Webpack 5.x.x

Maksim Kirylka
Senior Software Engineer at EPAM LT

RS School   2021
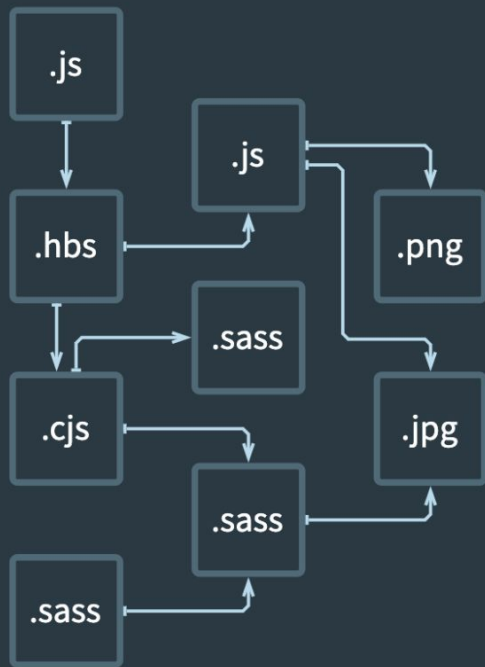
Definition

What is webpack

a static module bundler which runs through all the code and its dependencies starting from entry point and joins them into a bundle.
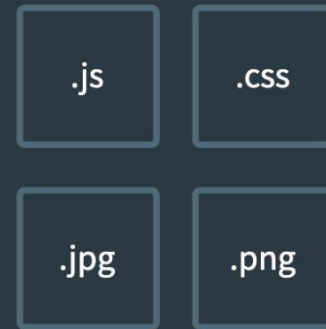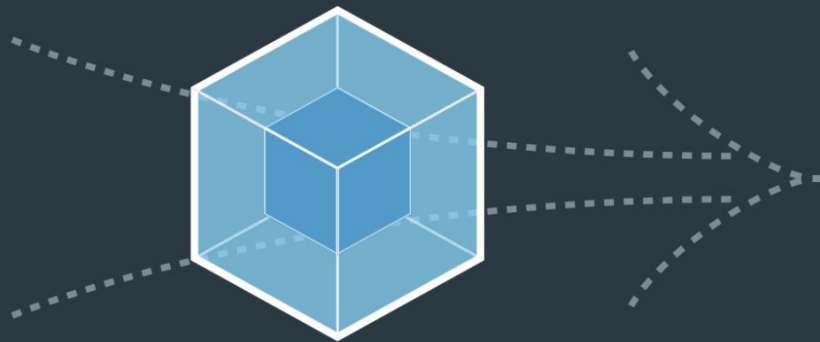
The reference to the documentation webpack.js.org

bundle your scripts

MODULES WITH DEPENDENCIES

STATIC ASSETS

Advantages

- helps bundle javascript and static assets

```
1  <html>
2      <script src="/src/main.js"></script>
3      <script src="/src/auth.js"></script>
4      <script src="/src/home.js"></script>
5      <script src="/src/http-request.js"></script>
6      <script src="/src/utils.js"></script>
7  </html>
```

```
1  <html>
2      <script src="/dist/bundle.js"></script>
3  </html>
```

Advantages

- helps manage the dependency relationship and load your assets in dependency order

```
1  // CommonJS
2  const path = require('./path');
3  module.exports = path;
4
5  // ES Modules
6  import path from './path';
7  export default path;
8
```

Advantages

- using loaders webpack can transpile code with the latest features to ES5 syntax to enable old browsers support

```
1  const all = [...[1, 10, 100], ...[-1, -10, -100]];
2
3  const value = null ?? 0;
4
5  const sum = (arr) => {
6    return arr.reduce((acc, value) => acc + value, 0);
7  }
8
9
10
11
12
13
```

```
1  "use strict";
2
3  require("core-js/modules/es.array.concat.js");
4
5  var _ref;
6
7  var all = [1, 10, 100].concat([-1, -10, -100]);
8  var value = (_ref = null) !== null && _ref !== void 0 ? _ref : 0;
9
10 var sum = function sum(arr) {
11   return arr.reduce(function (acc, value) {
12     return acc + value;
13   }, 0);
14 };
15
```

