild()

    // show OS notifications when builds finish/fail
    .enableBuildNotifications()

    // create hashed filenames (e.g. app.abc123.css)
    .enableVersioning()
;

module.exports = Encore.getWebpackConfig();
```

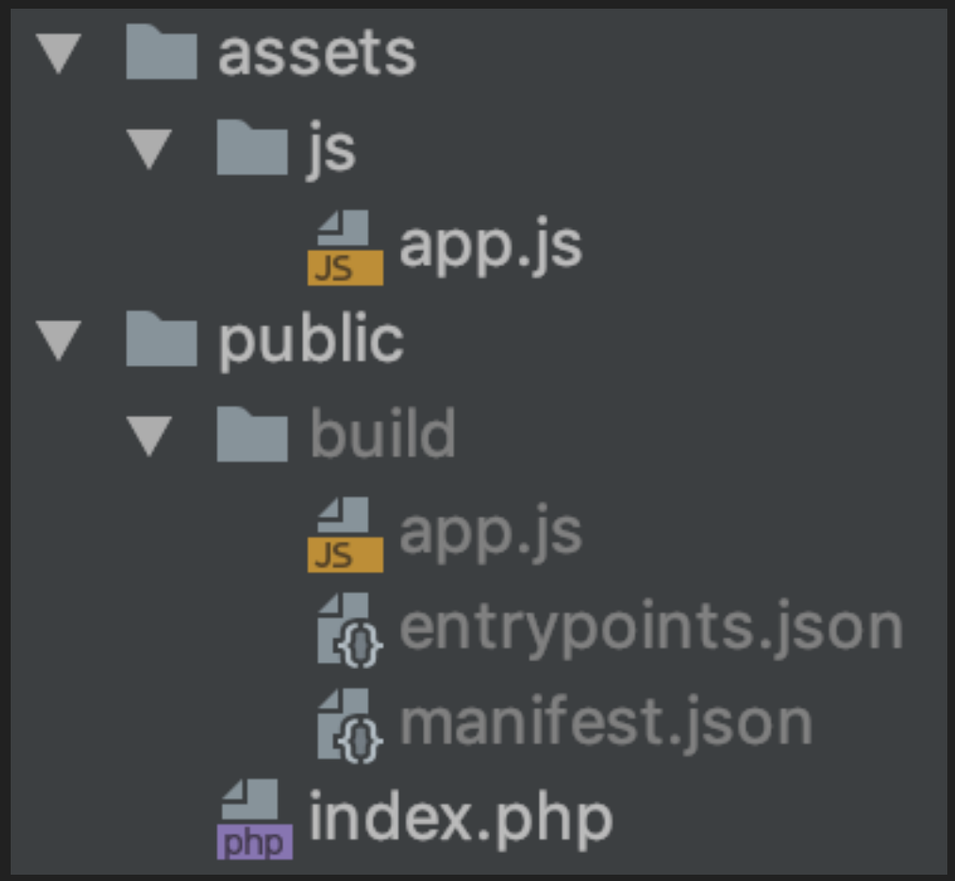
It's helper to generate standard config!

```
> yarn encore dev --watch

Running webpack ...

