

# Introduction a Node.JS

# C'est quoi Node.JS

- Node.js est un environnement de serveur open source.
- Node.js vous permet d'exécuter JavaScript sur le serveur.
- Node.JS est l'environnement d'exécution JavaScript (JavaScript Runtime Environment) de l'extérieur navigateur.
- Node.JS comprend également autres composants et autres bibliothèques qui lui permettent de fonctionner comme une Application Serveur Web.

# Node.JS utilise un type de programmation asynchrone

- Une tâche courante pour un serveur Web peut être d'ouvrir un fichier sur le serveur et de renvoyer le contenu au client.
- Voici comment PHP ou ASP gère une demande de fichier :
  - Envoie la tâche au système de fichiers de l'ordinateur.
  - Attend pendant que le système de fichiers s'ouvre et lit le fichier.
  - Renvoie le contenu au client.
  - Prêt à traiter la prochaine demande.

# Node.JS utilise un type de programmation asynchrone

- Voici comment Node.js gère une demande de fichier :
  - Envoie la tâche au système de fichiers de l'ordinateur.
  - Prêt à traiter la prochaine demande.
  - Lorsque le système de fichiers a ouvert et lu le fichier, le serveur renvoie le contenu au client.
  - Node.js élimine l'attente et continue simplement avec la requête suivante.
- Node.js exécute une programmation asynchrone à un seul thread, non bloquante, qui est très économe en mémoire.

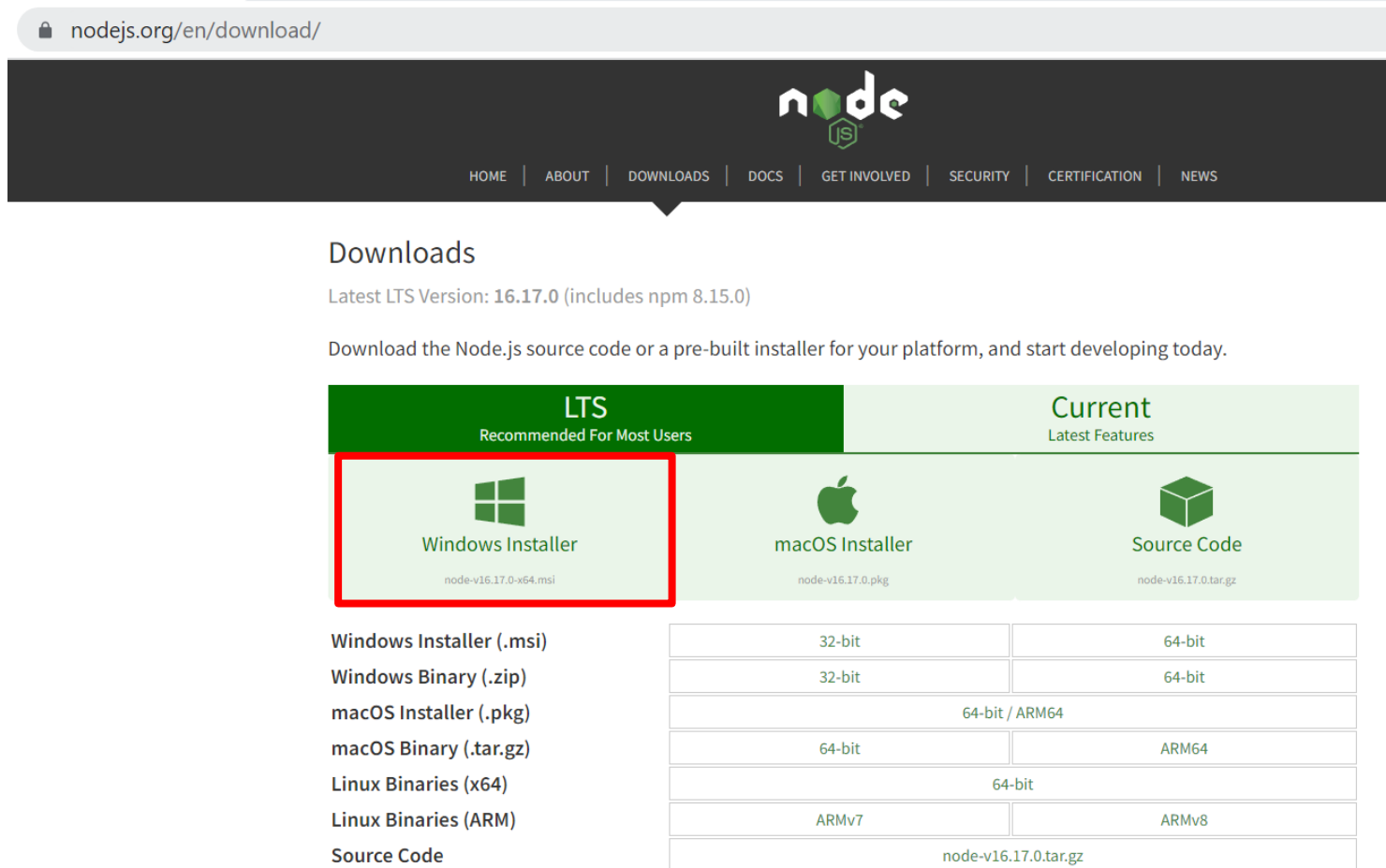
# Fonctionnalités de Node.JS

- Node.js peut générer du contenu de page dynamique
- Node.js peut créer, ouvrir, lire, écrire, supprimer et fermer des fichiers sur le serveur
- Node.js peut collecter des données de formulaire
- Node.js peut ajouter, supprimer, modifier des données dans votre base de données

# Installation de node JS

- Aller sur le site : <https://nodejs.org/en/download/> pour télécharger nodeJS et choisir Windows Installer

nodejs.org/en/download/



node

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS


## Downloads


Latest LTS Version: **16.17.0** (includes npm 8.15.0)


Download the Node.js source code or a pre-built installer for your platform, and start developing today.

