

C/

CRÉER UNE PREMIÈRE APPLICATION EN NODE JS

P/3

INSTALLATION DE NPM (NODE
PACKAGE MANAGER) ET NODE JS

RÉSULTAT

P/5

CRÉATION D'UN PROJET NODE JS

RÉSULTAT

P/8

ON AMÉLIORE NOTRE
APPLICATION

RÉSULTAT

P/9 CRÉDITS

Node.js est une plateforme logicielle libre et événementielle en JavaScript. Elle permet de réaliser une application complète en Javascript que ce soit du côté du client ou du serveur.

Un bon nombre de plateformes web connues à fort trafic tournent sur Node JS, on peut citer LinkedIn, Yahoo !, Rakuten,...

De nombreux frameworks Javascript tournent sur Node JS, on peut citer notamment [Express](#) et [Meteor](#)

Avec Node JS, on peut construire des applications mobiles cross-platform (avec [Ionic](#)) ou des applications desktop cross-platform (avec [electron](#))

Les meilleurs exemples dans ce domaine sont [Discord](#) et [Slack](#), des outils de collaboration intégrant la discussion instantanée notamment utilisés par les gamers et les développeurs.



Au préalable

Pour acquérir cette compétence, vous avez besoin de connaître le HTML et CSS et d'être à l'aise avec la programmation en Javascript.



De l'aide

La documentation est essentielle pour bien comprendre le fonctionnement et le développement. Comme tout bon développeur, ayez le réflexe d'avoir la documentation de Node JS ou encore le site des communautés à disposition.

[Site officiel](#)

[Une documentation de développeur](#)

INSTALLATION DE NPM (NODE PACKAGE MANAGER) ET NODE JS

NPM (Node Package Manager)

NPM est un outil en ligne de commande permettant de gérer les dépendances en Javascript (Node JS ici).

Il est à télécharger et à installer. L'installateur Node JS installe également NPM : <https://nodejs.org/en/download/>

NPM permet de :

- Garder à jour les différentes bibliothèques dont un projet a besoin pour fonctionner.
- Gérer les dépendances entre bibliothèques (installation & mises à jour).
- Générer un fichier de chargement automatique commun à toutes les bibliothèques générées.

Centralisation des informations de chaque bibliothèque :

- Sur le site www.npmjs.com
- Accès à Internet requis (configurer pour le proxy).

Node JS

Pour installer la plateforme Node JS,

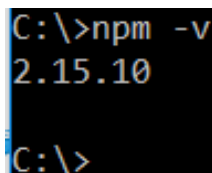
Elle est à télécharger et à installer. L'installateur Node JS installe également NPM : <https://nodejs.org/en/download/>

Résultat

Vérification des installations de NPM et Node JS

Pour vérifier que NPM et Node JS sont bien installés. En ligne de commande, on va exécuter :

