

Gestion des assets avec Webpack

Par défaut Webpack attend que vous placiez vos assets (images, font, etc...) dans le dossier `src/assets`.

Dès que vos assets sont positionnés dans ce dossier, ils seront accessibles dans votre code.

Exemple avec une image

Pour commencer, il faut créer le dossier `assets` dans `src` si ce n'est pas encore fait.

Ensuite, vous pouvez créer le dossier `src/assets/images` afin de regrouper toutes les images du projet.

Vous pouvez déposer vos images dans ce dossier. Pour cette exemple on utilisera l'image `logo.png`.

Le chemin complet de l'image depuis la racine du projet est donc : `src/assets/images/logo.png`.

On peut dès à présent inclure l'image dans le css, par exemple avec une `background-image` :

```
background-image: url('./assets/images/logo.png');
```

En ajoutant cette propriété css sur un élément html vous pourrez obtenir l'image en fond de l'élément HTML. (Pour plus d'info sur `background-image` vous pouvez vous référer à la documentation sur [MDN](#))

Supplément pour pouvoir utiliser les assets dans `index.html`

Afin de pouvoir utiliser nos assets dans `index.html`, nous devons ajouter un nouveau loader webpack au projet, le `html-loader` :

```
npm install html-loader --save-dev
```

Et modifier la configuration de webpack :

webpack.config.js

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

module.exports = {
  entry: './src/main.js', // Le fichier js principale de notre code
  source,
  mode: 'development',
  output: {
    filename: 'main.js', // Le nom du bundle à générer
    path: path.resolve(__dirname, 'dist'), // Le chemin dans lequel le
    bundle doit être généré
  },
```

```
module: {
  rules: [
    {
      test: /\.scss$/,
      use: ["style-loader", "css-loader", "sass-loader"],
    },
    {
      test: /\.html$/i,
      loader: "html-loader",
    },
  ],
},
plugins: [new HtmlWebpackPlugin({template: 'src/index.html'})],
devServer: {open: true},
};
```

NOTE : Pensez à redémarrer le serveur de développement après le changement de configuration.

Vous pouvez maintenant utiliser vos assets dans `index.html`.

Pour notre image d'exemple, cela donnerait :

```

```