**LTS**  
Recommended For Most Users

**Current**  
Latest Features

  
**Windows Installer**  
node-v16.17.0-x64.msi

  
**macOS Installer**  
node-v16.17.0.pkg

  
**Source Code**  
node-v16.17.0.tar.gz

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)


Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v16.17.0.tar.gz	

**Un fichier d'extension msi sera téléchargé et double cliquer sur le fichier pour faire l'installation.**

 node-v16.17.0-x64

# Configurer Node JS

- Inspecter la version de Node JS et de NPM
- Dans le bouton de recherche de Windows, taper la commande CMD
- On tape la commande
  - node -v
  - Npm -v



```
C:\> Invite de commandes
Microsoft Windows [version 10.0.22000.856]
(c) Microsoft Corporation. Tous droits réservés.

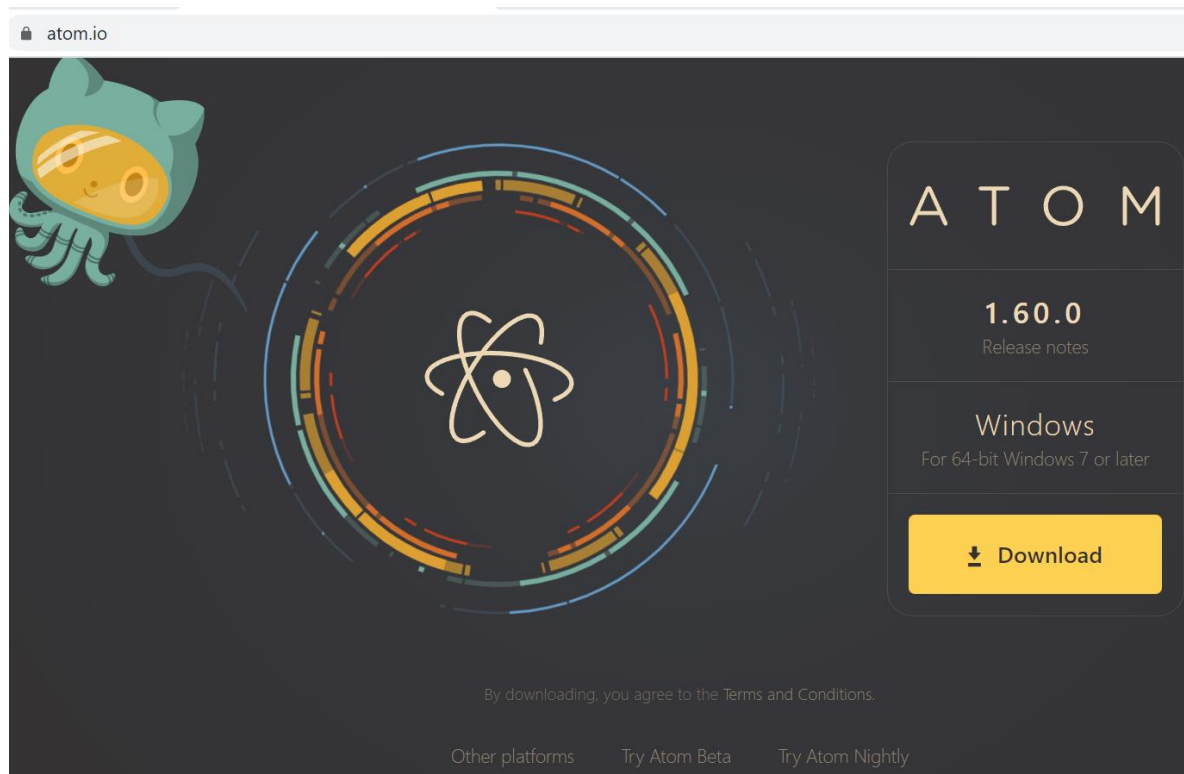
C:\Users\15147>node -v
v16.17.0

C:\Users\15147>npm -v
8.15.0

C:\Users\15147>
```

# L'éditeur de texte atom

- Atom est un bon éditeur de texte pour programmer avec Node.JS
- Pour télécharger atom, il faut cliquer sur : <https://atom.io/>





# Qu'est ce que le NPM

- NPM est l'abréviation de Node Package Manager, qui est un outil (programme) gérant les bibliothèques de programmation Javascript pour Node.js.
- Cet outil est réellement nécessaire pour le monde open source. Dans la communauté Javascript, les développeurs partagent des centaines de milliers des extraits de code qui aident les nouveaux projets à éviter de réécrire des composants de base, des bibliothèques de programmation ou même des frameworks.
- Chacun de ces extraits de code peut dépendre de nombreux autres codes source ouverts.
- Heureusement, les outils de gestion de bibliothèque sont nés. Sinon, il faudra beaucoup d'efforts pour gérer ces bibliothèques.

# NPM

- A la base, NPM est similaire à Maven. La différence ici est que le NPM gère des bibliothèques Javascript alors que Maven gères des bibliothèques Java.
- Un logiciel similaire au NPM est Yarn, développé par Facebook, avec des fonctionnalités exceptionnelles qui émergent comme une alternative du NPM.
- Dans le même cas, le logiciel de gestion de bibliothèque Java est Gradle, avec ses fonctionnalités exceptionnelles, qui remplace progressivement Maven.
- Afin d'exécuter le logiciel NPM votre système doit installer NodeJS. Le logiciel NPM est disponible dans l'installateur NodeJS, donc lorsque vous terminez l'installation de NodeJS vous obtiendrez tous les deux logiciels

# Exemple d'utilisation de npm

- Comme le NPM est un logiciel installé sur votre ordinateur, vous pouvez l'utiliser pour installer des bibliothèques Javascript à partir de l'Internet. Pour installer une bibliothèque, il est nécessaire d'ouvrir uniquement la fenêtre CMD et d'exécuter la commande comme ci-dessous :
  - `npm install <package name>`
  - `npm install vue`
  - Et vous pouvez utiliser Vue.js dans votre code Javascript :

```
** javascript **
```

```
var Vue = require('vue');
```

