



NODE-JS / EXPRESS / MONGO-DB

par Stéphane MASCARON Architecte Logiciels Libres
Greta - Février 2019

Node, Express, MongoDB

• Introduction

Définition : « *Node.js, en abrégé Node est une plate-forme logicielle libre et évènementielle en **JavaScript** orienté vers les applications réseau qui doivent pouvoir monter en charge* ».

Node n'est donc pas un Framework ! Comme on peut le lire parfois.

On pourrait plutôt comparer Node à la JVM pour Java, c'est un environnement d'exécution de scripts JavaScript (ES5, ES6).

C'est donc un interpréteur JavaScript basé sur le moteur JS de Google : [V8](#).

Node, Express, MongoDB

- **Pourquoi Node.js**

- **Sa simplicité** : de part l'utilisation du langage JavaScript : pas de multi-thread, mono-thread d'exécution, langage de script dynamique orienté Objets peu typé, langage évènementiel.
- **Ses performances** : Node.js utilise V8 le moteur JavaScript de Google, reconnu pour ses performances et surtout sa capacité à compiler le JavaScript en langage machine (x86, ARM ou MIPS)
- **Unification du langage** : Utiliser qu'un seul langage de développement côté client et serveur dans la mesure où vous n'utilisez pas de framework à transpilation (TypeScript-Angular >2).

Node, Express, MongoDB

• Installation d'express

- *Pour créer des applications **express** il existe plusieurs possibilités, la plus souple semble être l'installation du module express-generator*

```
$ npm install express-generator -g
```

- *Ensuite il peut être nécessaire de configurer un lien dans /usr/bin pour accéder à la commande “express” du module express-generator. (Linux)*

```
$ sudo ln -s <chemin vers votre module>/express-generator/bin/express  
/usr/bin/express
```

NB : Cela peut arriver si vous utilisez l'outil “nvm” qui stocke les nodejs dans le dossier ‘**.nvm**’ de votre **/home**.

Node, Express, MongoDB

- **Installation d'express (suite)**

- Maintenant que vous avez un express, générateur d'application, vous pouvez l'utiliser pour créer votre première application Express : **--view=hbs** précise l'utilisation d'Handlebars comme moteur de template)

```
stephane@UX303UB:~/workspaceGreta$ express --view=hbs myExpressHbsApp
create : myExpressHbsApp
create : myExpressHbsApp/package.json
create : myExpressHbsApp/app.js
create : myExpressHbsApp/public
create : myExpressHbsApp/public/javascripts
create : myExpressHbsApp/routes
create : myExpressHbsApp/routes/index.js
create : myExpressHbsApp/routes/users.js
create : myExpressHbsApp/views
create : myExpressHbsApp/views/index.hbs
create : myExpressHbsApp/views/layout.hbs
create : myExpressHbsApp/views/error.hbs
create : myExpressHbsApp/bin
create : myExpressHbsApp/bin/www
create : myExpressHbsApp/public/images
create : myExpressHbsApp/public/stylesheets
create : myExpressHbsApp/public/stylesheets/style.css
install dependencies:
$ cd myExpressHbsApp && npm install
run the app:
$ DEBUG=myexpresshbsapp:* npm start
```

Node, Express, MongoDB

• Configuration première application

- *Nous allons configurer les dépendances de notre première application Express :*

```
stephane@UX303UB:~/workspaceGreta/myExpressHbsApp$ npm install
```

- *La configuration est terminée lorsque le prompt revient :*

```
(...)  
├── morgan@1.7.0  
│   ├── basic-auth@1.0.4  
│   ├── debug@2.2.0  
│   │   └── ms@0.7.1  
│   └── on-headers@1.0.1  
└── serve-favicon@2.3.2  
stephane@UX303UB:~/workspaceGreta/myExpressHbsApp$
```