webpack is watching the files…

 DONE  Compiled successfully in 528ms

 I  1 files written to public/build
Entrypoint app = app.js
```

```
> yarn encore dev --watch
```

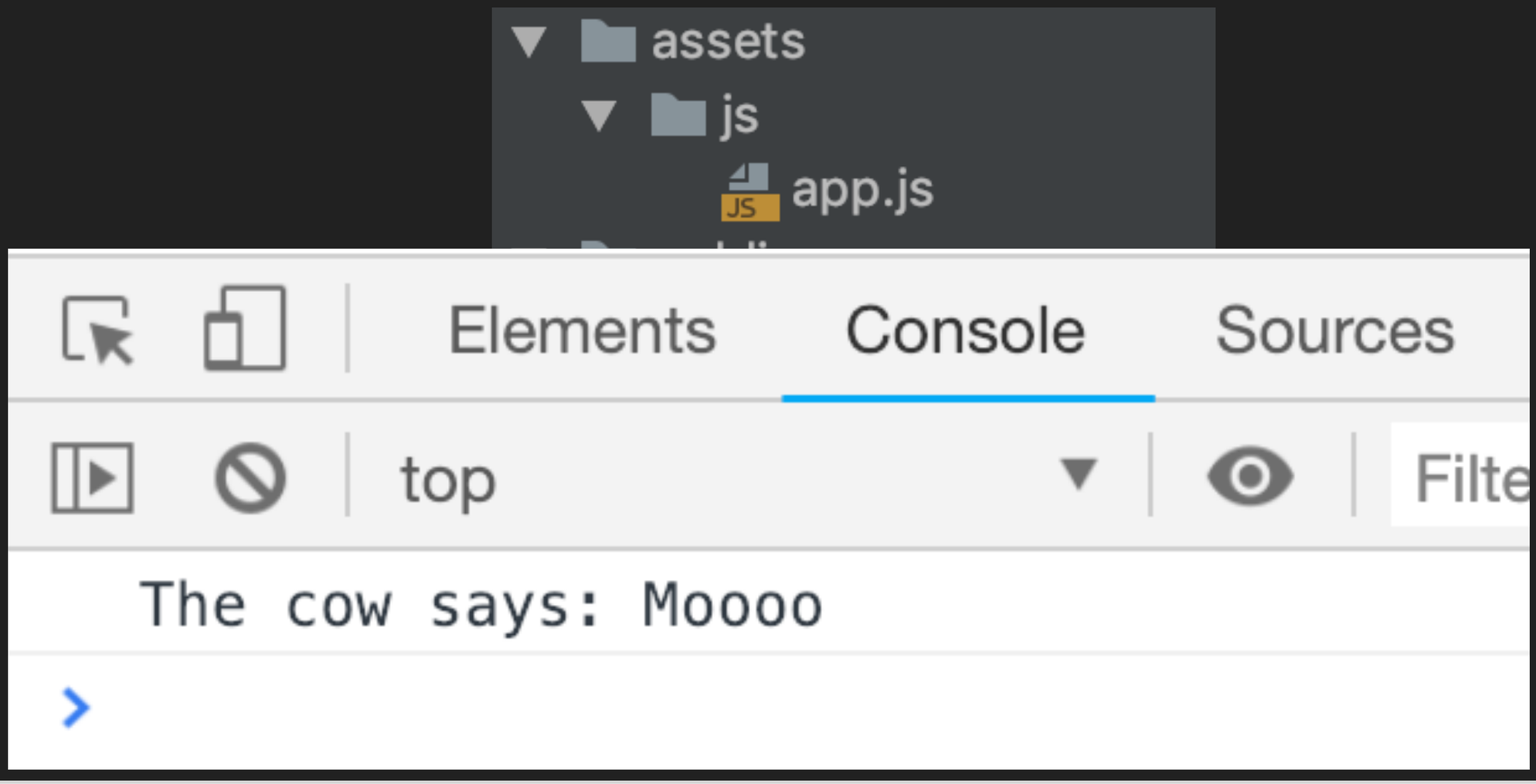


```
> yarn encore dev --watch
```




```
<!-- ... in your app -->
<script src="/build/app.js"></script>
```

```
> yarn encore dev --watch
```




```
<!-- ... in your app -->
<script src="/build/app.js"></script>
```

Need to use a third-party lib?

@weaverryan

```
> yarn add lodash --dev
```

@weaverryan

```
> yarn add lodash --dev
```

```
// assets/js/cow_say.js
import _ from 'lodash';

export default function (msg) {
    return `The cow says: ${_.lowerFirst(msg)}`;
}
```

The compiled app.js now contains lodash

```
// assets/js/cow_say.js
import lowerFirst from 'lodash/lowerFirst';

export default function (msg) {
    return `The cow says: ${lowerFirst(msg)}`;
}
```

Or keep your app.js smaller by only importing the one file your need

CSS: An un-handled dependency of your JS app

@weaverryan

Let’s print some HTML!

```
// assets/js/app.js
import cow_say from './cow_say';

document.body.innerHTML += `
    <div class="the-cow">
        ${cow_say('Moooo')}
    </div>
`;
```

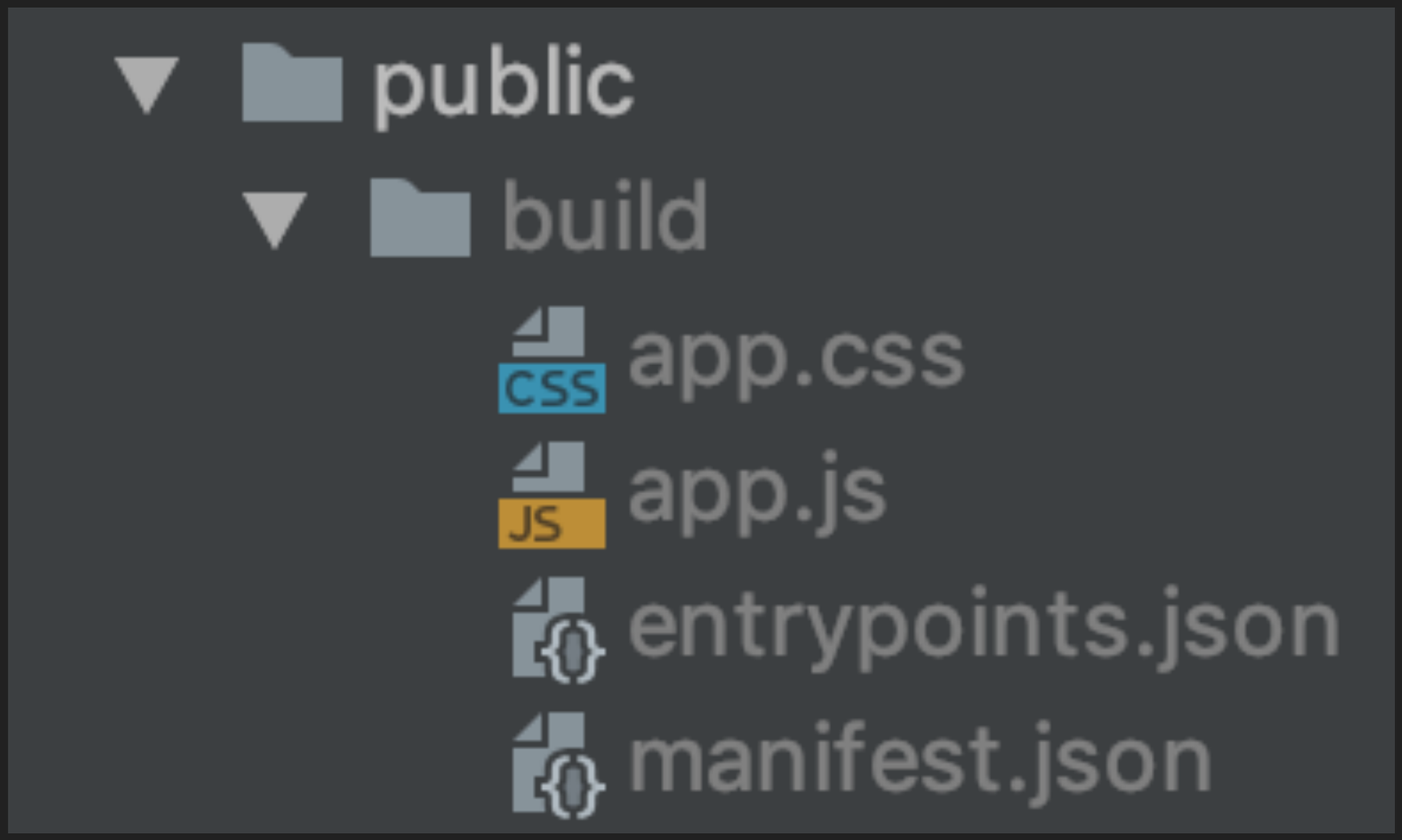
But this won’t look right unless I include the CSS for that on this page!

Let’s print some HTML!

```
// assets/js/app.js
import cow_say from ‘./cow_say';
import '../css/cow.css';

document.body.innerHTML += `
    <div class="the-cow">
        ${cow_say('Moooo')}
    </div>
`;
```

Cool: just import the CSS!