# Créer un projet Node.js

- Créer un dossier nommé Projet1 dans vos dossiers disponibles pour Node JS
- À partir du menu démarrer de Windows, taper CMD
- Changer de répertoire pour aller dans Projet 1
  - **Cd \NodeJS\Projet1**
- Puis exécutez la commande ci-dessous pour que NPM crée un projet pour vous.
  - **npm init**

# Création d'un projet NodeJS

```
C:\> npm init

C:\Users\15147>cd \nodejs\projet1

C:\NodeJS\Projet1>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 init` for definitive documentation on these fields
and exactly what they do.

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

Press ^C at any time to quit.
package name: (projet1)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\NodeJS\Projet1\package.json:

{
  "name": "projet1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
```

**Appuyer sur Enter  
plusieurs fois**

**Taper yes**

# Création d'un projet NodeJS

```
Is this OK? (yes) yes
npm notice
npm notice New minor version of npm available! 8.15.0 -> 8.19.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.19.1
npm notice Run npm install -g npm@8.19.1 to update!
npm notice
C:\NodeJS\Projet1>
```

Fichier JSON créé dans le dossier

« NodeJS » Projet1			Rechercher dans : Projet1
Nom	Modifié le	Type	
package	2022-09-08 04:50	Fichier source JSON	

# Installation de express et ejs

- Il y a certaines packages de bibliothèques nécessaires pour votre projet et vous devez l'installer en soutien à NPM :
  - Express
  - Express.js (Ou tout simplement Express) est un Web Application Framework de NodeJS. Il fournit un ensemble de fonctionnalités puissantes aux applications web et mobiles.
  - Ejs
  - EJS est l'abréviation de "Embedded JavaScript templating", qui est une bibliothèque, utilisée pour analyser les fichiers ejs et créer du HTML pour retourner au client (navigateur).
- Taper à la ligne de commande :
  - **npm install express ejs**

# Installation de Express et EJS



Invite de commandes

```
C:\NodeJS\Projet1>npm install express ejs

added 73 packages, and audited 74 packages in 6s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\NodeJS\Projet1>
```



