

Programmation Web – Avancé

JavaScript & Node.js

Partie 29 : Module bundler

Version 2020



Attribution –
Partage dans les
Mêmes Conditions
4.0 International
(CC BY-SA 4.0)

*Presentation template
by [SlidesCarnival](#)*



Module bundler

Comment gérer les dépendances de votre frontend de manière raisonnable ?



Packages Node.js utilisables côté client ?

🕒 DEMO : Problème lors de l'utilisation d'un package dans le browser via require

MyCMS aimerait gérer l'affichage d'un message aléatoire (HomePage), sur base du package **npm popular-movie-quotes** [\[97.\]](#). Cette gestion doit être purement frontend !

```
✖ ▶ Uncaught ReferenceError: require is not defined  
    at HomePage.js:1
```



Introduction aux modules (node.JS) mis à disposition du browser (ES6)

- Module bundlers :
 - Utilisation des packages faits pour Node.js via le browser
 - Utilisation de la méthode **require** côté client
 - Gestion aisée des packages et de leurs dépendances...
- Parmi les plus connus : **webpack** [\[98.\]](#),
browserify



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

- Création d'un **package.json** pour votre application (après création d'un répertoire)
 - **npm init -y**
 - Enlever le « main entry »: - **"main": "index.js"**
- Installation de **webpack** via npm :
 - **npm i webpack-cli webpack -D**



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

- Création d'une structure standard pour votre application

```
./package.json  
./webpack.config.js  
./dist  
  ./dist/index.html  
./src  
  ./src/index.js
```



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

- Intégration de votre « bundle file »
./dist/bundle.js généré par webpack au sein
de votre fichier HTML

```
<script src="./bundle.js" type="module"></script>
```



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

- Générer le bundle sans fichier de configuration
: **npx webpack**



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

- Création d'un fichier de configuration
webpack.config.js à la racine :

```
const path = require("path");

module.exports = {
  entry: "./src/index.js",
  output: {
    path: __dirname + "/dist",
    filename: "bundle.js",
  },
};
```

```
./package.json
./webpack.config.js
./dist
  ./dist/index.html
./src
  ./src/index.js
```



Introduction aux modules (node.JS) mis à disposition du browser (ES6): webpack

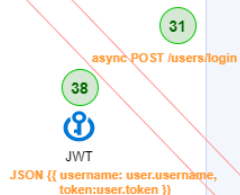
- NPM scripts : utiliser **npm run dev** ou **npm run build** au lieu de **webpack-cli**. Dans **./package.json**
 - + **"build": "webpack" ou**
 - + **"build": "webpack --mode production"**
 - + **"dev": "webpack --mode development"**