```
> yarn encore dev --watch
```



app.css holds ALL the CSS imported from ALL the JS Files

```
<link ... href="/build/app.css">
```

Want Bootstrap?

@weaverryan

```
> yarn add bootstrap --dev
```

@weaverryan

```
> yarn add bootstrap --dev
```

```
/* assets/css/cow.css */
@import "~bootstrap/dist/css/bootstrap.css";
/* ... */
```

The compiled app.css now contains Bootstrap

Stop

thinking of your JavaScript as random code that executes

Start

thinking of your JavaScript as
a single application with dependencies
that are all packaged up together

Sass instead of CSS?

@weaverryan

```
// assets/js/app.js
// ...
require('../css/cow.scss');
```

```
// assets/js/app.js
// ...
require('../css/cow.scss');
```

```
> yarn run encore dev
```

```
 ERROR  Failed to compile with 1 errors

Error loading ./assets/css/app.scss

 FIX  To load SASS files:
        1. Add Encore.enableSassLoader() to your webpack.config.js file.
        2. Install sass-loader & node-sass
               yarn add sass-loader node-sass --dev
```

@weaverryan

```
// webpack.config.js

Encore
    // ...

    .enableSassLoader()
```

```
> yarn add sass-loader node-sass --dev
```

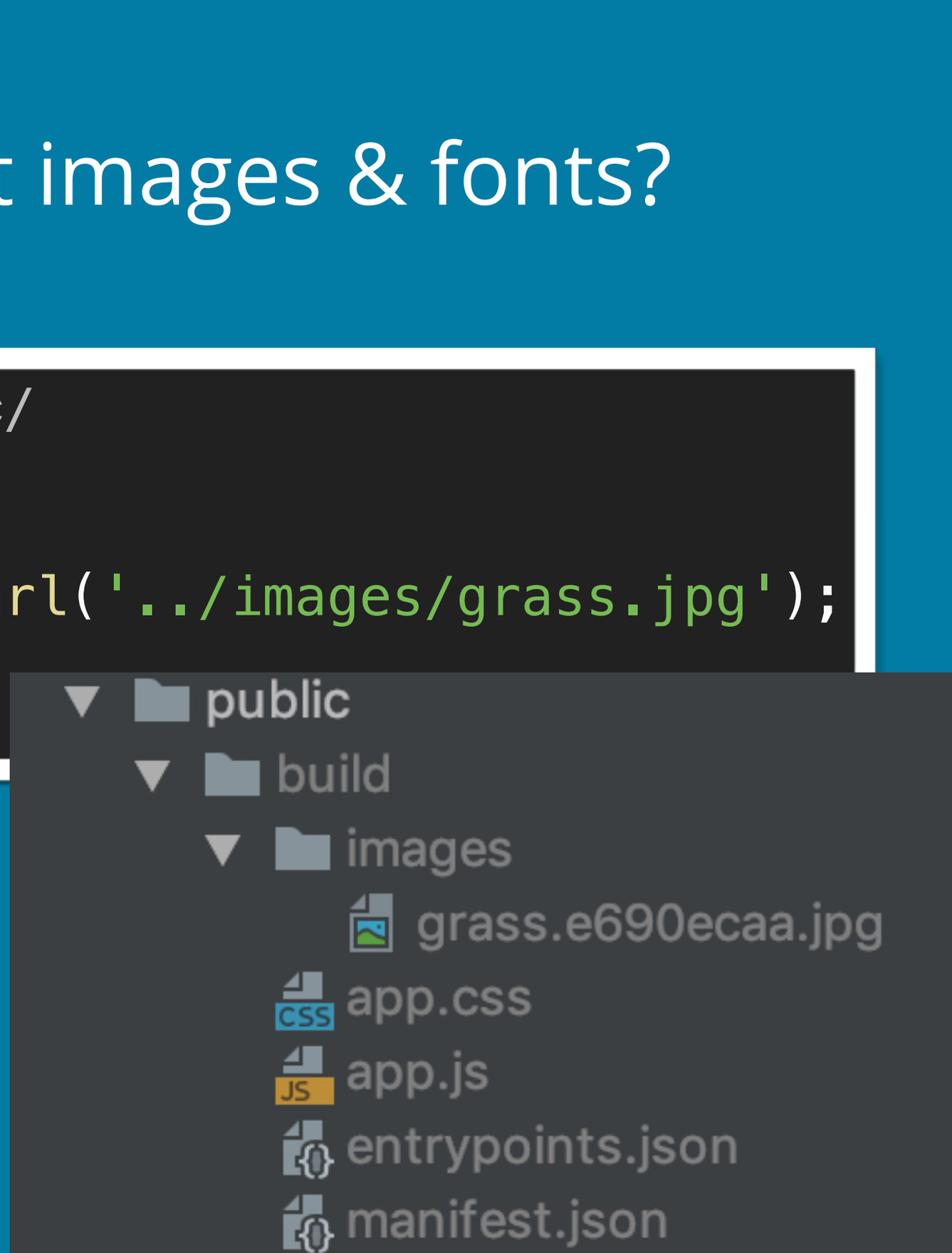
```
/* assets/css/cow.scss */
$blue: #3490dc;
@import '~bootstrap/scss/bootstrap';
```

But what about images & fonts?

```
/* assets/css/cow.css */
.grass {
    color: green;
    background-image: url('../images/grass.jpg');
}
```

But what about images & fonts?

```
/* assets/css/cow.css */
.grass {
    color: green;
    background-image: url('../images/grass.jpg');
}
```



But I want page-specific CSS and JS files!

@weaverryan

```
// assets/js/checkout.js
import '../css/checkout.css'
import $ from 'jquery';

// ...
```

```
// webpack.config.js

Encore
    // ...