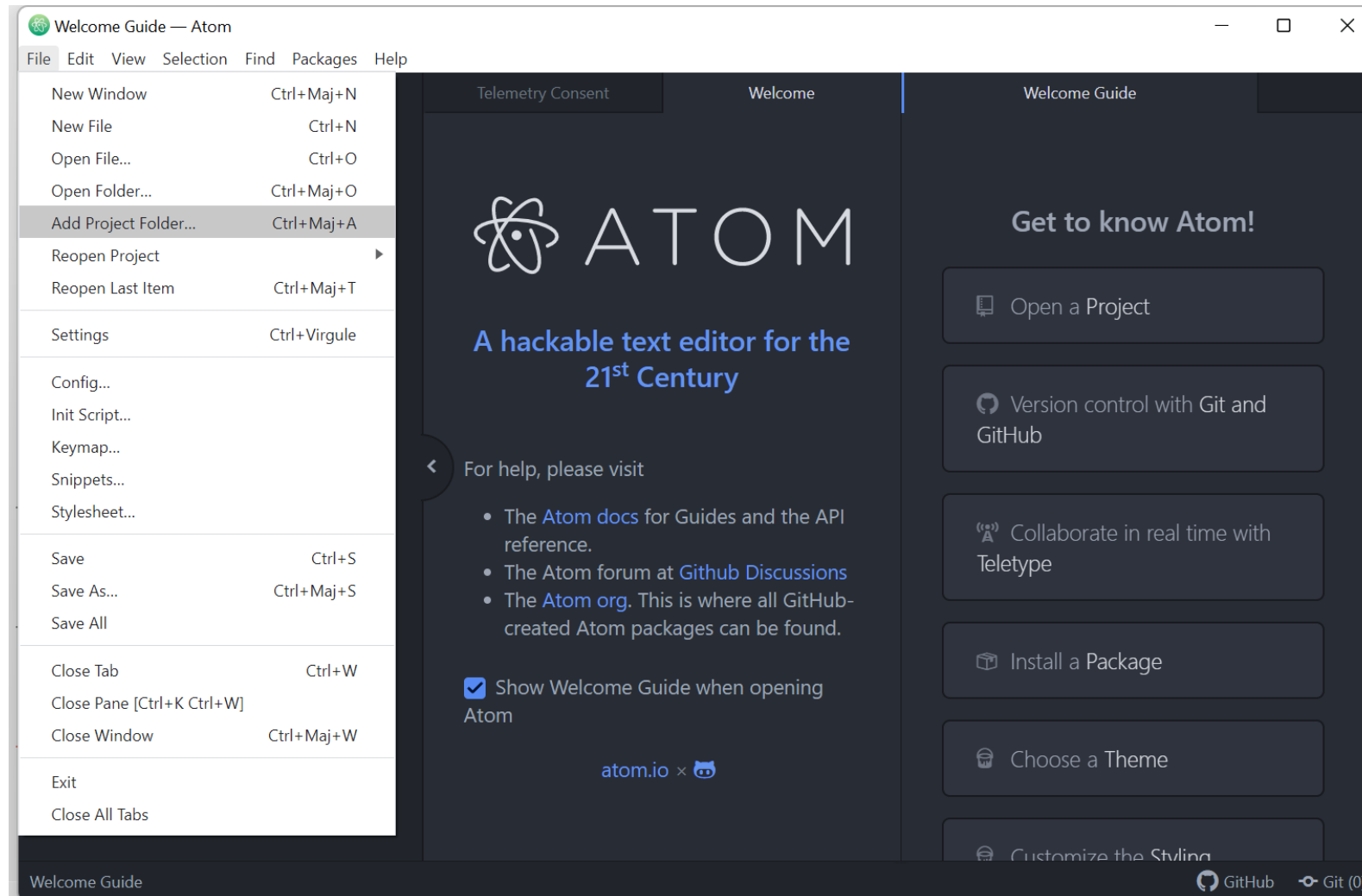
# Installation de express et ejs

- À la fin de l'installation, remarquez le changement dans le dossier projet1

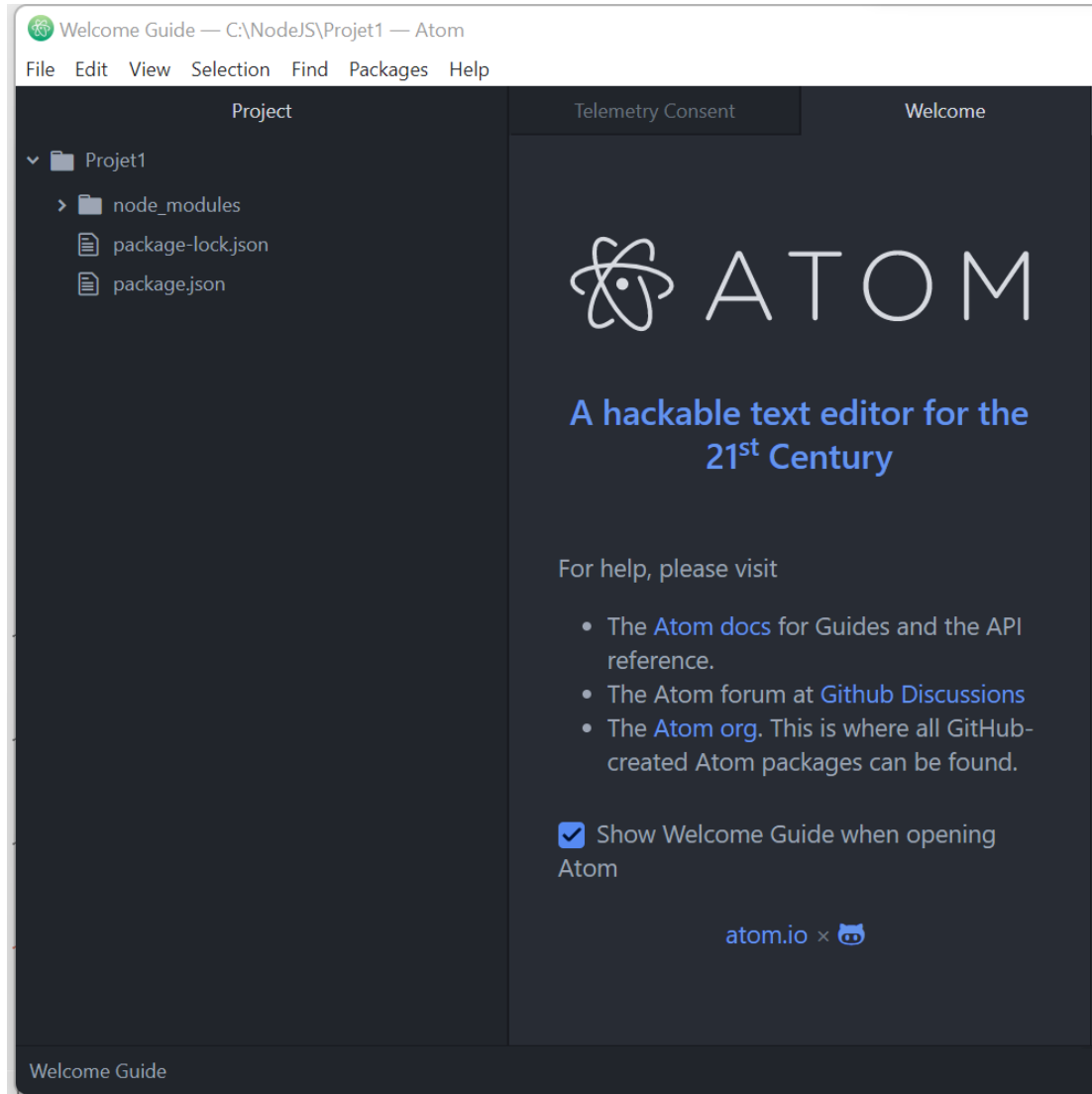
« NodeJS » Projet1		Rechercher dans : Projet1	
Nom	Modifié le	Type	Ta
node_modules	2022-09-08 04:58	Dossier de fichiers	
package	2022-09-08 04:58	Fichier source JSON	
package-lock	2022-09-08 04:58	Fichier source JSON	

