

Développement web moderne

Matthieu Nicolas
Licence Pro CIASIE

Objectifs

- Vous faire découvrir plusieurs problématiques liés au développement web moderne
- Vous présenter des outils existants pour y répondre

Plan

- Transpilation
- Minification
- Bundle

Transpilation

Développement web moderne

C'est quoi ?

- Un cas particulier de la compilation
- La compilation consiste à transformer du code **source** en code **machine**
- La transpilation consiste à transformer du code **source** en un autre code **source**

Pourquoi faire?

- Jusqu'ici, on a développé en profitant des dernières fonctionnalités de JS
 - ESM Modules, `let` et `const`, `class`, `Promises`...
- Permet d'écrire du code plus simple...
- ... mais pas compatible avec toutes les versions des navigateurs

Exemple

```
function verifyUser(username, password, callback) {
  dataBase.verifyUser(username, password, function(error, userInfo) {
    if (error) {
      callback(error);
    } else {
      dataBase.getRoles(username, function(error, roles) {
        if (error) {
          callback(error);
        } else {
          dataBase.logAccess(username, function(error) {
            if (error) {
              callback(error);
            } else {
              callback(null, userInfo, roles);
            }
          });
        }
      });
    }
  });
}
```

Older JavaScript:

- ✓ Runs on all browsers
- ✗ Hard for developers to read & write
- ✗ Bug-prone

```
async function verifyUser(username, password) {
  try {
    const userInfo = await dataBase.verifyUser(username, password);
    const rolesInfo = await dataBase.getRoles(userInfo);
    const logStatus = await dataBase.logAccess(userInfo);
    return userInfo;
  } catch (e) {
    //handle errors as needed
  }
}
```

Latest JavaScript:

- ✓ Easy for developers to read & write
- ✓ Easier to test and understand (fewer bugs)
- ✗ Doesn't run on all browsers

<https://www.vrk.dev/2019/07/11/why-is-modern-web-development-so-complicated-a-long-yet-hasty-explanation-part-1/>

Délai d'adoption

- Existe obligatoirement un délai entre
 - Le moment où une nouvelle fonctionnalité est adoptée dans le standard ECMAScript
 - Le moment où cette fonctionnalité est supportée par les navigateurs
 - Le moment où les utilisateurs ont mis à jour leurs navigateurs

But de la transpilation

- La transpilation va nous permettre de prendre du code JS moderne...
- ... et de le convertir en code JS de l'ancien temps

Babel

Developer codes in
easy-to-use, robust
modern JavaScript

```
async function verifyUser(username, password) {  
  try {  
    const userInfo = await dataBase.verifyUser(username, password);  
    const rolesInfo = await dataBase.getRoles(userInfo);  
    const logStatus = await dataBase.logAccess(userInfo);  
    return userInfo;  
  } catch (e) {  
    //handle errors as needed  
  }  
}
```



Babel compiles it to older JavaScript
that works on all browsers.



```
function verifyUser(username, password, callback) {  
  dataBase.verifyUser(username, password, function(error, userInfo) {  
    if (error) {  
      callback(error);  
    } else {  
      dataBase.getRoles(username, function(error, roles) {  
        if (error) {  
          callback(error);  
        } else {  
          dataBase.logAccess(username, function(error) {  
            if (error) {  
              callback(error);  
            } else {  
              callback(null, userInfo, roles);  
            }  
          });  
        }  
      });  
    }  
  });  
}
```



Latest JavaScript + Babel:

- ✓ Easy for developers to read & write
- ✓ Easier to test and understand (fewer bugs)
- ✓ Runs on all browsers

Install & config

- S'installe via npm
 - `npm install --save-dev @babel/core`
- Utilise un fichier de configuration: `.babelrc`
- Mais peut utiliser des configurations pré-faites:
 - `npm install --save-dev @babel/preset-env`

```
{  
  | ... "presets": [ "@babel/preset-env" ]  
}
```

Infos supplémentaires

- Dans notre cas, on ne va pas utiliser Babel directement
- Va plutôt l'utiliser par l'intermédiaire d'un autre outil, le bundler
- <https://babeljs.io/>
- <https://github.com/babel/babel>

Minification

Développement web moderne

Performances - 1

- Applications de plus en plus conséquentes
 - + en + de code
- Cela impacte négativement les performances
 - Et donc l'expérience utilisateur

Trop de code...

- Le code est écrit pour être lu
 - Plus précisément, lu par des humains
- Cela se répercute sur son contenu
 - Noms de variable explicites
 - Code aéré, indenté

Bande passante

- L'ordinateur n'a pas besoin de tout ces octets en trop...

```
1  class Media {  
2  
3  ... constructor(title, releaseDate, rating, img) {  
4      ... this.title = title  
5      ... this.releaseDate = releaseDate  
6      ... this.rating = rating  
7      ... this.img = img  
8  ... }  
9  }  
10
```

- ... et nous, on a pas tous la fibre

Minification

- Peut supprimer les caractères inutiles, raccourcir les noms de variables...
- On appelle ce processus minification du code (30-40% de gain)

```
1 class e{constructor(t,i,n,e){this.title=t,this.releaseDate=i,this.rating=n,this.img=e}}
```

terser

- Outil permettant de minimiser son code JS
- Successeur de uglify
 - Supporte la syntaxe ES6+
- <https://terser.org/>
- <https://github.com/terser/terser>

Install et utilisation

- Comme Babel, s'installe via npm...
- ... et comme Babel, on va l'utiliser par l'intermédiaire du bundler

Bundle

Développement web moderne

Performances - 2

- Applications de plus en plus conséquentes
 - + en + de fichiers
- Cela impacte négativement les performances
 - Et donc l'expérience utilisateur

Trop de fichiers...