    .addEntry('app', './assets/js/app.js')
    .addEntry('checkout', ‘./assets/js/checkout.js’)
```

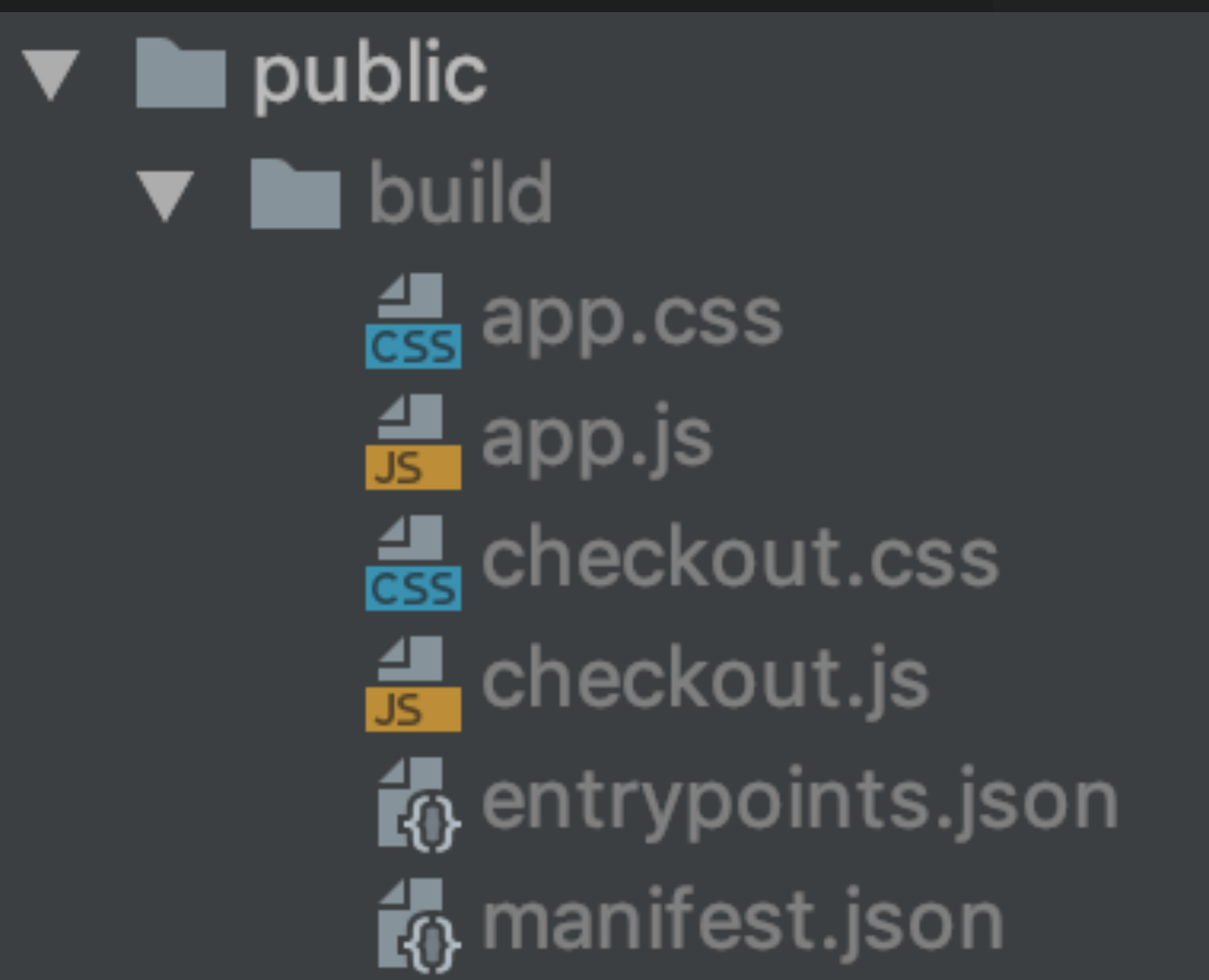
```
<!-- on the checkout page only -->

<link href="/build/checkout.css">

<script src=“/build/checkout.js"></script>
```

```
// webpack.config.js

Encore
    // ...

    .addEntry('app', './assets/js/app.js')
    .addEntry('checkout', ‘./assets/js/checkout.js’)
```

```
<!-- on the checkout page only -->

<link href="/build/checkout.css">

<script src=“/build/checkout.js"></script>
```

> yarn encore dev --watch

```
 DONE  Compiled successfully in 3676ms

 I  5 files written to public/build
Entrypoint app [big] = app.css app.js
Entrypoint checkout [big] = checkout.css checkout.js
```



What else can I do?

Automatic Features

- babel transpiling
- dev server
- production optimizations
- CSS handling
- browserslist support

Opt-in Features

```
Encore
    .enableVersioning()
    .copyFiles()
    .enablePostCssLoader()
    .enableReactPreset()
    .enablePreactPreset()
    .enableTypeScriptLoader()
    .enableForkedTypeScriptTypesChecking()
    .enableVueLoader()
    .enableEslintLoader()
    .enableBuildNotifications()
    .enableHandlebarsLoader()
    .cleanupOutputBeforeBuild()
;
```

Plus anything else Webpack can do

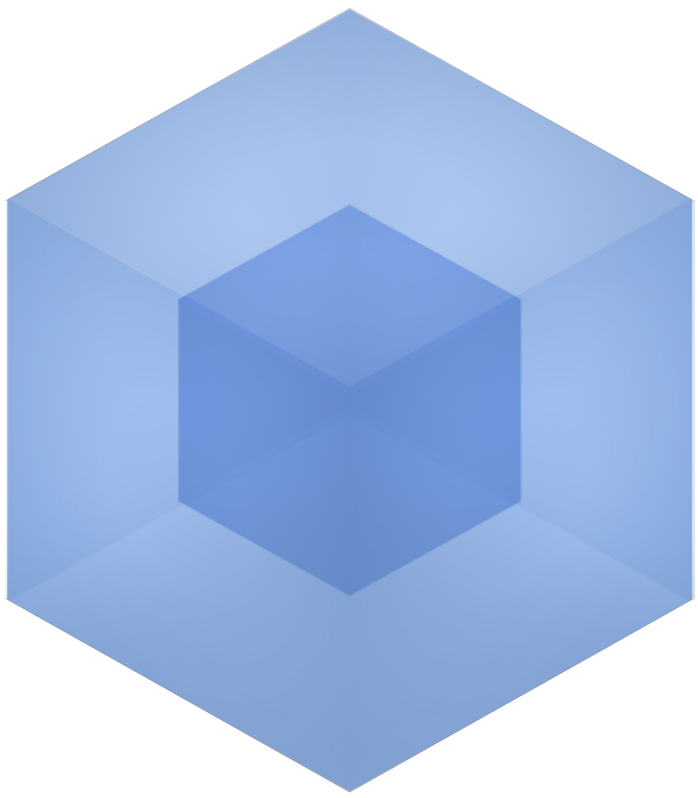
… because it’s 100% Webpack under the hood!

PART 3

Creating an Efficient Build

Production & Code Splitting

@weaverryan



Yo! Don’t worry, I’ll build and wire up the final code correctly for you. It’s going to be awesome!

> yarn encore dev --watch

```
   288 Jan  9 13:23 .
   128 Jan  9 13:29 ..
705229 Jan  9 13:30 app.css
 57209 Jan  9 13:32 app.js
   384 Jan  9 13:30 checkout.css
757367 Jan  9 13:32 checkout.js
   270 Jan  9 13:30 entrypoints.json
    96 Jan  9 13:14 images
   230 Jan  9 13:32 manifest.json
```

> yarn encore production

```
   288 Jan  9 13:23 .
   128 Jan  9 13:29 ..
137990 Jan  9 13:32 app.css
  4299 Jan  9 13:32 app.js
    26 Jan  9 13:32 checkout.css
 88212 Jan  9 13:32 checkout.js
   270 Jan  9 13:32 entrypoints.json
    96 Jan  9 13:14 images
   230 Jan  9 13:32 manifest.json
```

But what if multiple entries import the same module?