# Créer la structure du projet

- Ouvrir le projet avec atom
  - File/Add Project Folder

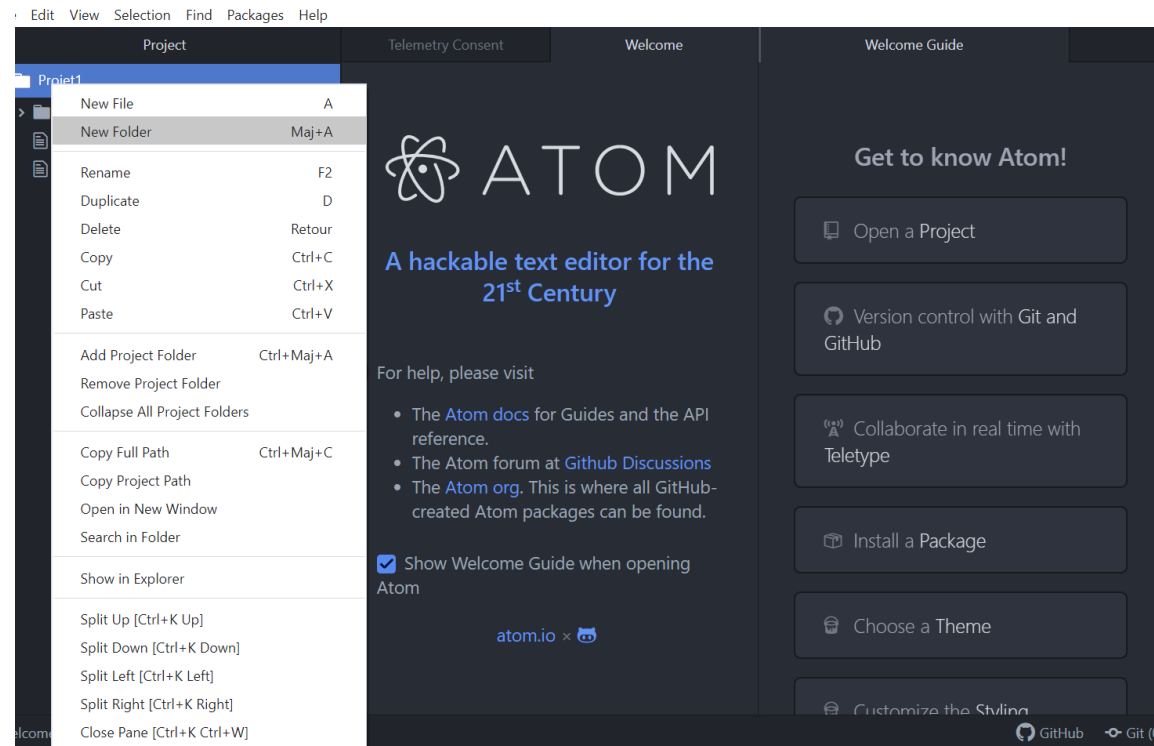


# Projet ouvert sur atom

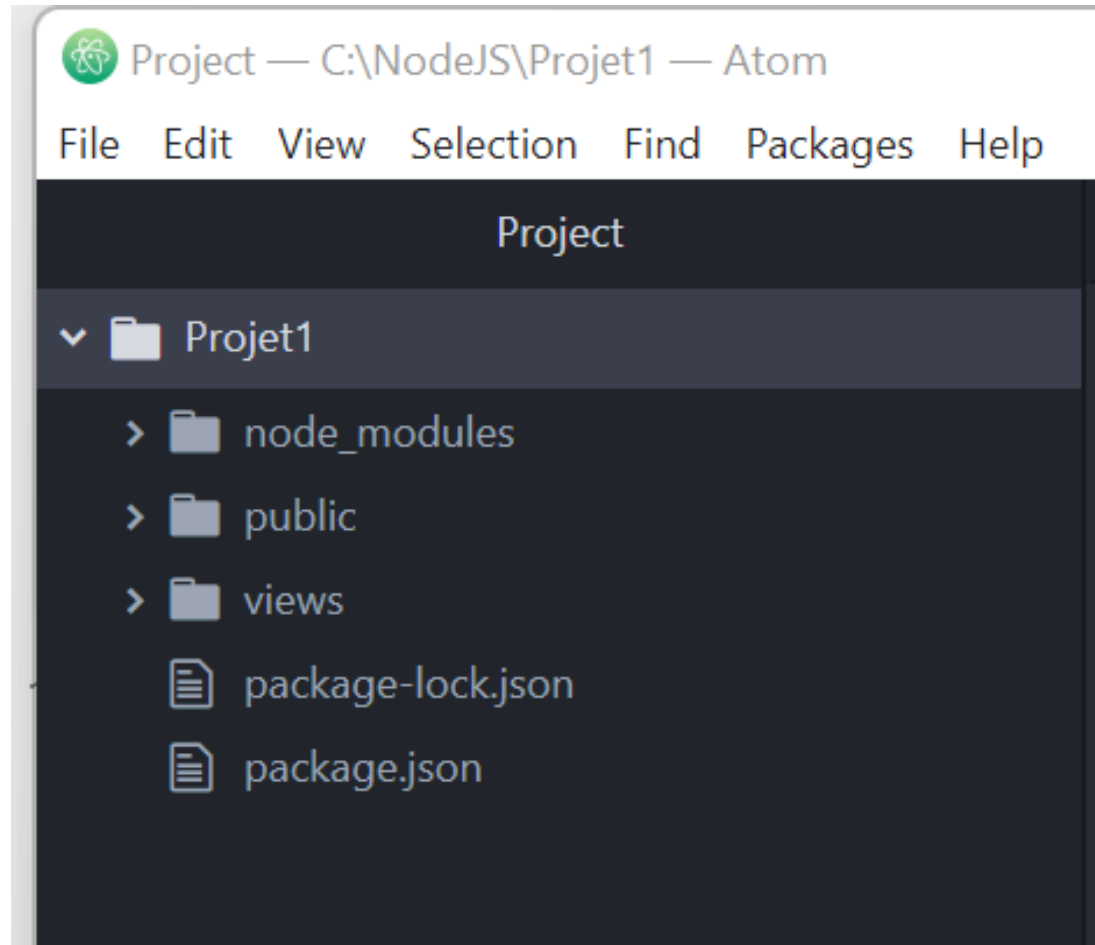


# Créer deux dossiers dans le projet

- Créer les dossiers public et Views dans le projet
  - public : Est le dossier contenant tous les fichiers que des utilisateurs peuvent accéder, par exemple image, video,...
  - views : Votre site web contient plusieurs pages, par exemple, page d'accueil, page de connexion, ... Ce dossier est l'endroit contenant toutes vos pages.
- Clic droit sur le projet/New Folder

