

Développement front

Danielo **JEAN-LOUIS**
Michele **LINARDI**

MMI 2 – TP#5 S4

nodejs

- **Outil gratuit et open source** permettant de faire du javascript côté serveur
- Apparu en 2009
- Plus performant et rapide comparé à un serveur Apache (notamment utilisé pour PHP)

Source(s) :

- <https://nodejs.org/en/>

nodejs

- Dernière version (2022) : version 18.X.X
- A permis d'améliorer l'environnement de travail des développeurs front-end
 - Compilation / optimisation de fichiers...

Source(s) :

- <https://nodejs.org/en/>

nodejs

- Certaines fonctions javascript pour navigateur ne sont pas compatibles avec le serveur
 - ex : `alert()` ou `document.querySelector()`
- Fonctionne via un système de modules
 - Appelés également packages

Source(s) :

- <https://nodejs.org/en/>

node-modules

- Ensemble de modules permettant de rajouter des fonctionnalités
 - Côté serveur ou client
- Installation grâce à npm
 - node package manager
 - **Installation en ligne de commandes**

Source(s) :

- <https://nodejs.org/en/>

node_modules

- **Ne jamais commiter les node_modules**
 - C'est beaucoup trop lourd, chaque développeur doit les installer à chaque fois qu'il récupère un projet
 - Toujours rajouter “node_modules” dans le .gitignore
- npmjs.org – Contient la liste des modules publics accessibles

Source(s) :

- <https://nodejs.org/en/>
- <https://www.npmjs.com/>

.gitignore

- Fichier racine à un projet .git
- Permet à git d'ignorer fichiers et dossiers
- Penser à toujours ignorer les dépendances... :
 - node_modules (js), vendor (php)...
- ...et fichiers de logs
 - Ils n'ont rien à faire sur git

Source(s) :

- <https://www.atlassian.com/git/tutorials/saving-changes/gitignore>
- https://www.w3schools.com/git/git_ignore.asp


npm

- Installé en même temps que nodejs
- Permet d'installer des modules externes
- Donne à un ensemble de commandes :
 - Initialiser un projet avec npm
 - Installer un module
 - ...

Source(s) :

- <https://nodejs.org/en/>
- <https://www.npmjs.com/>

npm – Initialiser un projet



```
// A faire impérativement dans un dossier  
npm init -y
```

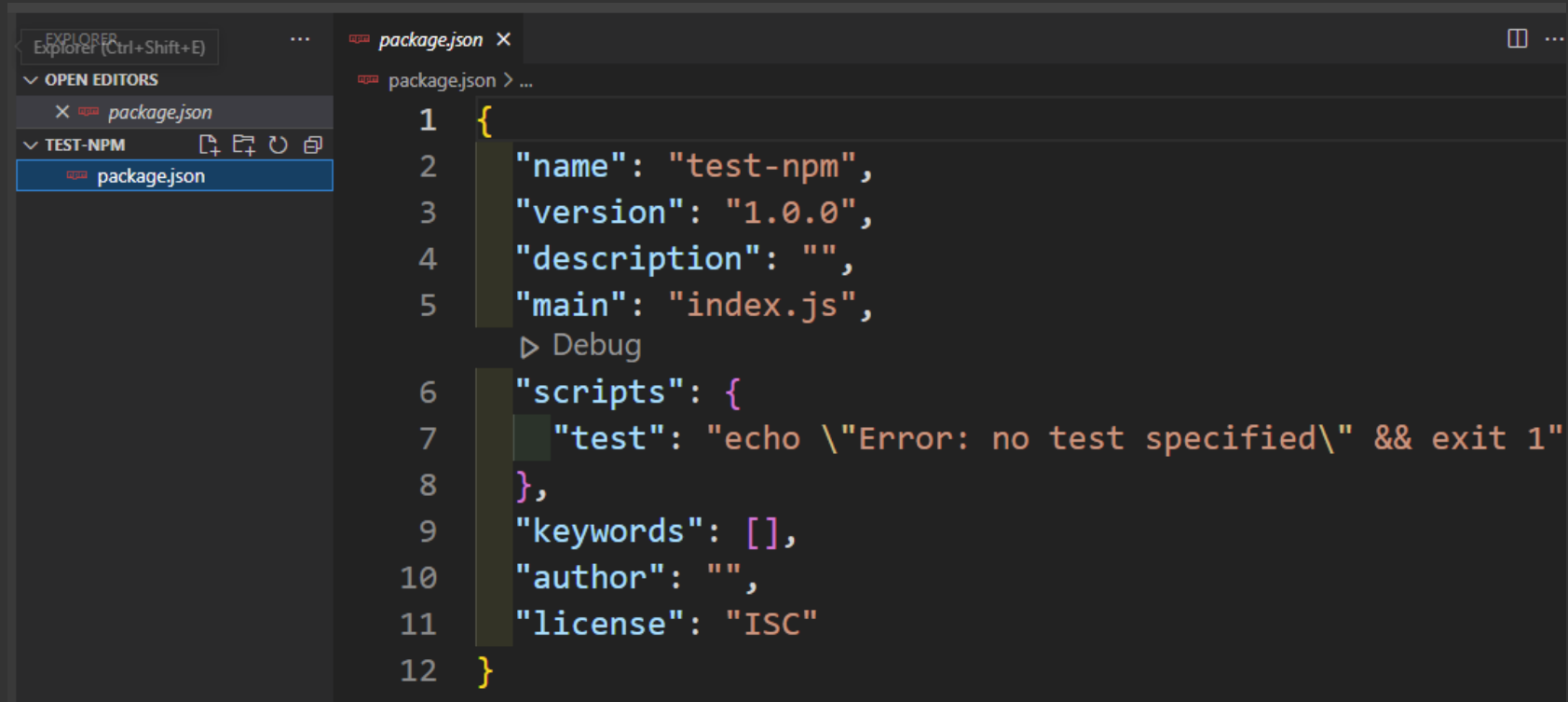
npm init -y va créer le nécessaire pour démarrer un projet avec npm.
Le paramètre "-y" permet d'exécuter "npm init" avec les paramètres par défaut.