Advantages

- improve developers experience running dev server and watching for changes

Advantages

- code minification and obfuscation

Advantages

- tree shaking

Advantages

- enable micro-frontends approach

Source control   Build and test pipeline   Production

Micro frontend A

Micro frontend B

Micro frontend C

Three apps composed into one in production
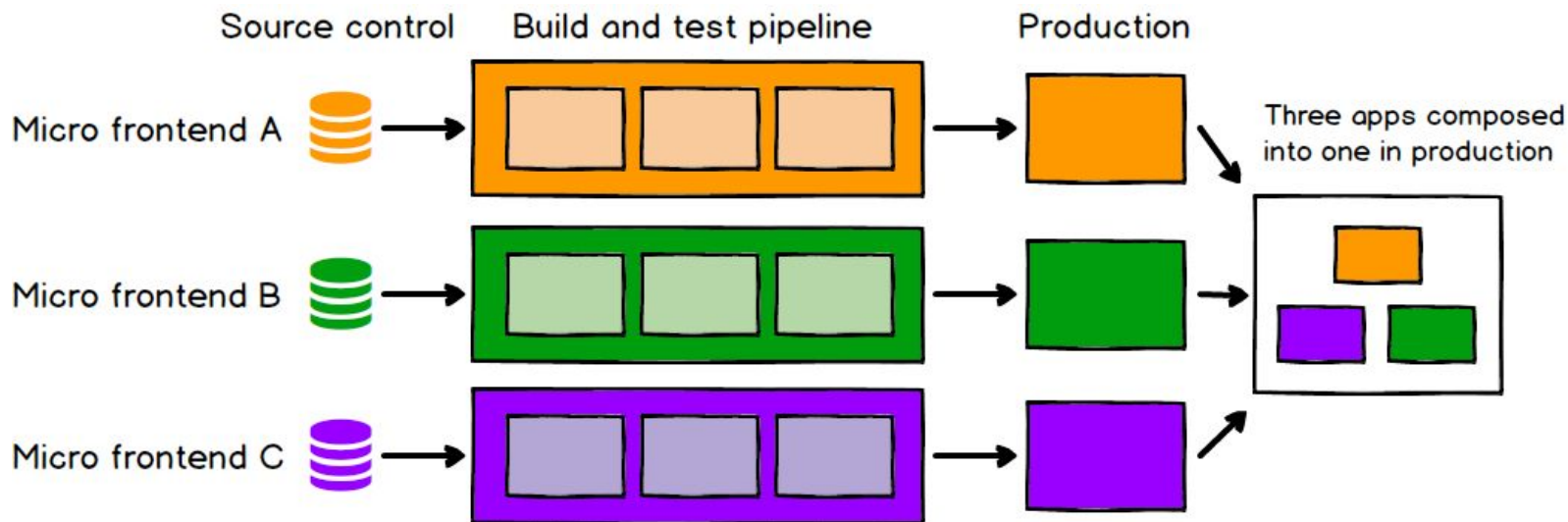
Why should we use webpack?

- helps bundle javascript and static assets
- helps manage the dependency relationship and load your assets in dependency order
- using loaders webpack can transpile code with the latest features to ES5 syntax to enable old browsers support
- improve developers experience running dev server and watching for changes
- code minification and obfuscation
- tree shaking
- enable micro-frontend approach

Simple config

No config is needed*

* There are limitations:
- The entry point is src/index.js
- Output file is dist/main.js
- Production mode

Simple webpack.config.js

```js
const path = require('path');
const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {
    mode: 'production',
    entry: './src/index.js',
    output: {
      path: path.resolve(__dirname, 'dist'),
      filename: '[name].[hash].js'
    },
    module: {
      rules: [ /* list of loaders */]
    },
    plugins: [/* list of plugins */],
    resolve: {
        extensions: ['.js', '.ts', '...'],
        alias: {}
      },
    devtool: 'source-map',
    devServer: {
        contentBase: path.join(__dirname, 'dist'),
        port: 3000,
    },
};
```