Node, Express, MongoDB

- *Exécution de votre première application Express :*

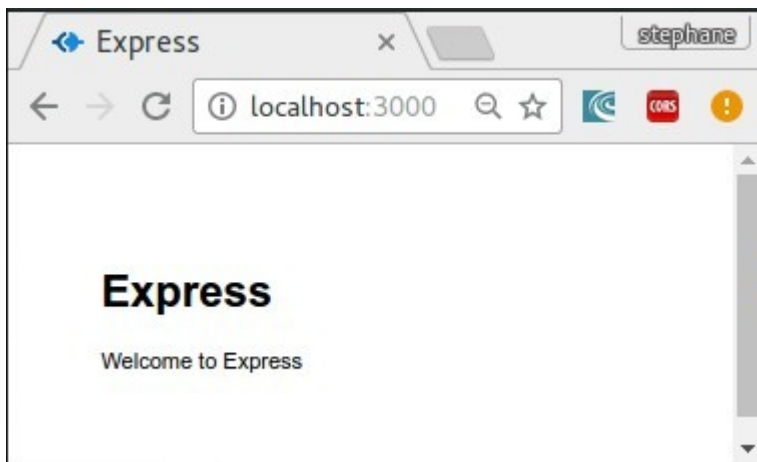
```
stephane@UX303UB:~/workspaceGreta/myExpressHbsApp$ npm start
```

```
> myexpresshbsapp@0.0.0 start /home/stephane/workspaceGreta/myExpressHbsApp  
> node ./bin/www
```

- *Lancez un navigateur sur l'url suivante :*

```
http://localhost:3000
```

- *Vous devriez voir ceci dans le navigateur et les logs dans la console:*



```
stephane@UX303UB:~/workspaceGreta/myExpressHbsApp$ npm start  
  
> myexpresshbsapp@0.0.0 start /home/stephane/workspaceGreta/myExpressHbsApp  
> node ./bin/www  
  
GET / 200 61.222 ms - 204  
GET /stylesheets/style.css 200 4.395 ms - 111
```

It Works ! ;-)

Node, Express, MongoDB

• Analyse du code généré

- *Nous allons regarder le code généré pour cette application : arborescence description :*

```
stephane@UX303UB:~/workspaceGreta/myExpressHbsApp$ tree
```

```
├── app.js .....(contrôleur)..... le code source de l'application
├── bin ..... le dossier du lanceur
│   └── www ... (main-point entrée)..... le lanceur de l'application (main)
├── package.json ..... la config de l'app + les dépendances
├── public ..... (top serveur web)..... le dossier public du serveur web node
│   ├── images ..... répertoire des images clientes
│   ├── javascripts ..... répertoire des fichiers JS clients
│   └── stylesheets ..... répertoire des feuilles de style css
│       └── style.css ..... une feuille css
├── routes ..... répertoire contenant les Contrôleurs
│   ├── index.js (code exécuté en ..... contrôleur sur action '/'
│   └── users.js fonction du pathname) contrôleur sur action '/users'
└── views .....vues-> HTML+balise moustache répertoire contenant les Vues
    ├── error.hbs .....Page écran..... Vue 'error'
    ├── index.hbs .....de l'application... Vue 'accueil' pour action '/'
    └── layout.hbs ..... Découpage de l'interface (head, body, footer, nav)
```


Node, Express, MongoDB

- **Les vues utilisent handlebars**

- *Un premier fichier est créé pour définir le “layout”, c’est à dire le découpage de l’interface :*

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    {{{body}}}
  </body>
</html>
```

- *Le code HTML des vues : “**index.hbs**” et “**error.hbs**” seront affichées à la place du tag **{{{body}}}**.*

Node, Express, MongoDB

- Découpons plus en détail l'interface.

- Il faut pour ajouter des “**partials**”, c’est à dire des blocs de HTML qui seront assemblés pour créer l’interface sur le navigateur. Dans “*app.js*” :

```
(...)  
var hbs = require('hbs');  
hbs.registerPartials(__dirname + '/views/partials', function() {  
  console.log('partials registered');  
});  
var app = express();  
(...)
```

- Ensuite vous allez dans le dossier ‘views’ créer un répertoire “**partials**”. Dans ce dossier créez un fichier nommé “**head.hbs**” :