La commande doit être effectuée dans votre dossier de projet. Pas n'importe où !

Source(s) :

- <https://fr.reactjs.org/docs/components-and-props.html>

npm – Initialiser un projet



The image shows a screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar is open, showing a file named 'package.json' under a folder named 'TEST-NPM'. The main editor area displays the content of 'package.json' with line numbers 1 through 12. The file contains a JSON object with the following properties: 'name' (test-npm), 'version' (1.0.0), 'description' (empty string), 'main' (index.js), 'scripts' (a test script), 'keywords' (empty array), 'author' (empty string), and 'license' (ISC). A 'Debug' icon is visible below the 'main' property.

```
1  {
2    "name": "test-npm",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC"
12 }
```

Source(s) :

- <https://fr.reactjs.org/docs/components-and-props.html>

npm – Initialiser un projet – package.json

```
5   "main": "index.js",
```

- Point d'entrée de l'application
 - Exemple `root.render()` pour React
- Un seul point d'entrée par projet

Source(s) :

- <https://fr.reactjs.org/docs/components-and-props.html>

npm – Initialiser un projet – package.json

```
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1",  
8   },  
  },  
}
```


- Liste des commandes du projet
- Accessible, via ligne de commande :
 - *npm run nom-de-la-commande*
 - Exemple ici : « npm run test »

npm – Initialiser un projet – package.json

```
"start": "node index.js"
```

Avec la ligne suivante (dans mon package.json), à l'appel de la commande "npm run start", je vais demander à mon serveur node d'exécuter le fichier "index.js"

npm – Installer un projet

A terminal window with a dark purple background and a light purple border. In the top-left corner, there are three small, light purple circles representing window control buttons. The text 'npm install react' is displayed in a light green monospace font.

```
npm install react
```

Ici, on installe react. Le contenu du téléchargement sera dans un dossier `node_modules/` (**dossier à ne jamais commiter**)

Source(s) :

- <https://fr.reactjs.org/docs/components-and-props.html>

npm – Installer un projet

```
import react from "react";  
// ou  
const react = require('react');
```

Dans un fichier javascript, on peut importer "react" et l'utiliser.
Les deux syntaxes sont valides, mais on utilisera plutôt la première, plus moderne.
La suite dépendra de l'utilisation de votre module.

Source(s) :

- <https://nodejs.org/en/>
- <https://www.npmjs.com/>

Importation de modules

- Contrevient aux limitations de la balise `<script>`
 - Le dev n'est plus tributaire de l'ordre des balises
- Mot-clé "import" pour importer et "export" pour exporter un module
- Permet d'accéder à un module depuis n'importe où dans le projet

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

Pratiquons ! - Découvrons nodejs (Partie 1/2)

Pré-requis :

- Avoir la ressource `ressources/nodejs`

A télécharger ici :

