

Programmation web sur client

TD5 - Développement web moderne

Matthieu Nicolas

Exercice 1 - Bundle

Dans ce 1er exercice, nous allons nous contenter de *bundler* les différents fichiers JS de notre application à l'aide de Rollup.

Dans le dossier de votre application,

1. Installer Rollup à l'aide de la commande `npm install --save-dev rollup`. Les outils installés via `npm` sont placés dans le répertoire `/node_modules/.bin/` de votre projet. Pour les exécuter, vous pouvez soit taper la commande `./node_modules/.bin/<nom de l'outil>`, soit utiliser le raccourci `npx <nom de l'outil>`.
2. Utiliser Rollup en ligne de commande pour générer au format UMD le bundle `public/bundle.js` à partir du fichier d'entrée `public/main.js`.
3. Afin que cette démarche soit plus facilement reproductible, écrire le fichier de configuration `rollup.config.js` correspondant.
4. Pour simplifier le processus de build de l'application, créer la tâche `build` dans le fichier `package.json` qui correspondra à exécuter Rollup en utilisant la configuration indiquée par le fichier `rollup.config.js`. Cette tâche est ensuite utilisable via la commande `npm run build`.
5. Il faut maintenant exploiter le bundle nouvellement créé. Modifier votre fichier `index.html` pour utiliser seulement le fichier `public/bundle.js`.

Exercice 2 - Transpilation

Nous allons maintenant ajouter une nouvelle étape à notre processus de build : la *transpilation* de notre code en une version compatible avec les anciens navigateurs à l'aide de Babel.

Dans le dossier de votre application,

1. Faites une copie du fichier `public/bundle.js`.
2. Installer Babel et le plugin de Rollup qui lui est dédié grâce à la commande suivante `npm install --save-dev @babel-core rollup-plugin-babel`.
3. Modifier la configuration de Rollup pour faire appel à Babel au cours de la génération du bundle.

4. Re-exécuter Rollup pour régénérer le bundle. Comparer le fichier nouvellement créé avec la copie effectuée précédemment. Que constatez-vous ?
5. Par défaut, Babel est configuré pour ne rien faire. Pour transpiler notre code, il est nécessaire de modifier cette configuration. Installer une configuration pré-faite de Babel via la commande `npm install --save-dev @babel/preset-env`.
6. Créer le fichier `.babelrc` et configurer Babel pour utiliser la configuration précédemment installée.
7. Re-exécuter Rollup pour régénérer le bundle.

Exercice 3 - Minification

Pour finir, nous allons rajouter terser pour minifier notre code pour réduire la taille du bundle.

1. Installer le plugin de terser pour Rollup via la commande suivante:
`npm install --save-dev rollup-plugin-terser`.
2. Modifier la configuration de Rollup pour faire appel à terser au cours de la génération du bundle.
3. Re-exécuter Rollup pour régénérer le bundle.