```
<head>  
  <title>{{title}}</title>  
  <link rel='stylesheet' href='/stylesheets/style.css' />  
</head>
```

Node, Express, MongoDB

- *Découpons plus en détail l'interface*
 - *Le code du **layout.hbs** va changer lui aussi :*

```
<!DOCTYPE html>
<html>
  {{> head}}
  <body>
    {{{body}}}
  </body>
</html>
```

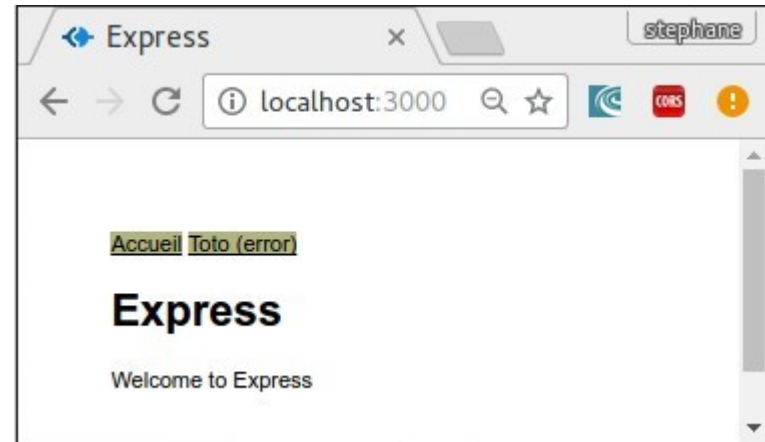
- *De la même façon nous allons ajouter une barre de navigation (lien href). On crée un fichier **nav.hbs** dans '**partials**' :*

```
<nav class="barnav">
  <ul id="nav">
    <li><a href="/">Accueil</a></li>
    <li><a href="/toto">Toto (error)</a></li>
  </ul>
</nav>
```

Node, Express, MongoDB

- *Découpons plus en détail l'interface*
 - *Le fichier **layout.hbs** a changé lui aussi :*

```
<!DOCTYPE html>
<html>
  {{> head}}
  <body>
    {{> nav}}
    {{{body}}}
  </body>
</html>
```



- *La feuille **style.css** est à modifier aussi :*

```
.barnav {
  width: 100%;
}

#nav {
  margin:0;
  padding:0;
}
(...) //→ colonne suivante
```

```
#nav li {
  display:inline;
  padding:0;
  margin:0;
}
#nav a:link, #nav a:visited {
  color:#000;
  background:#b2b580;
}
```

Node, Express, MongoDB

Debugger une application Node.js via Chrome

- Pour lancer node.js en mode Debug, il faut utiliser l'option :

node --inspect ./bin/www

- Elle peut aussi s'utiliser avec nodemon, module permettant le rechargement des fichiers modifiés dans l'éditeur de code après un CTRL+S
- Installer nodemon :

```
$ npm install -g nodemon
(...)
├── create-error-class@3.0.2
└── capture-stack-trace@1.0.1
```

- Puis lancer le Node.js en mode debug :