```
// assets/js/app.js
import React from 'react';
// ...

// assets/js/checkout.js
import React from 'react';

// ...
```

> yarn encore production

```
   272 Oct 11 17:08 .
   136 Oct 11 09:54 ..
140218 Oct 11 17:08 app.css
168785 Oct 11 17:08 app.js
    30 Oct 11 17:08 checkout.css
193197 Oct 11 17:08 checkout.js
```

> yarn encore production

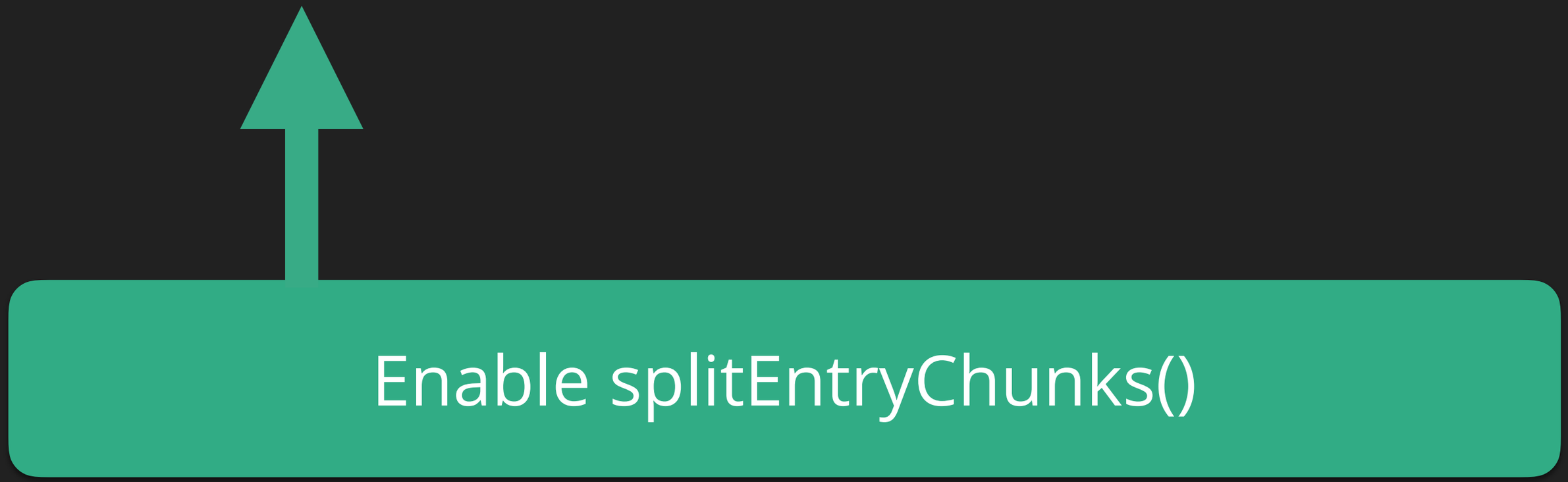
```
   272 Oct 11 17:08 .
   136 Oct 11 09:54 ..
140218 Oct 11 17:08 app.css
168785 Oct 11 17:08 app.js
    30 Oct 11 17:08 checkout.css
193197 Oct 11 17:08 checkout.js
```

Woh! So many Reacts!

So now what?

@weaverryan

```
// webpack.config.js
Encore
    // ...
    .addEntry('app', './assets/js/app.js')
    .addEntry('checkout', ‘./assets/js/checkout.js')
    .splitEntryChunks()
;
```



> yarn encore production

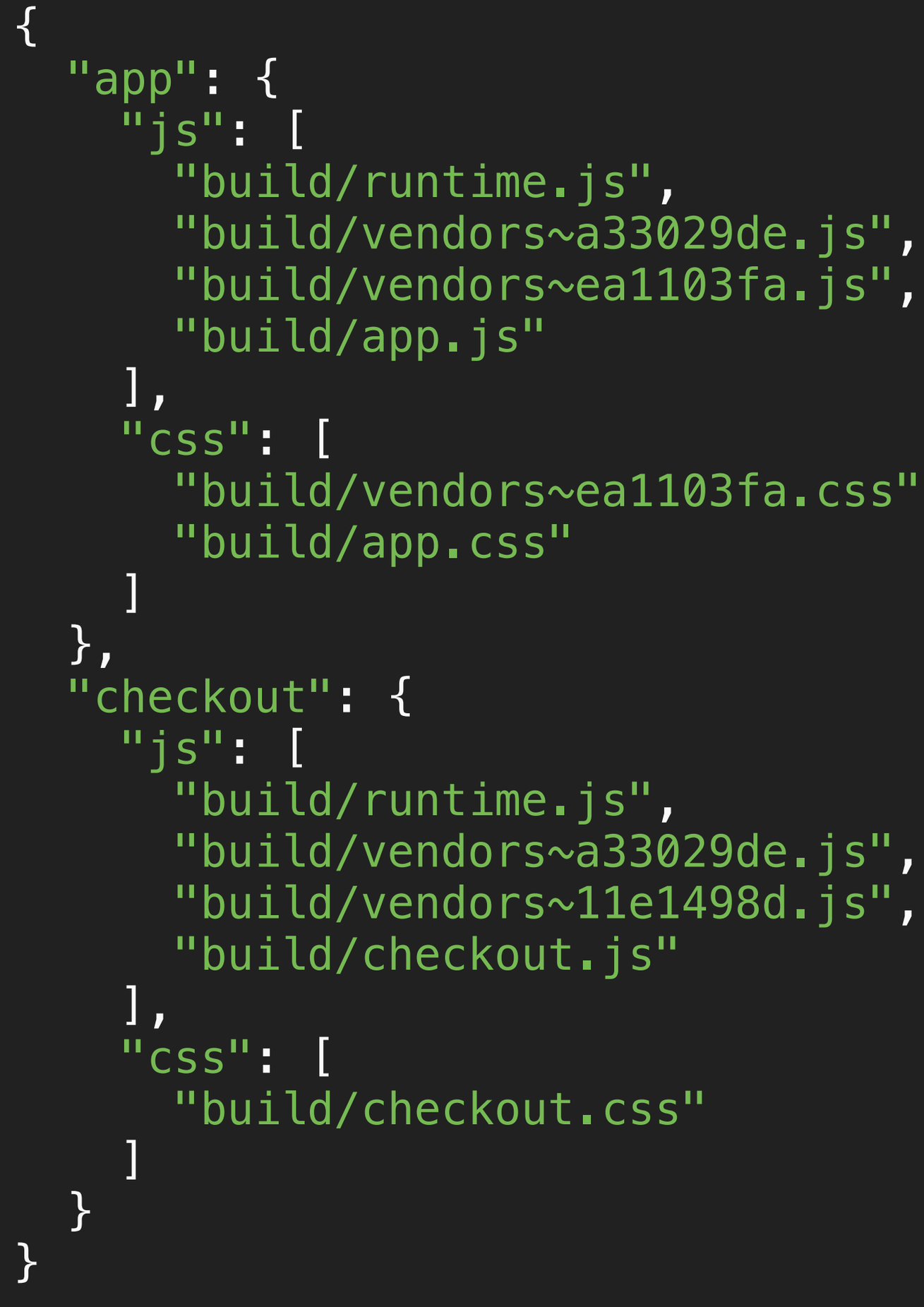
```
Running webpack ...