npm -v



```
C:\>npm -v
2.15.10
C:\>
```

node -v

```
C:\>node -v  
v10.15.3  
  
C:\>
```

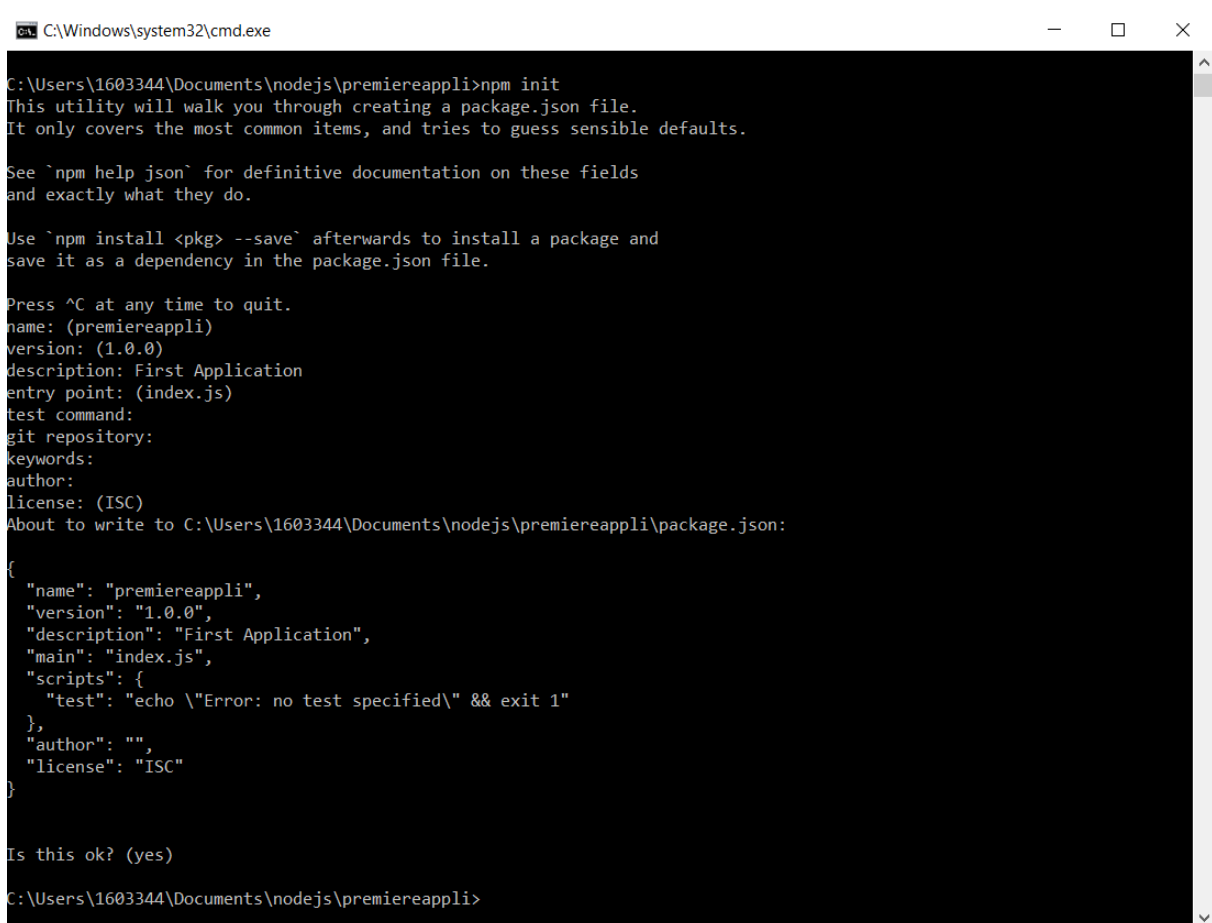
CRÉATION D'UN PROJET NODE JS

NPM initialisation

Nous allons créer un répertoire sur le local (peu importe l'emplacement) que l'on va nommer par exemple "PremiereAppli".

Nous allons ensuite exécuter la ligne de commande **npm init**

On répond aux questions qui nous sont posées (nom de l'application, description de l'application, auteur,...)



```
C:\Windows\system32\cmd.exe

C:\Users\1603344\Documents\nodejs\premiereappli>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (premiereappli)
version: (1.0.0)
description: First Application
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\1603344\Documents\nodejs\premiereappli\package.json:

{
  "name": "premiereappli",
  "version": "1.0.0",
  "description": "First Application",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this ok? (yes)
C:\Users\1603344\Documents\nodejs\premiereappli>
```

On va regarder dans le répertoire, un seul fichier a été créé (package.json).