```
$ nodemon --inspect ./bin/www
```

Node, Express, MongoDB

– Debugger une application Node.js via Chrome

- Pour lancer l'éditeur de code Debug de chrome, lancer "chrome" et dans l'URL tapez :

chrome://inspect

- Vous devez voir apparaître l'application Node.js

- Cliquez sur le lien

"Inspect"

il ouvre le debugger de chrome dans une nouvelle fenêtre

Remote Target #LOCALHOST

Target (v10.1.0)



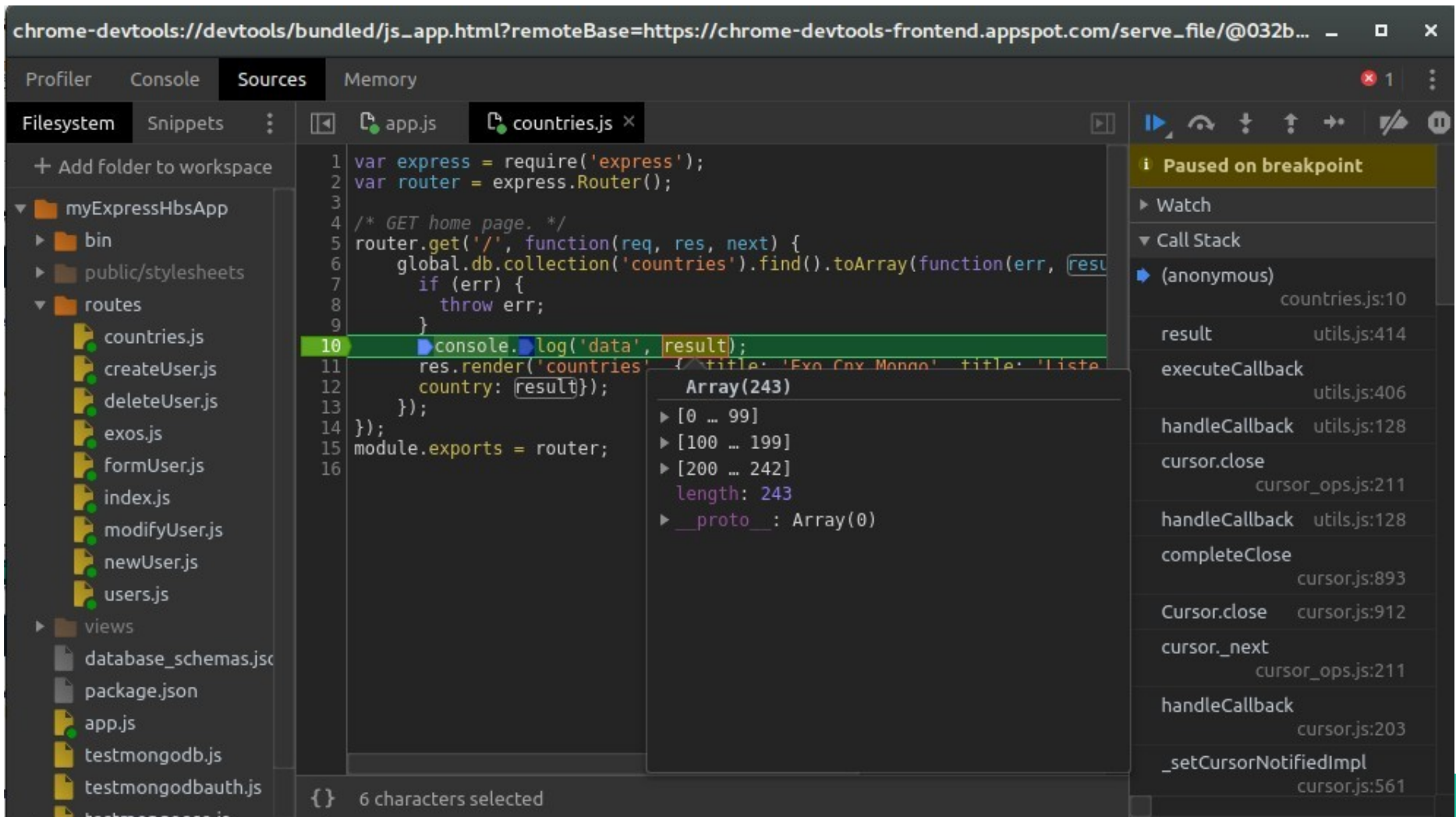
./bin/www

file:///home/stephane/Documents/formation/Clients/Simple

inspect

Node, Express, MongoDB

- **Debugger une application Node.js via Chrome : le debugger s'affiche dans une fenêtre :**



Node, Express, MongoDB

- **Exercice : ajouter un footer à notre application**
 - En utilisant les exemples précédents, ajouter un **footer.hbs** donnant le nom de votre entreprise ;-), la date, des logos twitters Facebook, ... ce que vous voulez