

ES6 Modules

Matthieu Nicolas
Licence Pro CIASIE

But des modules

*Good authors divide their books into chapters and sections;
good programmers divide their programs into modules.*

- Décomposer notre application
 - Maintenabilité
 - Isolation (**namespace**)
 - Réutilisation

Historique

Design pattern

Module - 1

```
1  MyMediatek.models.Media = (function() {  
2    ...class Media {  
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   ...return Media  
11 }())  
12 |
```

- Permet de définir des variables, des objets ou des fonctions dans un scope isolé
- Permet de rendre accessible uniquement ce que nous souhaitons via un objet global

Design pattern

Module - 2

```
1  MyMediatek.models.Game = (function() {  
2  ...class Game extends MyMediatek.models.Media {  
3  ...constructor(title, releaseDate, rating, img, studio, nbPlayers, plot) {  
4  ...super(title, releaseDate, rating, img)  
5  ...this.studio = studio  
6  ...this.nbPlayers = nbPlayers  
7  ...this.plot = plot  
8  ...}  
9  ...}  
10 ...return Game  
11 })()  
12
```

- Peut être réutilisé par un ou plusieurs autres modules via l'objet global

Design pattern

Module - 3

```
103 .....<script type="text/javascript">
104 .....    MyMediatek = {
105 .....        models: {}
106 .....    }
107
108 .....    $(document).ready(function() {
109 .....        const game = new MyMediatek.models.Game( ... )
110 .....    })
111 .....</script>
112 .....<script type="text/javascript" src="src/models/Media.js"></script>
113 .....<script type="text/javascript" src="src/models/Game.js"></script>
114
```

- Doit faire attention à l'ordre de chargement des modules !

CommonJS

- Apparition de **Node.js** en 2009
- Propose un système de modules
 - Permet notamment de faire apparaître les dépendances entre modules

Exemple

```
1  let counter = 1
2
3  function increment() {
4    counter++
5  }
6
7  function decrement() {
8    counter--
9  }
10
11 function value() {
12   return counter
13 }
14
15 module.exports = {
16   counter: counter,
17   increment: increment,
18   decrement: decrement,
19   value: value
20 }
21
```

models/counter.js

```
1  const counter = require('./models/counter')
2
3  counter.increment()
4  console.log(counter.value()) // 2
5
6
7
8  console.log(counter.counter) // WARNING : 1
9
```

main.js

Du côté du navigateur...

- **CommonJS** n'est pas compatible avec les navigateurs...
- ... mais la communauté se rend compte qu'un vrai système de module, c'est cool !
- **Asynchronous Module Definition (AMD)** s'impose dans la communauté JS

Compatibilité navigateur - Node.js

- À la base, **Node.js** devait permettre de partager le code entre le client et le serveur
- Mais comme les systèmes de modules utilisés sont différents, cela échoue...
- Apparition de **Universal Module Definition (UMD)**
 - Permet de définir des modules à la fois compatibles avec le navigateur et **Node.js**

ES6 Modules

- **CommonJS, AMD et UMD** ne sont pas standards...
- ... mais avec ES6, JavaScript dispose enfin d'un système de modules !
- **Avantages**
 - Chargement asynchrone
 - Gestion des dépendances circulaires

Exemple - 1

main.js

```
1 import { increment, value } from "../models/counter.js"
2 |....// N'oubliez pas l'extension du fichier
3 |
4 increment()
5 console.log(value()) // 2
6
```

```
1 export let counter = 1
2
3 export function increment() {
4 |....counter++
5 |}
6
7 export function decrement() {
8 |....counter--
9 |}
10
11 export function value() {
12 |....return counter
13 |}
14
```

models/counter.js

```
1 <html>
2 |....<head>
3 |....|....<meta charset="utf-8">
4 |....|....<title>Exemple</title>
5 |....</head>
6 |....<body>
7 |....|....<!-- include src files here ... -->
8 |....|....<script type="module" src="main.js"></script>
9 |....</body>
10 </html>
11
```

index.html

Exemple - 2

```
1 export let counter = 1
2
3 function increment() {
4   ...counter++
5 }
6
7 function value() {
8   ...return counter
9 }
10
11 export { increment as inc }
12 export { value as val }
13
```

models/counter.js

```
1 import {
2   ...inc as incCounter,
3   ...val as valCounter
4 } from './models/counter.js'
5 // N'oubliez pas l'extension du fichier
6
7 incCounter()
8 console.log(valCounter()) // 2
9
```

main.js

- Possible de changer le nom lors de l'export ou de l'import

Exemple - 3

```
1 export let counter = 1
2
3 export function increment() {
4   ...counter++
5 }
6
7 export function decrement() {
8   ...counter--
9 }
10
11 export function value() {
12   ...return counter
13 }
14
```

models/counter.js

```
1 import * as counter from "../models/counter.js"
2   ...// N'oubliez pas l'extension du fichier
3
4 counter.increment()
5 console.log(counter.value()) // 2
6
```

main.js

- Possible de regrouper tous les imports sous un **namespace**

Singletons

- **Les ES6 Modules sont des Singletons**
 - Ils ne sont chargés qu'une fois
 - Les variables sont partagés entre tous les modules *clients*

