

webpack

101

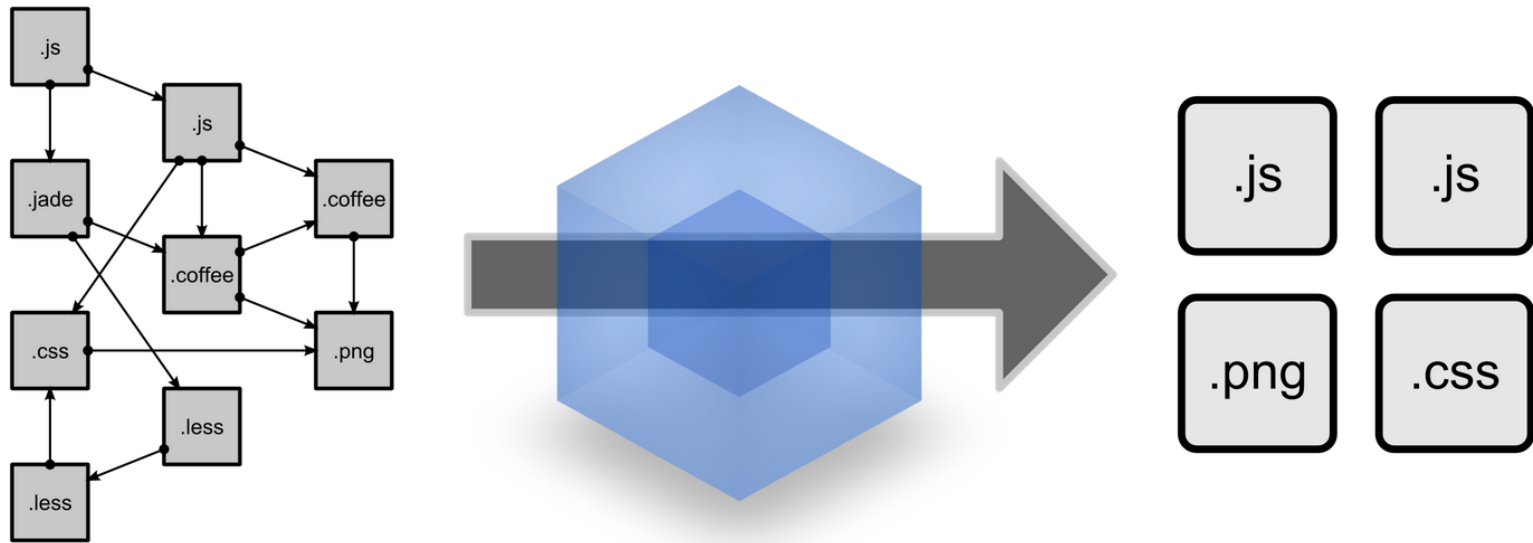
purpose

transform and bundle your files

where it differs...philosophy

- everything is a module
- load only “what” you need and “when” you need it

how does it work?



btw...webpack is hard

It has terrible documentation

terminology

- loaders
- plugins

3 main things webpack needs to know

1. the starting point of your application
2. which transformations to make on your code
3. where it should save the new transformed code

config

- webpack.config.dev.js
- webpack.config.prod.js

1. specify an entry point

```
// In webpack.config.js
module.exports = {
  entry: [
    './app/index.js'
  ]
}
```

2. add a loader

```
// npm install --save-dev babel-loader
// In webpack.config.js
module.exports = {
  entry: [
    './app/index.js'
  ],
  module: {
    loaders: [
      {test: /\.js$/, exclude: /node_modules/, loader: "babel-loader"}
    ]
  }
}
```

3. specify the output

```
// In webpack.config.js
module.exports = {
  entry: [
    './app/index.js'
  ],
  module: {
    loaders: [
      {test: /\.js$/, exclude: /node_modules/, loader: "babel-loader"}
    ]
  },
  output: {
    path: __dirname + '/dist',
    filename: "index_bundle.js"
  }
}
```

install and run webpack

```
// npm install webpack webpack-dev-server
// In package.json
{
  "name": "Foo Bar",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "build": "webpack -p",
    "start": "webpack-dev-server"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "react": "^15.3.2",
    "react-dom": "^15.3.2"
  },
  "devDependencies": {
    "babel-core": "^6.18.2",
    "babel-loader": "^6.2.7",
    "babel-preset-react": "^6.16.0",
    "html-webpack-plugin": "^2.24.1",
    "webpack": "^1.13.3",
    "webpack-dev-server": "^1.16.2"
  }
}
```

bundle created!

```
/app
- components
- containers
- config
- utils
index.js
index.html
/dist
  index_bundle.js
package.json
webpack.config.js
.gitignore
```