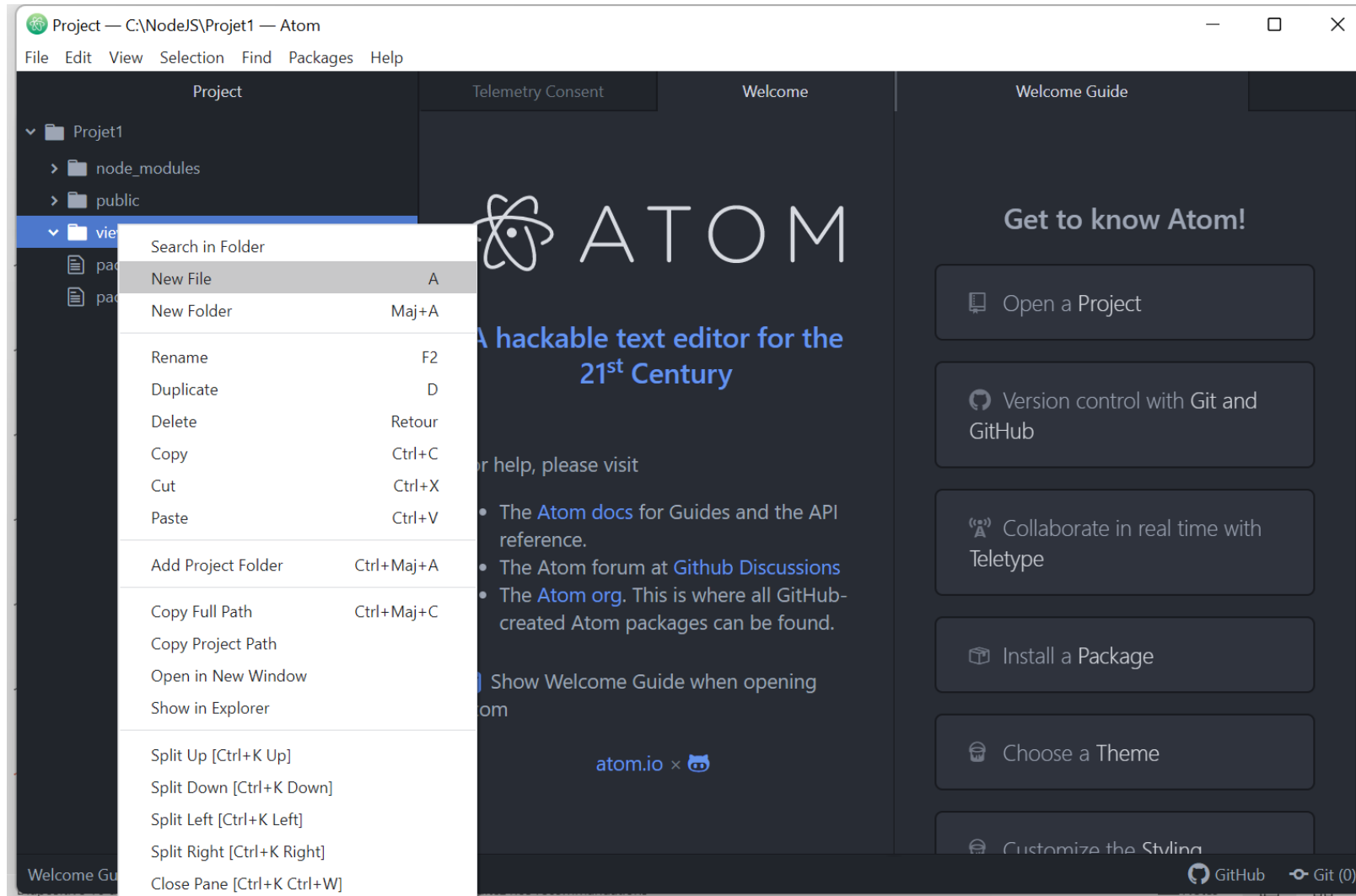


# Les dossiers public et views

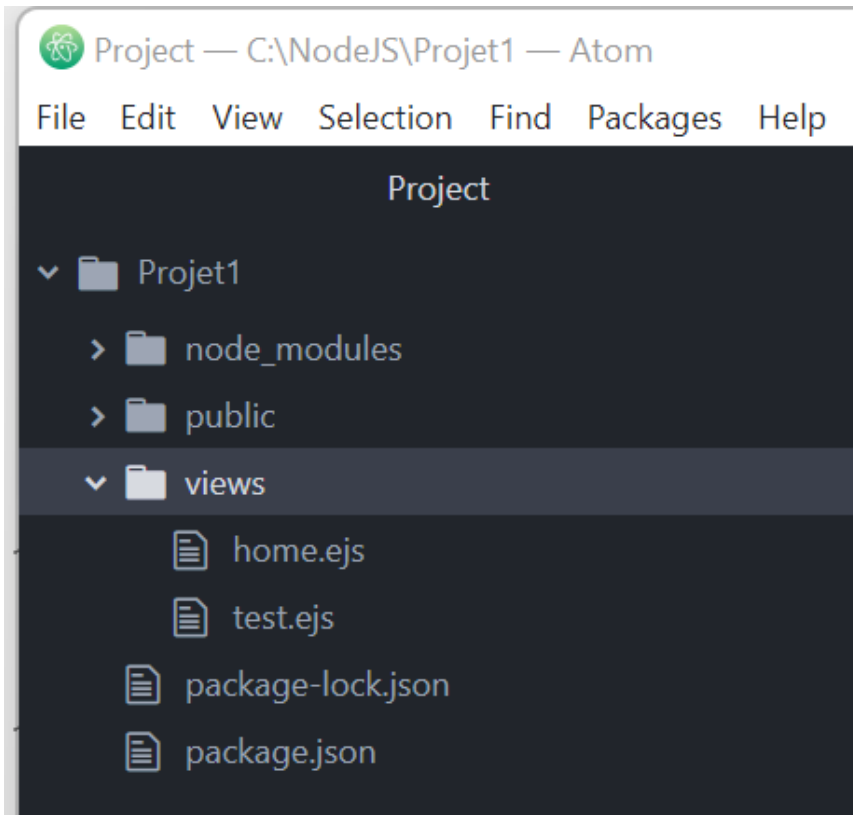


# Création des deux pages dans views

- Home.ejs
- Test.ejs



# Deux fichiers dans views



# Home.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Home</title>
  </head>
  <body>

    <h1>Ceci est la page d'accueil</h1>

  </body>
</html>
```

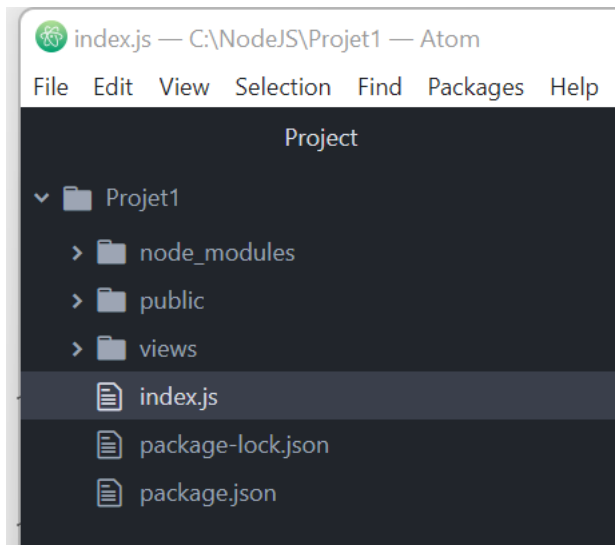


# Test.ejs

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Test</title>
6    </head>
7    <body>
8
9      <h1>Ceci est la page de test</h1>
10
11    </body>
12  </html>
```

# Créer le fichier de configuration index.js

- Ce fichier de configuration doit se trouver dans le dossier racine du projet1



# index.js

```
1  var express = require("express");
2
3  var app = express();
4
5  app.use(express.static("public"));
6
7  app.set("view engine", "ejs");
8  app.set("views", "./views");
9
10 app.listen(3000);
11
12 app.get("/", function(request, response) {
13
14     response.render("home");
15 });
16
17 app.get("/test", function(request, response) {
18
19     response.render("test");
20 });
```

# Explication du contenu de index.js

Code	Description
<code>var express = require("express");</code>	<b>ExpressJS</b> est un <b>Web Application Framework</b> , cette ligne de code indique que vous voulez l'utiliser.
<code>var app = express();</code>	Créer un objet <b>Express</b> .
<code>app.use(express.static("public"));</code>	Cette ligne de code indique <b>Application Server</b> que vous voulez utiliser le dossier <b>public</b> pour stocker des données statiques, l'utilisateur peut accéder aux fichiers dans ce dossier.