Références

```
1 export let counter = 1
2
3 export function increment() {
4   ...counter++
5 }
6
7 export function decrement() {
8   ...counter--
9 }
10
11 export function value() {
12   ...return counter
13 }
14
```

models/counter.js

```
1 import { counter, increment, value } from "../models/counter.js"
2   ...// N'oubliez pas l'extension du fichier
3
4 increment()
5 console.log(value()) // 2
6
7
8 console.log(counter) // WARNING :: 2
9 counter = 3 // WARNING :: Type Error
10
```

main.js

- Les exports sont passés par référence
 - Contrairement à **CommonJS**
- Par contre, peut pas modifier leur valeur

npm

♥ Nemo's Parental Misguidance

[npm Enterprise](#) [Features](#) [Pricing](#) [Docs](#) [Support](#)



Search packages

Search



npm is the package manager for javascript

Popular libraries

lodash
request
chalk
react
express
commander
moment
async
debug
bluebird
prop-types
react-dom

Discover packages

IoT
mobile
front end
backend
robotics
css
testing
cli
documentation
math
coverage
frameworks

By the numbers

Packages

840,636

Downloads · Last Week

8,086,509,461

Downloads · Last Month

23,343,280,108

Recently updated packages

rn_xxf

Rn_xxf

xxf_xuanyouwu published 1.3.3 • 3 minutes ago

vita.admin.plugin

None

maraboujr published 0.1.0 • 3 minutes ago

npm

- **Package manager** pour JS
- Initialement lié à **Node.JS**
- Dorénavant utilisé pour les modules côté navigateur et serveur
- Quelques chiffres
 - 840 636 paquets
 - 23+ milliards de téléchargement ces 30 derniers jours

npm

- Outil en ligne de commande
- Permet de
 - Gérer notre propre package
 - Gérer les dépendances de notre projet
 - Installer des programmes

npm init

- Permet de générer le fichier *package.json*
- Ce fichier décrit notre projet
 - nom
 - numéro de version
 - description
- Mais aussi liste les dépendances
 - Pour l'exécution (bibliothèques utilisées)
 - Pour le développement (compilateur, linter, build...)

Exemple

```
1  {  
2    "name": "my-mediatek",  
3    "version": "0.0.1",  
4    "description": "A simple mediatek",  
5    "author": "Matthieu Nicolas",  
6    "license": "ISC"  
7  }  
8
```

Un simple *package.json*

scripts

- Le champ *scripts* permet de définir des tâches à exécuter
 - *npm run <nom du script>*

```
1  {
2    · "name": "my-mediatek",
3    · "version": "0.0.1",
4    · "description": "A simple mediatek",
5    · "scripts": {
6      · "toto": "echo \"TOTO !\" && exit 0"
7    },
8    · "author": "Matthieu Nicolas",
9    · "license": "ISC"
10 }
11
```

- Les scripts *start* et *test* sont supportés par défaut
 - Peut exécuter *npm start* ou *npm test* directement

npm install

- npm install permet de spécifier et d'installer les dépendances de notre projet
 - *npm install <nom de la dépendance>*
- ATTENTION : il faut ajouter des options pour que la dépendance soit ajoutée au package.json
 - *npm install -- save <nom de la dépendance>*
 - *npm install -- save-dev <nom de la dépendance>*

Exemple

```
52  .. "dependencies": {  
53    .. "mute-structs": "^0.4.2",  
54    .. "rxjs": "^6.3.3",  
55    .. "safe-any": "^1.0.2"  
56  },  
57  .. "devDependencies": {  
58    .. "@commitlint/cli": "^7.1.2",  
59    .. "@commitlint/config-conventional": "^7.1.2",  
60    .. "@types/node": "^8.10.29",  
61    .. "assert": "^1.4.1",  
62    .. "ava": "^1.0.0-beta.8",  
63    .. "commitizen": "^2.10.1",  
64    .. "cz-conventional-changelog": "^2.1.0",  
65    .. "husky": "^0.14.3",  
66    .. "lint-staged": "^7.2.2",  
67    .. "markdownlint-cli": "^0.13.0",  
68    .. "prettier": "^1.14.2",  
69    .. "protobufjs": "^6.8.8",  
70    .. "rollup": "^0.65.2",  
71    .. "rollup-plugin-cleaner": "^3.0.0",  
72    .. "rollup-plugin-commonjs": "^9.1.6",  
73    .. "rollup-plugin-filesize": "^4.0.1",  
74    .. "rollup-plugin-node-resolve": "^3.4.0",  
75    .. "rollup-plugin-re": "^1.0.7",  
76    .. "rollup-plugin-typescript2": "^0.17.0",  
77    .. "semantic-release": "^15.9.14",  
78    .. "travis-deploy-once": "^5.0.7",  
79    .. "tslint": "^5.11.0",  
80    .. "tslint-config-prettier": "^1.15.0",  
81    .. "typescript": "^3.0.3"  
82  },
```