- Une requête HTTP est nécessaire pour chaque fichier


Status	Method	File	Domain	Cause	Type	Transferred	Size	0 ms160 ms320 ms480 ms							
200	GET	index.js	localhost:8080	script	js	2.08 KB	1.78 KB				10 ms				
200	GET	Album.js	localhost:8080	script	js	564 B	260 B				4 ms				
200	GET	Game.js	localhost:8080	script	js	594 B	290 B				4 ms				
200	GET	Movie.js	localhost:8080	script	js	634 B	330 B				4 ms				
200	GET	App.js	localhost:8080	script	js	8.45 KB	8.15 KB				4 ms				
200	GET	css?family=Lato:400,700,400itali...	fonts.googleapis.com	stylesheet	css	948 B	3.02 KB				18 ms				
200	GET	Media.js	localhost:8080	script	js	503 B	199 B							58 ms	
200	GET	Collection.js	localhost:8080	script	js	722 B	418 B							58 ms	

Latence réseau

- Tout le monde n'a pas le serveur qui tourne en local sur sa machine
- Chaque requête implique une latence réseau
 - 30-50ms pour les requêtes continentales
 - 90ms pour chaque requête transatlantique
- Ces latences réseaux s'additionnent

Bundle

- Pour limiter ce problème, on va réduire le nombre de fichiers
- On va regrouper tous les fichiers ensemble dans un bundle

Status	Method	File	Domain	Cause	Type	Transferred	Size	0 ms	160 ms	320 ms	480 ms	640 ms	800 ms ▲
200	GET	 jquery-3.3.1.js	 localhost:8080	script	js	265.68 KB	265.38 KB		 11 ms				
200	GET	 semantic.min.js	 localhost:8080	script	js	269.57 KB	269.27 KB		 10 ms				
200	GET	 bundle.js	 localhost:8080	script	js	8.26 KB	7.97 KB		 3 ms				
200	GET	 css?family=Lato:400,700,400itali...	 fonts.googleapis.com	stylesheet	css	948 B	3.02 KB			 164 ms			
200	GET	 S6u9w4BMUTPHh6UVSwiPGQ3...	 fonts.gstatic.com	font	woff2	14.21 KB	13.75 KB				 175 ms		
200	GET	 S6uyw4BMUTPHjx4wXiWtFCc.w...	 fonts.gstatic.com	font	woff2	14.08 KB	13.62 KB				 208 ms		

HTTP/2

- La nouvelle version du protocole HTTP, HTTP/2, a notamment pour but de répondre à ce problème
- Permet d'effectuer plusieurs requêtes sur une même connexion TCP
- Le jour où HTTP/2 sera répandu, moins d'intérêt à bundler

Rollup

- Un des nombreux bundlers existants
 - Voir webpack, microbundle...
- S'installe via npm
 - `npm install --save-dev rollup`
- <https://rollupjs.org/guide/en/>
- <https://github.com/rollup/rollup>

Utilisation

- Outil en ligne de commande
 - `npx rollup -f umd -o public/bundle.js public/main.js`
- Où
 - `npx` permet d'exécuter un outil installé via `npm` en local
 - `-f` permet d'indiquer le format du bundle (`iife`, `cjs`, `umd...`)
 - `-o` permet d'indiquer le fichier de sortie
 - et qui prend en argument le fichier d'entrée

Configuration

- Possible de passer par un fichier de configuration: `rollup.config.js`

```
export default ({
  ...input: "public/main.js",
  ...output: {
    ...file: "public/bundle.js",
    ...format: "umd"
  }
})
```

- Dans ce cas, utiliser la commande:
 - `npx rollup -c`

Fonctionnalités supplémentaires

- Rollup permet de regrouper l'ensemble des fichiers JS de notre application en 1 seul...
- ... mais pas que !
 - Tree Shaking
 - Plugins

Tree shaking

- En étudiant les import/export de notre application, capable de déterminer le code mort
 - Par exemple une fonction exportée qui n'est jamais importée
- Rollup va supprimer ce code inutile du bundle final

Plugins - 1

- Comme dit précédemment, on va utiliser les outils de transpilation et de minification via Rollup
- L'intégration se fait par le biais de plugins

Plugins - 2

- Babel
 - `npm install --save-dev rollup-plugin-babel@latest`
- terser
 - `npm install --save-dev rollup-plugin-terser`

MàJ de la config

```
import babel from "rollup-plugin-babel"
import { terser } from "rollup-plugin-terser"

export default ({
  ...input: "public/main.js",
  ...output: {
    ...file: "public/bundle.js",
    ...format: "umd"
  },
  ...plugins: [
    ...terser(),
    ...babel({
      ...exclude: "node_modules/**"
    })
  ]
})
```

Conclusion

- Il est nécessaire d'adopter des bonnes pratiques de packaging pour les applis web
- De nombreux outils existent, choisissez-en un
- Optimiser votre algorithme, c'est bien...
- ... mais optimiser votre build, c'est aussi important

Pour aller plus loin

- Web Performance 101
 - <https://3perf.com/talks/web-perf-101/>
- Google PageSpeed Insights
 - <https://developers.google.com/speed/pagespeed/insights/>

Ouverture

- State of JS 2018 (y a pas encore 2019)
 - <https://2018.stateofjs.com/>

TD

Développement web moderne

TD - 1

- On reprend l'application de chat
- On va utiliser Rollup pour packager correctement l'application

TD - 2

- `git add remote classroom`
`https://github.com/MatthieuNICOLAS/2019-lp-ciasie-prog-web-td5.git`
- `git pull classroom master`