HtmlWebpackPlugin

```
// npm install --save-dev html-webpack-plugin
// In webpack.config.js
var HtmlWebpackPlugin = require('html-webpack-plugin')
var HTMLWebpackPluginConfig = new HtmlWebpackPlugin({
  template: __dirname + '/app/index.html',
  filename: 'index.html',
  inject: 'body'
});
module.exports = {
  entry: [
    './app/index.js'
  ],
  module: {
    loaders: [
      {test: /\.js$/, exclude: /node_modules/, loader: "babel-loader"}
    ]
  },
  output: {
    path: __dirname + '/dist',
    filename: "index_bundle.js"
  },
  plugins: [HTMLWebpackPluginConfig]
}
```

html template

app/index.html

```
<!DOCTYPE html>
<html lang= "en">
<head>
  <meta charset= "UTF-8">
  <title>My App</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id= "app"></div>
</body>
</html>
```

dist/index.html

```
<!DOCTYPE html>
<html lang= "en">
<head>
  <meta charset= "UTF-8">
  <title>My App</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id= "app"></div>
  <script src="index_bundle.js"></ script>
</body>
</html>
```

now some css

- **css-loader** takes a CSS file and reads off all its dependencies - supercharges require/import for CSS
- **style-loader** will embed those styles directly into the markup

how it works

style.css

```
.element {  
  width: 100px;  
  height: 100px;  
}
```

webpack.config.js

```
//npm install --save-dev style-loader css-loader  
module: {  
  loaders: [  
    {test: /\.css$/, loader: "style!css"}  
  ],  
},
```

index.js

```
var React = require('react');  
var ReactDOM = require('react-dom');  
require('./style.css')  
  
var ProfilePic = React.createClass({  
  render: function() {  
    const {imageUrl} = this.props;  
    return (  
      <img src={imageUrl}  
        className="element" />  
    )  
  }  
});
```

dist/index_bundle.js

```
function(module, exports, __webpack_require__) {  
  
  // module  
  exports.push([module.id,  
    ".element {\n\twidth: 100px;\n\theight: 100px;\n}\n",  
    ""]);  
  
  /***/ }  
}
```

ExtractTextPlugin

webpack.config.js

```
//npm install --save-dev extract-text-webpack-plugin
var ExtractTextPlugin = require("extract-text-webpack-plugin" );
...
module: {
  loaders: [
    { test: /\.css$/, loader: ExtractTextPlugin.extract( "css" ) }
  ]
},
plugins: [
  new ExtractTextPlugin( "styles.css" )
]
```

dist/styles.css

```
.element {
  width: 100px;
  height: 100px;
}
```

dist/index.html

```
<html lang= "en">
<head>
  <meta charset= "UTF-8">
  <title>Github Battle< /title>
  <link href="styles.css" rel="stylesheet">
</head>
<body>
```

what you (may) want

- css modules

- local scope by default
- unique, dynamically generated classes
- new classes mapped to correct styles
- keep your styles with your component

how it works

webpack.config.js

```
module: {  
  loaders: [  
    { test: /\.css$/, loader: ExtractTextPlugin.extract(  
      'css?modules&importLoaders=1&localIdentName=[name]__[local]__[hash:base64:5]'  
    )}  
  ]  
}
```

index.js

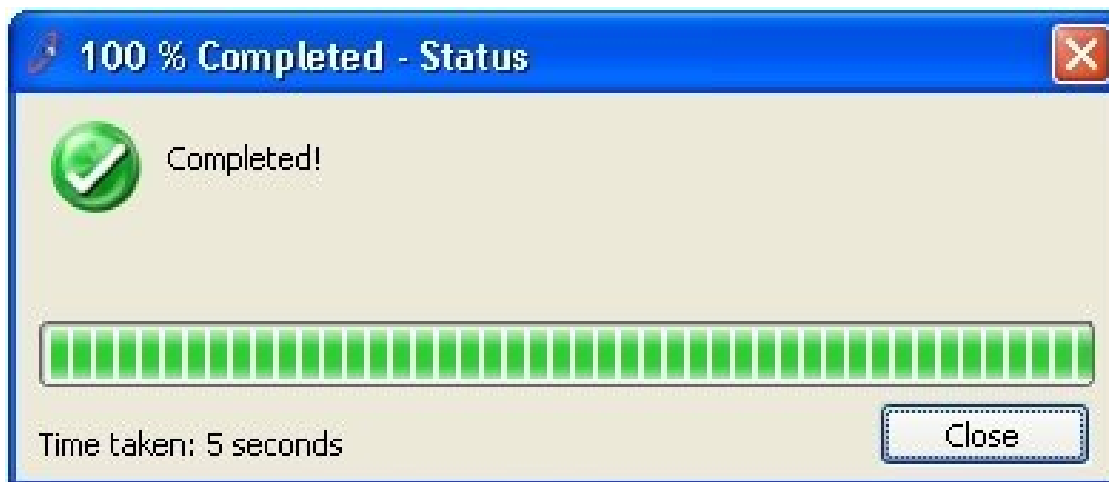
```
var style = require('./style.css')  
  
var ProfilePic = React.createClass({  
  render: function() {  
    const {imageUrl} = this.props;  
    return (  
      <img src={imageUrl}  
        className={style.element}/>  
    )  
  }  
});
```

dist/styles.css

```
.style__element__lGfyX {  
  width: 100px;  
  height: 100px;  
}
```

a modular environment

```
/components
  Head.js
  /Button
    Button.js
    styles.css
  /Input
    Input.js
    style.css
  /Title
    Title.js
    style.css
```



further reading

- projects using webpack:
<https://github.com/AusDTO/dto-digitalmarketplace-frontend/tree/master/config>
- optimization
- code splitting
- **webpack v2!!**

credits

- <https://github.com/nathf>
- <https://reacttraining.com/>
- <https://medium.com/@rajaraodv/webpack-the-confusing-parts-58712f8fcad9#.78xauw8tp>
- <https://css-tricks.com/css-modules-part-1-need/>

the end