 DONE  Compiled successfully in 10591ms                       10:32:49 AM

 I  10 files written to public/build
Entrypoint app [big] = runtime.js vendors~a33029de.js vendors~ea1103fa.css vendors~ea1103fa.js app.css app.js
Entrypoint checkout = runtime.js vendors~a33029de.js vendors~11e1498d.js checkout.css checkout.js
✨  Done in 13.24s.
```

```
<script src="/build/runtime.js"></script>
<script src="/build/vendor~a33029de.js"></script>
<script src="/build/vendor~ea1103fa.js"></script>
<script src="/build/app.js"></script>
```

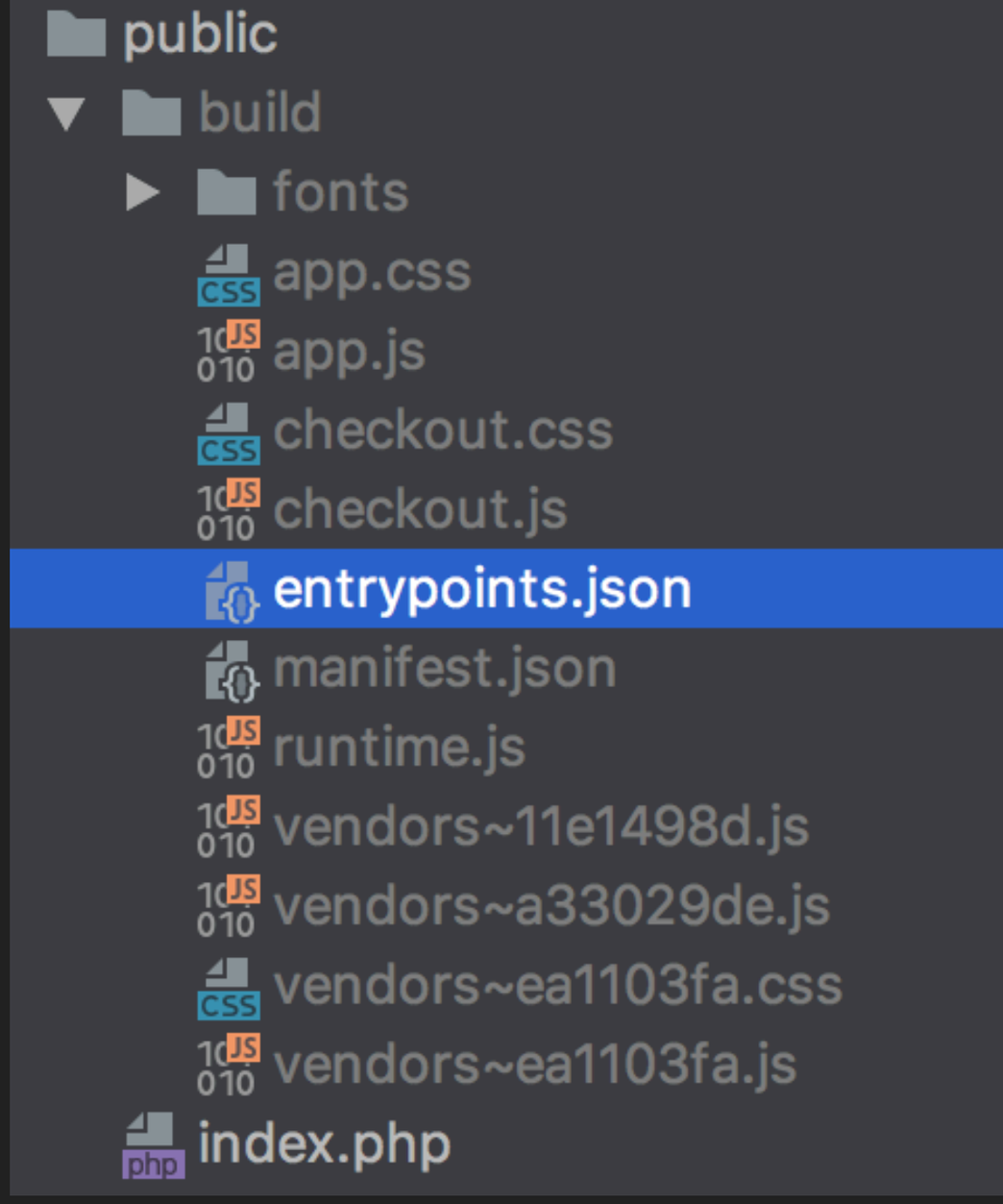
All of these files are needed to for the “app” entry to function