<code>app.set("view engine", "ejs");</code>	Cette ligne de code indique <b>Application Server</b> que vous voulez utiliser la bibliothèque <b>EJS</b> . C'est une machine pour gérer votre page. Le <b>EJS</b> créera du <b>HTML</b> pour retourner au navigateur de l'utilisateur.
<code>app.set("views", "./views");</code>	Cette ligne de code indique <b>Application Server</b> le chemin d'accès au dossier contenant vos pages.
<code>app.listen(3000);</code>	Votre application écoutera sur le port 3000 quand elle est déployée.
<code>app.get("/test", function(req, res) { ... });</code>	Définir le chemin d'accès à une page

# Exécuter le serveur

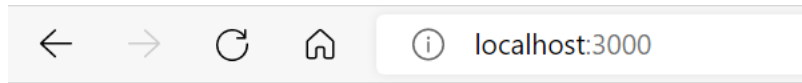
- Ouvrez les fenêtres CMD et CD dans le dossier de votre projet. Puis exécutez les commandes ci-dessous pour déployer (deploy) votre application.

Invite de commandes - node index.js

```
C:\NodeJS\Projet1>node index.js
```

- Lancer l'application à partir d'un navigateur en tapant l'adresse(à laisser le cmd ouvert) :
  - <http://localhost:3000/>
  - <http://localhost:3000/test>

https://localhost:3000/



**Ceci est la page d'accueil**

# https://localhost:3000/test



**Ceci est la page de test**

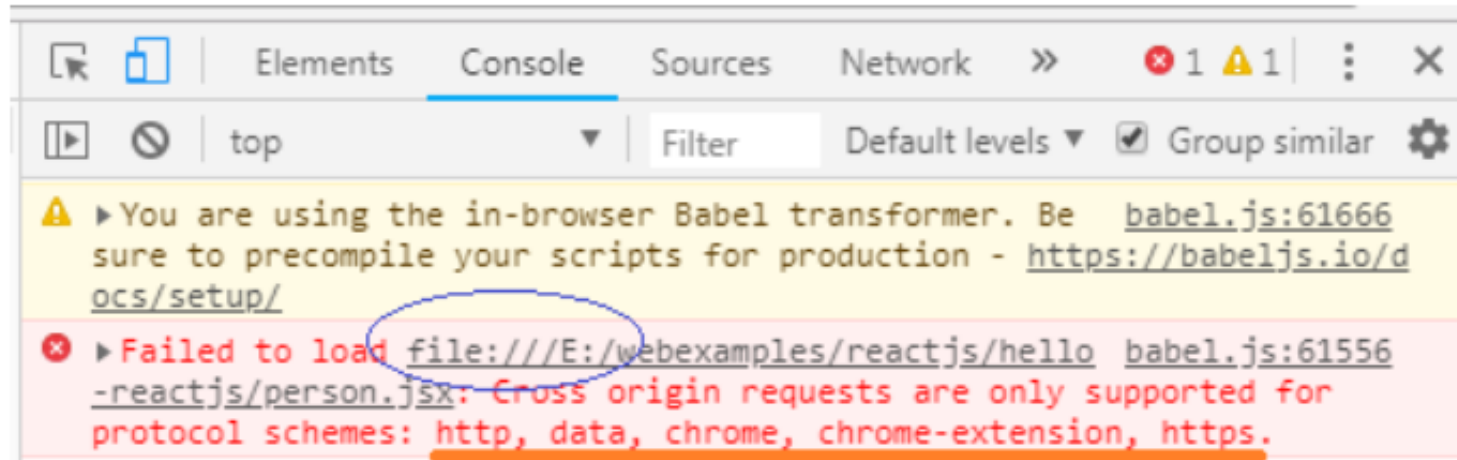
# Création d'un serveur HTTP avec Node.JS



# Pourquoi un serveur http

- Un serveur http permet d'accéder aux sources des données statiques telles que HTML, Javascript, CSS,... dans le dossier via http, tel que :
  - `http://localhost:8080/index.html`
  - `http://localhost:8080/produit.js`
- Les bibliothèques telles que ReactJS exige un serveur http parce que parfois des navigateurs empêchent l'opération des fichiers Javascript locaux, ils demandent aux fichiers Javascript de venir d'un des schema: http, data, chrome, chrome-extension, https.

