

Webpack

Table des matières

1. Installation	2
2. Générer un bundle	2
3. Générer plusieurs bundles.....	2
4. Définir le format généré	2
5. Générer la page d'index (avec le bundle référencé).....	3
6. Hash	3

[Site](#)

1. Installation

```
npm i webpack -D
```

2. Générer un bundle

« **webpack.config.js** »

```
module.exports = {
  entry: './src/lib.js',
  output: {
    path: './dist',
    filename: 'bundle.js'
  }
};
```

Référencer le bundle dans la page d'index

```
<script src="dist/bundle.js"></script>
```

3. Générer plusieurs bundles

On pourrait par exemple avoir un bundle « app », un autre « vendor » comme Angular 2

« **webpack.config.js** »

```
module.exports = {
  entry: {
    app: './src/lib.js',
    other: ['./src/other.js'],
    item: ['./src/item.js']
  },
  output: {
    path: './dist',
    filename: '[name].js'
  }
};
```

4. Définir le format généré

```
module.exports = {
  entry: {
    app: './src/lib.js',
    other: ['./src/other.js'],
    item: ['./src/item.js']
  },
  output: {
    path: './dist',
    filename: '[name].js',
    libraryTarget: 'commonjs2'
  }
};
```

5. Générer la page d'index (avec le bundle référencé)

```
npm i html-webpack-plugin -D
```

« **webpack.config.js** »

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

module.exports = {
  entry: './src/app.js',
  output: {
    filename: 'bundle.js',
    path: __dirname + '/dist'
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html'
    })
  ]
};
```

Indiquer si un template est à utiliser

Plugins

6. Hash

Le hash sera le même tant qu'aucune modification n'est apportée à un fichier.

```
▼ dist
  app.595a0453f1596e3c3048.js
  item.bf86046497ec39e90b43.js
```

```
var webpack = require('webpack');

module.exports = {
  entry: {
    app: './src/app.js',
    item: ['./src/item.js']
  },
  output: {
    path: './dist',
    filename: '[name].[chunkhash].js'
  },
  plugins: [
    new webpack.optimize.CommonsChunkPlugin({
      names: ["app", "item"]
    })
  ]
};
```