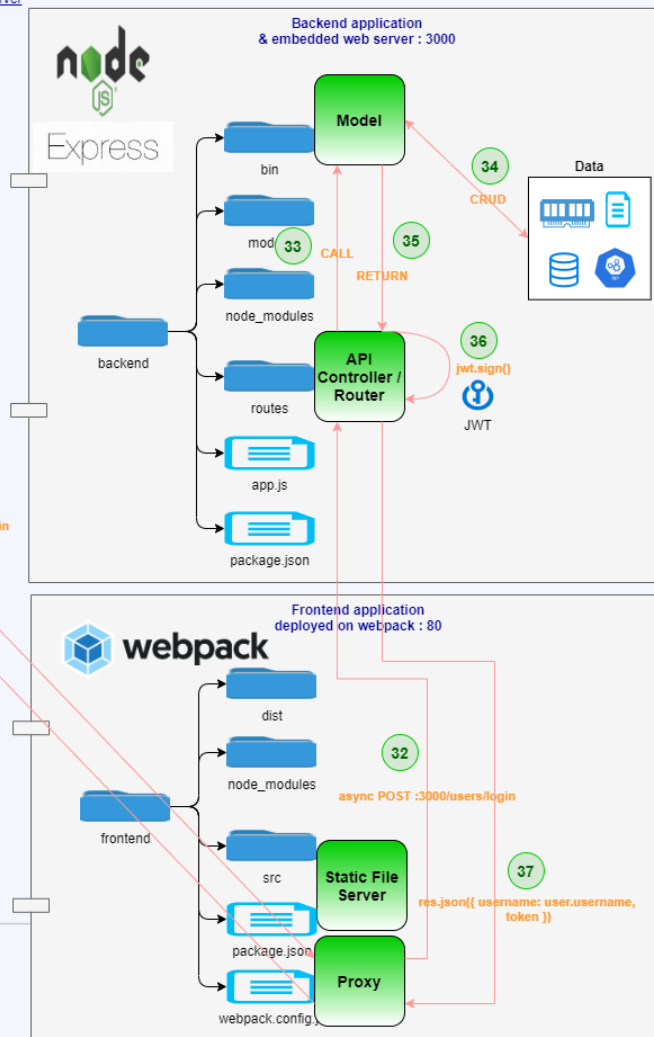
Utilisation de Webpack plus avancée

- Tout savoir sur webpack [\[98.\]](#)
- Installation de packages : `npm i webpack-dev-server css-loader style-loader file-loader babel-loader @babel/core @babel/preset-env -D`
- Configuration pour : Assets Management (CSS, images...)
- Configuration pour le proxy du devServer

Utilisation de webpack lors du développement



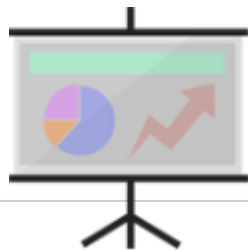
Server





Utilisation de Webpack plus avancée

- Tout savoir sur webpack [\[98.\]](#)
- Installation de packages : `npm i webpack-dev-server css-loader style-loader file-loader babel-loader @babel/core @babel/preset-env html-webpack-plugin -D`
- Configuration pour : Assets Management (CSS, images...)
- Configuration pour le proxy du devServer



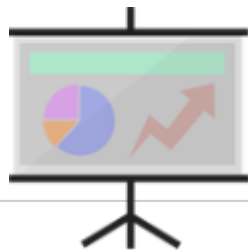
Utilisation de webpack

- **DEMO : Utilisation d'un package bundler (webpack)** pour le frontend de MyCMS
 - Suppression de tout ce qui touche à l'aspect serveur du frontend (**bin/www**, **app.js**...)
 - Réutilisation du contenu de **/public**
 - Structurer les dossiers : **/dist**, **/src**....
 - Modification de **package.json** : démarrage de Webpack (et plus **node ./bin/www**)
 - Installation des packages pour webpack
 - Configuration de départ pour webpack



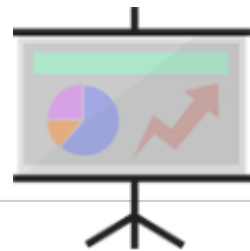
Utilisation de webpack

- ◎ **DEMO : Utilisation d'un package bundler (webpack)**
 - Gestion du devServer :
 - proxy sous Webpack
 - Servir **index.html** quand route n'amène à aucun fichier
 - Rendre le serveur accessible de l'extérieur
 - Ouvrir le browser par défaut à l'URL « localhost »



Utilisation de webpack

- ◎ DEMO : Utilisation d'un package bundler (webpack)
 - Gestion du fichier **index.html** :
 - Création automatique d'un fichier **index.html** dans **/dist** sur base du template **/src/index.html**
 - **/dist** enlevé de git



Utilisation de webpack

● DEMO : Utilisation d'un package bundler (webpack)

- Gestion du CSS (utilisation du style loader et du css loader)
- Utilisation d'un package pour **bootstrap** :
 - Bootstrap et dépendances : **npm i bootstrap jquery popper.js -D**
 - **index.js** :



```
import 'bootstrap/dist/css/bootstrap.css';
```