<https://download-directory.github.io?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s4/travaux-pratiques/numero-5/ressources>

node_modules – Suite

```
"dependencies": {  
  "underscore": "^1.13.4"  
}
```

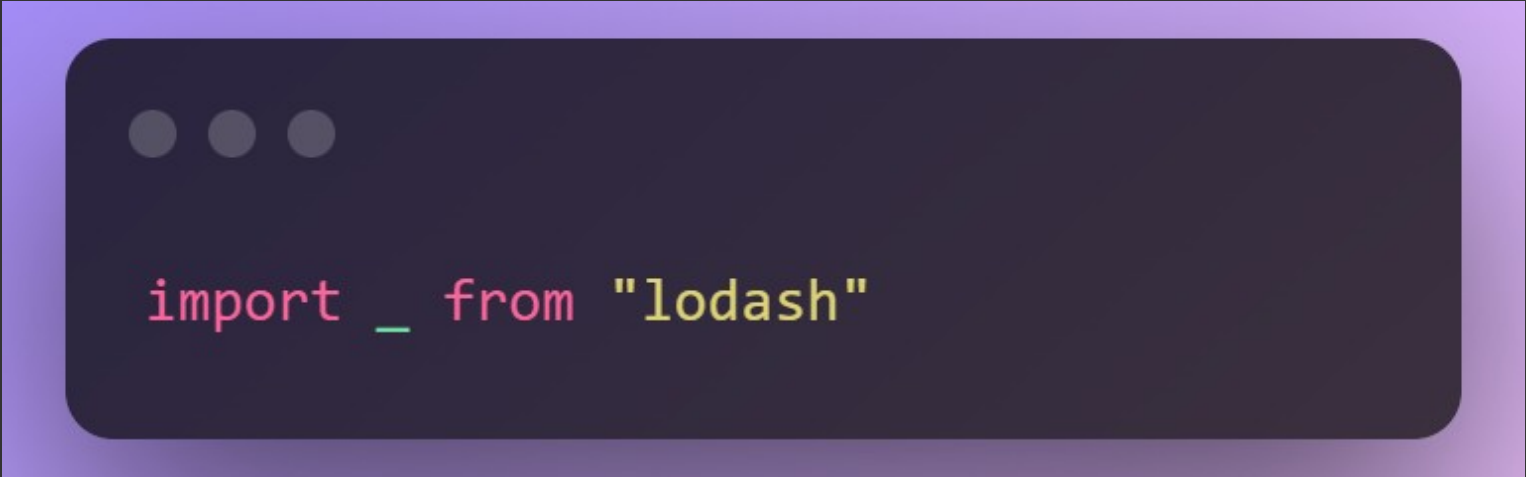
- Les dépendances sont listées dans le fichier `package.json`
 - Possibilité d'installer toutes les dépendances listées grâce à la commande `"npm install"`

Source(s) :

- <https://fr.reactjs.org/docs/components-and-props.html>

lodashjs

- Bibliothèque javascript proposant différents utilitaires pour manipuler des éléments

A code editor window with a dark purple background and a light purple border. It features three small circular window control buttons in the top-left corner. The text 'import _ from "lodash"' is displayed in a monospaced font, with 'import' in pink, '_' in green, 'from' in pink, and '"lodash"' in yellow.

```
import _ from "lodash"
```

Toutes les fonctions de la bibliothèque lodash
sont stockées dans "_"

Source(s) :

- <https://lodash.com/>

lodashjs

```
import _ from "lodash";  
  
console.log(_.flattenDeep([1, [2, [3, [4]], 5]]));
```

Nous utilisons ensuite une des fonctions de
lodash (via « _ ») dans notre code

Source(s) :

- <https://lodash.com/>

Pratiquons ! - Découvrons nodejs (Partie 3)

Pré-requis :

- Avoir la ressource `ressources/nodejs`

A télécharger ici :

<https://download-directory.github.io?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s4/travaux-pratiques/numero-5/ressources>

Importation de modules – Suite

- Possibilité de créer ses propres modules
 - Il faudra définir le **chemin relatif** entre les deux fichiers
- Un fichier peut importer plusieurs modules en même temps
 - De même pour l'exportation

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

Importation de modules – Suite

```
const myFunction = () => {  
  return "Bonjour tout le monde"  
}  
  
export default myFunction;
```

Export de la fonction "myFunction"

```
import myFunction from "./myFunction.js"
```

Import de la fonction "myFunction"

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

Importation de modules – Suite

```
export default myFunction;
```

- Indique ce qui sera chargé par défaut quand on importera le fichier (ici une fonction)
- **Il ne peut avoir qu'un seul "export default" par fichier, un seul**

Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

Pratiquons ! - Découvrons nodejs (Partie 4)

Pré-requis :

- Avoir la ressource ressources/nodejs

A télécharger ici :

<https://download-directory.github.io?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s4/travaux-pratiques/numero-5/ressources>

nodejs

- Outil permettant de faire du javascript dans la console du navigateur et côté serveur
- Permet le téléchargement de dépendances via npm (node package modules)

nodejs

- Résout le problème d'ordre des fichiers grâce à son système d'importation
- Facilite le développement via des outils tiers comme CRA (create react app)
 - CRA que nous allons découvrir au prochain TP

Questions ?