# Pourquoi un serveur http



## Error Message!

Failed to load file:///E:/webexamples/reactjs/hello-reactjs/person.jsx:

Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.

# Créer un serveur http node.js


- Créer un dossier vide nommé projetHttp
- Ouvrez la fenêtre CMD et CD au dossier que vous avez créé dans l'étape précédente.

 Invite de commandes

```
C:\NodeJS\Projet1>cd \nodejs\ProjetHttp  
C:\NodeJS\ProjetHttp>
```

# Installer http-server

- Commande : `npm install -g http-server`

 Invite de commandes

```
C:\NodeJS\Projet1>cd \nodejs\ProjetHttp  
  
C:\NodeJS\ProjetHttp>npm install -g http-server  
  
added 39 packages, and audited 40 packages in 6s  
  
9 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
  
C:\NodeJS\ProjetHttp>
```

# Démarrer http-server

- Commande : `http-server -c-1`

 `http-server`

```
C:\NodeJS\ProjetHttp>http-server -c-1
Starting up http-server, serving ./
```

```
http-server version: 14.1.1
```

```
http-server settings:
```

```
CORS: disabled
```

```
Cache: -1 seconds
```

```
Connection Timeout: 120 seconds
```

```
Directory Listings: visible
```

```
AutoIndex: visible
```

```
Serve GZIP Files: false
```

```
Serve Brotli Files: false
```

```
Default File Extension: none
```

```
Available on:
```

```
http://10.0.0.222:8080
```

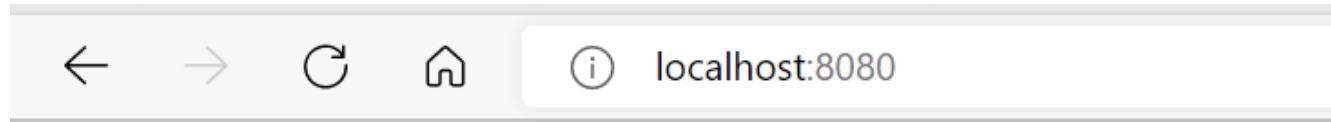
```
http://127.0.0.1:8080
```

```
Hit CTRL-C to stop the server
```

-c-1 pour désactiver la  
mise en cache

Votre HTTP Server a été démarré, il écoute au  
port 8080. Vous pouvez le vérifier en accédant  
au lien ci-dessous :  
<http://localhost:8080/>

# http://localhost:8080/



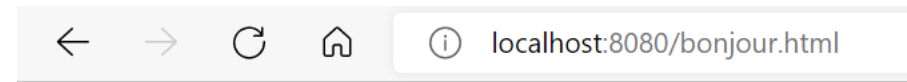
## Index of /

*Node.js v16.17.0/ [http-server](#) server running @ localhost:8080*

# Créer un fichier bonjour.html

- Créer un fichier bonjour.html dans le dossier projetHttp et vous pourrez y accéder via `http://localhost:8080/bonjour.html`

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Bonjour</title>
6    </head>
7    <body>
8
9      <h1>Bonjour tout le monde</h1>
10
11    </body>
12  </html>
```



**Bonjour tout le monde**

# NodeJS Module



# NodeJS Module

- Le module NodeJS est une bibliothèque Javascript. C'est un ensemble de fonctions (function), d'objets et de variables que vous pouvez utiliser dans vos applications. L'utilisation Module permet de simplifier l'écriture de code et de le gérer dans votre application. Normalement, chaque module sera écrit dans un fichier séparé.
- Le NodeJS intègre de nombreux Modules. Il a des bibliothèques standards pour que vous puissiez développer des applications. Voici la liste des modules :

# Module nodeJS

Module	Description
assert	ournit un ensemble des <b>assertion tests</b> (tests d'assertion)
buffer	Pour gérer les données binaires (binary data).
child_process	Pour exécuter un processus enfant (child process)
cluster	Pour diviser un seul processus (process) Node en plusieurs processus
crypto	Pour gérer les fonctions cryptographiques <b>OpenSSL</b> (OpenSSL cryptographic functions)
dgram	Fournit l'implémentation de <b>UDP sockets</b>
dns	Effectuer des recherches (lookups) et des fonctions de résolution (resolution) <b>DNS</b>
events	Pour gérer les événements (events)
fs	Pour gérer le système de fichiers
http	Pour faire <b>Node.js</b> agir comme un <b>HTTP server</b> .
https	Pour faire <b>Node.js</b> agir comme un <b>HTTPS server</b> .
net	Pour créer des <b>server</b> et des <b>client</b>
os	Fournit des informations sur le système d'exploitation
path	Pour gérer des chemins de fichiers (file paths).

# Module nodeJS

querystring	Pour gérer les chaînes de requête de URL
readline	Pour gérer les flux des données (data streams) lisibles une ligne (line) à la fois
stream	Pour gérer les flux des données (streaming data)
string_decoder	Décoder (decode) les objets du tampon (buffer objects) en chaînes
timers	Pour exécuter une fonction <b>Javascript</b> après un nombre donné de temps
tls	Pour implémenter des protocoles <b>TLS &amp; SSL</b> .
tty	Fournit des classes utilisées par <b>text terminal</b> .
url	Pour analyser (parse) les chaînes de <b>URL</b> (URL strings)
util	Pour accéder aux fonctions utilitaires (Utility functions).
v8	Pour accéder aux informations sur le moteur <b>V8 JavaScript engine</b> .
vm	Pour compiler (compile) du code <b>JavaScript</b> dans la machine virtuelle (Virtual machine)
zlib	Pour compresser ou décompresser des fichiers.