Un simple *package.json*

node_modules

- Les dépendances sont installées dans le répertoire *node_modules*
- Pas besoin de les *push* sur **git**
- À partir du *package.json*, un collaborateur peut récupérer l'ensemble des dépendances à l'aide de la commande *npm install*

Conclusion - 1

- Enfin un système de module en JS...
- ... mais on risque de mettre du temps à enterrer **CommonJS**, **AMD** et **UMD**

Conclusion - 2

- Faire ses modules avec **ES6 Modules**
- Des bundlers se chargent de transformer vers d'autres formats

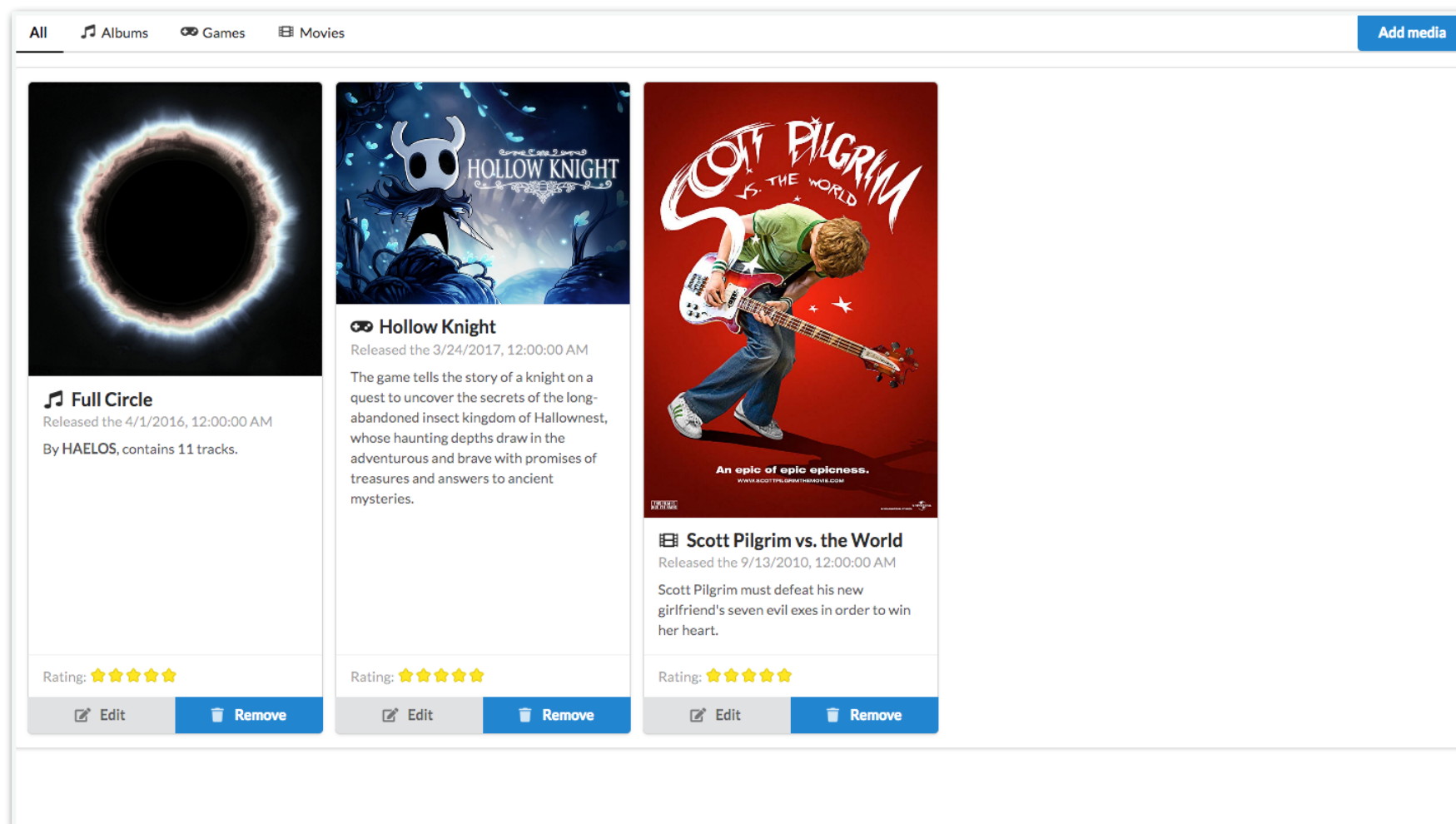
Quelques références

- ES6 & Beyond
 - <https://github.com/getify/You-Dont-Know-JS/tree/master/es6%20%26%20beyond>
- JavaScript Modules: A Beginner's Guide
 - <https://medium.freecodecamp.org/javascript-modules-a-beginner-s-guide-783f7d7a5fcc>
- Document *package.json*
 - <https://docs.npmjs.com/files/package.json>

TP

Consignes

- On reprend l'application de médiathèque du dernier TP



- Et on modifie le code pour dorénavant utiliser des ES6 Modules

Let's go

<https://classroom.github.com/a/CubhA-vy>
(lien dispo sur Arche)