```
<script src="/build/runtime.js"></script>
<script src="/build/vendor~a33029de.js"></script>
<script src="/build/vendor~ea1103fa.js"></script>
<script src="/build/app.js"></script>
```

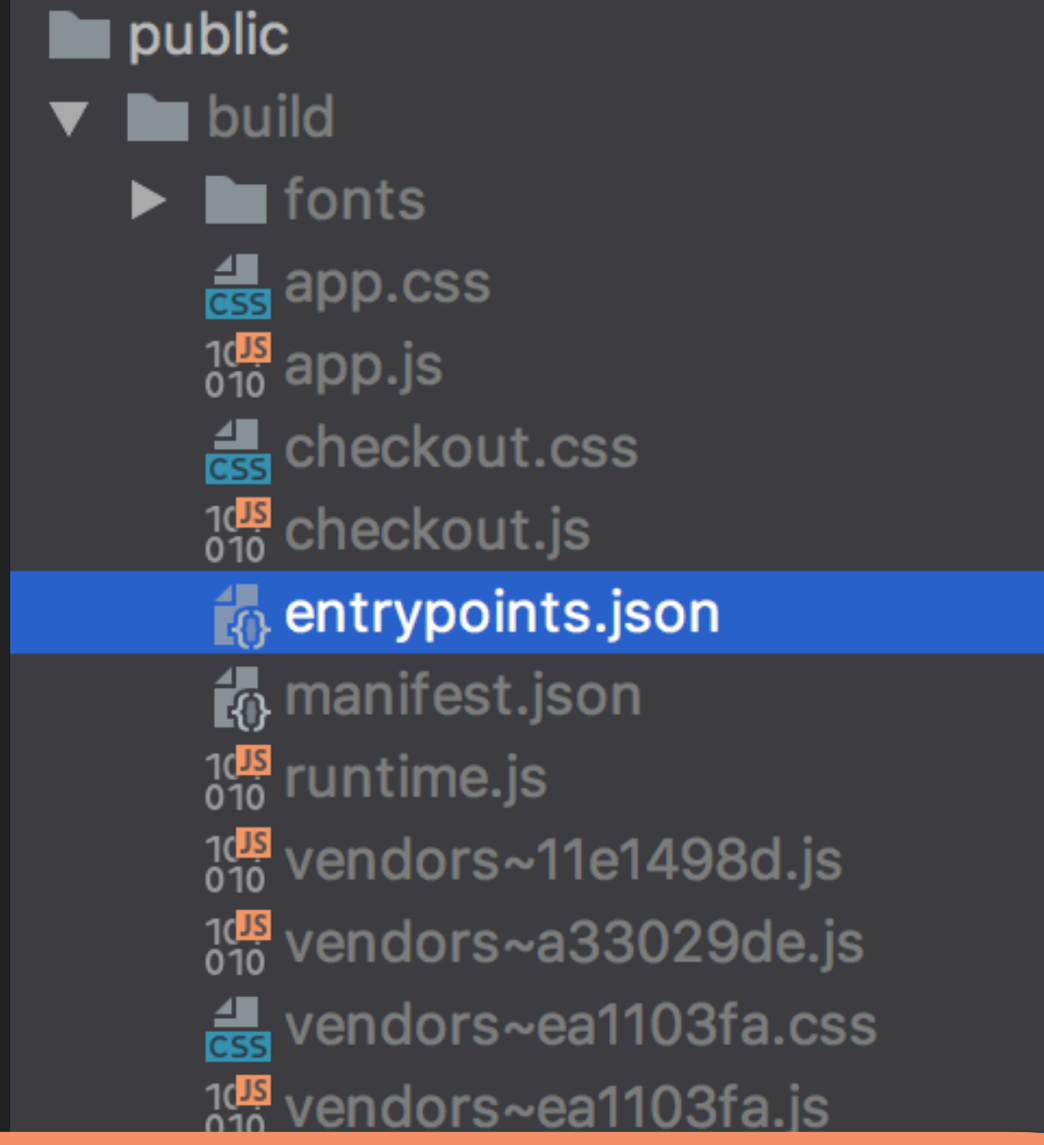
How are we supposed to know what script & link tags we need?



```
{
  "app": {
    "js": [
      "build/runtime.js",
      "build/vendors~a33029de.js",
      "build/vendors~ea1103fa.js",
      "build/app.js"
    ],
    "css": [
      "build/vendors~ea1103fa.css",
      "build/app.css"
    ]
  },
  "checkout": {
    "js": [
      "build/runtime.js",
      "build/vendors~a33029de.js",
      "build/vendors~11e1498d.js",
      "build/checkout.js"
    ],
    "css": [
      "build/checkout.css"
    ]
  }
}
```



```
{
  "app": {
    "js": [
      "build/runtime.js",
      "build/vendors~a33029de.js",
      "build/vendors~ea1103fa.js",
      "build/app.js"
    ],
    "css": [
      "build/vendors~ea1103fa.css",
      "build/app.css"
    ]
  },
"checkout": {
  "js": [
```



Homework: write a function that reads this and outputs the correct script & link tags

For Example:
In Symfony

WebpackEncoreBundle

@weaverryan

```
{# base layout #}

{{ encore_entry_link_tags('app') }}

{{ encore_entry_script_tags('app') }}
```

```
{# checkout page #}

{{ encore_entry_link_tags('checkout') }}

{{ encore_entry_script_tags(‘checkout') }}
```

Dynamic Code Splitting

import()

@weaverryan

```
// assets/js/app.js
import linker from './common/external_linker';

$('a.external').on('click', function(e) {
    linker(e.currentTarget);
});
```

```
// assets/js/common/external_linker.js
import $ from 'jquery';
import '../../css/external_link.css';

export default function(linkEl) {
    $(linkEl).addClass('clicked');
}
```

```
// assets/js/app.js
import linker from './common/external_linker';

$('a.external').on('click', function(e) {
    linker(e.currentTarget);
});
```

JS & CSS from external_linker is *always* packaged into built files.