Installation du framework Express

Dans le répertoire, en ligne de commande, nous allons maintenant installer le framework Express contenant les composants et les outils permettant de créer une application facilement.

```
C:\Windows\system32\cmd.exe

C:\Users\1603344\Documents\nodejs\premiereappli>npm install express
npm WARN package.json premiereappli@1.0.0 No repository field.
npm WARN package.json premiereappli@1.0.0 No README data
express@4.17.1 node_modules\express
├── array-flatten@1.1.1
├── escape-html@1.0.3
├── methods@1.1.2
├── merge-descriptors@1.0.1
├── encodeurl@1.0.2
├── content-type@1.0.4
├── cookie-signature@1.0.6
├── etag@1.8.1
├── fresh@0.5.2
├── vary@1.1.2
├── parseurl@1.3.3
├── cookie@0.4.0
├── range-parser@1.2.1
├── content-disposition@0.5.3
├── utils-merge@1.0.1
├── serve-static@1.14.1
├── safe-buffer@5.1.2
├── path-to-regexp@0.1.7
├── setprototypeof@1.1.1
├── statuses@1.5.0
├── depd@1.1.2
├── qs@6.7.0
├── on-finished@2.3.0 (ee-first@1.1.1)
├── proxy-addr@2.0.5 (forwarded@0.1.2, ipaddr.js@1.9.0)
├── finalhandler@1.1.2 (unpipe@1.0.0)
├── debug@2.6.9 (ms@2.0.0)
├── accepts@1.3.7 (negotiator@0.6.2, mime-types@2.1.24)
├── type-is@1.6.18 (media-typer@0.3.0, mime-types@2.1.24)
├── send@0.17.1 (destroy@1.0.4, ms@2.1.1, mime@1.6.0, http-errors@1.7.3)
├── body-parser@1.19.0 (bytes@3.1.0, raw-body@2.4.0, http-errors@1.7.2, iconv-lite@0.4.24)

C:\Users\1603344\Documents\nodejs\premiereappli>
```

Un répertoire « node_modules » a été créé dans notre répertoire de projet contenant l'ensemble des librairies installées grâce à express.

On commence à coder

On crée dans notre répertoire de projet un fichier **index.js** qui contiendra le code suivant.

```
1 //on inclut express
2 const express = require('express')
3 //notre application est une express
4 const app = express();
5
6 //on définit le contenu qui va s'afficher lorsque que
7 //l'on va arriver sur l'url par défaut
8 //ici un "Hello World!"
9 app.get('/', (req, res) => {
10   res.send('Hello World!')
11 });
12
13 //on lance notre serveur
14 app.listen(8000, () => {
15   console.log('Example app listening on port 8000!')
16 });
```

On va maintenant exécuter notre script avec Node en faisant la ligne de commande :

node index.js

```
C:\Users\1603344\Documents\nodejs\premiereappli>node index.js
Example app listening on port 8000!
```

Résultat

Sur le navigateur

On peut lancer la page sur un navigateur



← → ↻ ⓘ localhost:8000

Hello World!

ON AMÉLIORE NOTRE APPLICATION

Nous allons maintenant modifier un peu notre code pour afficher du HTML sur notre première page.

```
index.js
2  const express = require('express')
3  //notre application est une express
4  const app = express();
5
6  //on définit le contenu qui va s'afficher lorsque que
7  //l'on va arriver sur l'url par défaut
8  //ici un "Hello World !"
9  app.get('/', (req, res) => {
10     res.writeHead(200, {"Content-Type": "text/html"});
11     res.end('<p>Voici un paragraphe <strong>HTML</strong> !</p>');
12 });
13
14 //on lance notre serveur
15 app.listen(8000, () => {
16     console.log('Example app listening on port 8000!')
17 });
```

On va maintenant relancer l'exécution de notre script pour que la modification soit bien prise en compte.

node index.js

```
C:\Users\1603344\Documents\nodejs\premiereappli>node index.js
Example app listening on port 8000!
```

Résultat

Sur le navigateur

On peut lancer notre nouvelle page sur le navigateur

← → ↻ localhost:8000

Voici un paragraphe HTML !

CRÉDITS

OEUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la Direction de l'ingénierie

© **AFPA**

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconques. »