Entry

```javascript
module.exports = {
    entry: './src/index.js',
};
```

```javascript
module.exports = {
    entry: ['./src/main_1.js', './src/main_2.js'],
};
```

```javascript
module.exports = {
    entry: {
        main: './src/main.js',
        docs: './src/docs.js',
        blog: './src/blog.js',
    },
};
```

Output

```
module.exports = {
    output: {
        path: path.resolve(__dirname, 'dist'),
        filename: 'bundle.js'
    },
};
```

Loaders

```javascript
module.exports = {
    module: {
        rules: [
            {
                test: /\.css$/,
                use: ['style-loader', 'css-loader']
            },
            {
                test: /\.(png|jpg|gif|svg)$/,
                loader: 'file-loader',
                options: {
                    name: '[name].[ext]'
                }
            }
        ]
    },
};
```

Commonly used loaders

- babel-loader Loads ES6+ code and transpiles to ES5 using Babel
- file-loader resolves import on a file into a url and emits the file into the output directory.
- ts-loader Loads TypeScript 2.0+ like JavaScript
- style-loader Add exports of a module as style to DOM
- css-loader Loads CSS file with resolved imports and returns CSS code
- sass-loader Loads and compiles a SASS/SCSS file
- postcss-loader Loads and transforms a CSS/SSS file using PostCSS

Plugins

```javascript
const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {
    plugins: [
        new HtmlWebpackPlugin()
    ],
};
```

Commonly used plugins

- HtmlWebpackPlugin creates html file automatically
- MiniCssExtractPlugin extracts CSS into separate files. It creates a CSS file per JS file which contains CSS.
- TerserWebpackPlugin uses terser to minify JavaScript
- BundleAnalyzerPlugin is used to visualize size of webpack output files with an interactive zoomable treemap

Mode

```
module.exports = {
    mode: 'production',
};
```

```
webpack --mode=development
```

```
module.exports = {
    plugins: [
        new webpack.EnvironmentPlugin({
            NODE_ENV: 'development', // use 'development' unless process.env.NODE_ENV is defined
            DEBUG: false,
        })
    ],
};
```

Bundle size


```
                                              webpack ——mode=development
assets by status 9.04 KiB [cached] 1 asset
asset main.js 6.72 KiB [compared for emit] (name: main)
asset index.html 231 bytes [compared for emit]
runtime modules 1.72 KiB 5 modules
cacheable modules 491 bytes
  ./src/index.js 237 bytes [built] [code generated]
  ./src/util.js 168 bytes [built] [code generated]
  ./src/assets/image.png 86 bytes [built] [code generated]
webpack 5.38.1 compiled successfully in 298 ms
```

```
                                              webpack ——mode=production
assets by status 9.04 KiB [cached] 1 asset
asset main.js 728 bytes [emitted] [minimized] (name: main)
asset index.html 214 bytes [emitted]
orphan modules 254 bytes [orphan] 2 modules
runtime modules 1.06 KiB 2 modules
./src/index.js + 2 modules 491 bytes [built] [code generated]
webpack 5.38.1 compiled successfully in 506 ms
```

Dev Server

```
module.exports = {
    devServer: {
        contentBase: path.join(__dirname, 'dist'),
        port: 8080,
    },
};
```

Installation

npm init -y

npm i -D webpack webpack-cli webpack-dev-server

Usage

package.json

```
"scripts": {
    "build": "webpack"
},
```

```
"scripts": {
  "build": "webpack --config webpack.config.js"
}
```
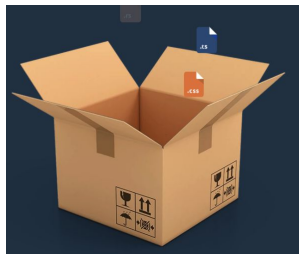
npm run build

Alternatives

Rollup

Parcel

Thank you!