```
// assets/js/common/external_linker.js
import $ from 'jquery';
import '../../css/external_link.css';

export default function(linkEl) {
    $(linkEl).addClass('clicked');
}
```

```
// assets/js/app.js
$('a.external').on('click', function(e) {
    import('./common/external_linker').then(linker => {
        linker.default(e.currentTarget);
    });
});
```

```
// assets/js/common/external_linker.js
import $ from 'jquery';
import '../../css/external_link.css';

export default function(linkEl) {
    $(linkEl).addClass('clicked');
}
```

```
// assets/js/app.js
$('a.external').on('click', function(e) {
    import('./common/external_linker').then(linker => {
        linker.default(e.currentTarget);
    });
});
```

The code from external_linker.js will not be included in the app.js

```
// assets/js/common/external_linker.js
import $ from 'jquery';
import '../../css/external_link.css';

export default function(linkEl) {
    $(linkEl).addClass('clicked');
}
```

```
// assets/js/app.js
$('a.external').on('click', function(e) {
    import('./common/external_linker').then(linker => {
        linker.default(e.currentTarget);
    });
});
```

It will be loaded via AJAX when needed (including the CSS file!)

```
// assets/js/common/external_linker.js
import $ from 'jquery';
import '../../css/external_link.css';

export default function(linkEl) {
    $(linkEl).addClass('clicked');
}
```

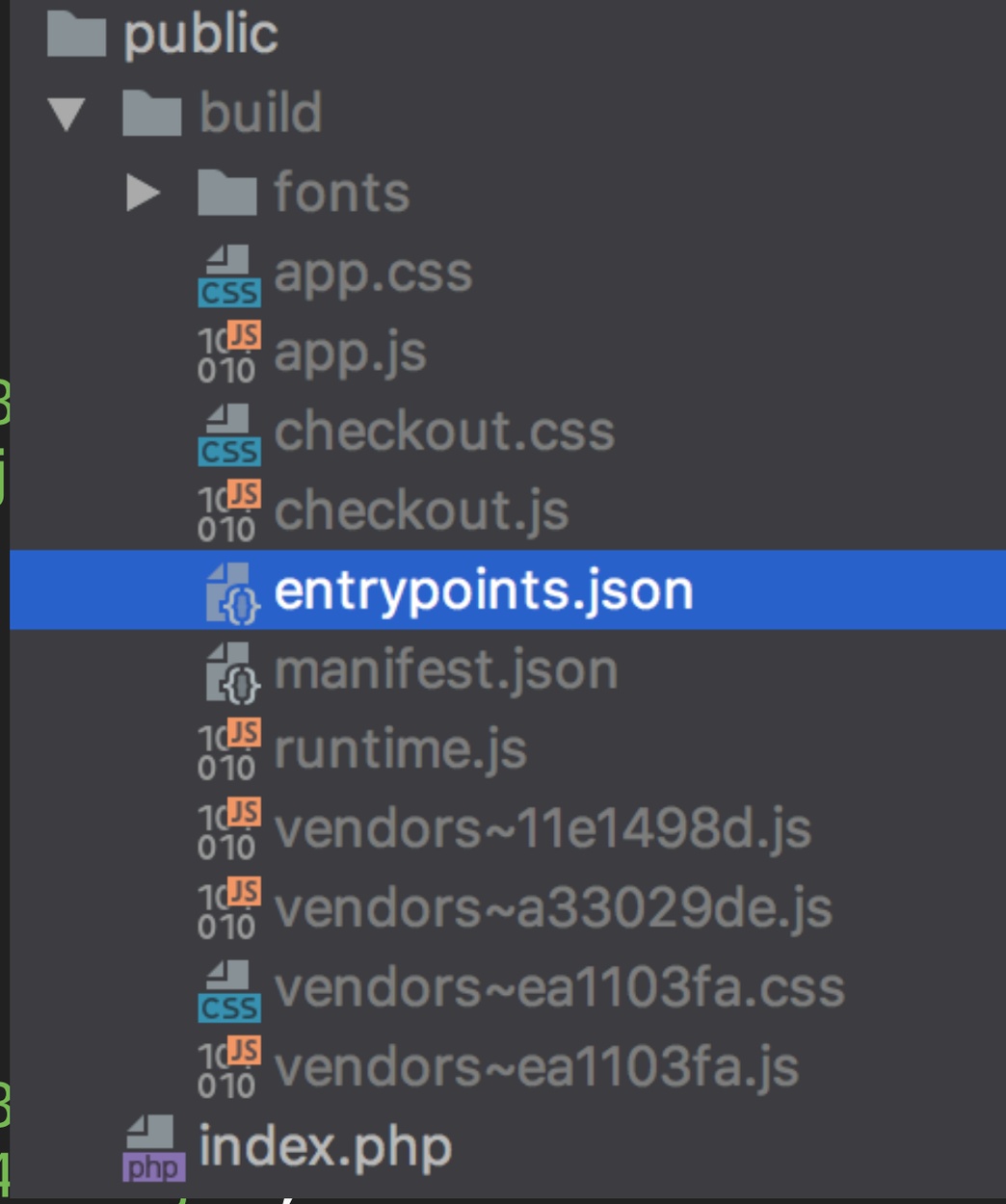
Versioning?

@weaverryan

```
// webpack.config.js
Encore
    // ...
+   .enableVersioning()
;
```

... and it just works!

```
{
  "entrypoints": {
    "app": {
      "js": [
        "/build/vendors~app~checkout.8
        "/build/vendors~app.6ef1ef8c.j
        "/build/app.245eca57.js"
      ],
      "css": [
        "/build/app.36aa118d.css"
      ]
    },
    "checkout": {
      "js": [
        "/build/vendors~app~checkout.8
        "/build/vendors~checkout.4bf64
        "/build/checkout.f2208d73.js"
      ],
      "css": [
        "/build/checkout.24df7f0c.css"
      ]
    }
  }
}
```

Putting it all together

@weaverryan

ES6/ES2015/ECMAScript 2015

"Newer" version of JavaScript,
not supported by all browsers

@weaverryan

Babel

A tool that can transform JavaScript to different JavaScript

A) ES6 js to “old” JS
B) React to raw JS

Webpack

A tool that follows imports to bundle JavaScript, CSS, and anything else you dream up into one JavaScript package

@weaverryan

Webpack Encore

A tool that makes configuring
webpack *not* suck

@weaverryan

So go write some amazing JavaScript!

Thank You!

Ryan Weaver
@weaverryan

https://symfonycasts.com/encore

SymfonyCasts

CodeMash