



Beste de savoir

Créer un serveur Express

16 janvier 2019

Table des matières

1.	Introduction	1
2.	Etape 1 : Installation	1
3.	Etape 2 : Préparation du serveur	2
4.	Etape 3 : Création de la 1ère route	3
5.	Etape 4 : Lancer le serveur	4
6.	Conclusion	4

% CRÉER UN SERVEUR EXPRESS % EtienneR % 06 juillet 2018

1. Introduction

Express est un framework pour développer des applications Web de manière simplifiée sur Node.js. Il se définit comme minimaliste, souple et rapide.

2. Etape 1 : Installation

Vérifiez que Node et NPM sont installés sur votre machine.

```
node -v && npm -v
```

Créez un nouveau dossier sur votre disque dur et placez-vous à l'intérieur.

```
mkdir test-express && cd test-express
```

Puis initialisez le manifest de l'application (**manifest.json**).

```
npm init -y
```

Ensuite, installez la dépendance d'Express en veillant à bien l'ajouter dans le manifest (option **--save**).

```
npm i --save express
```

Express est installé dans le dossier dédié aux modules **node_modules**.

3. Etape 2 : Préparation du serveur

3. Etape 2 : Préparation du serveur

Créez un fichier **index.js** qui contiendra le contenu du serveur.

```
touch index.js
```

Ouvrez le fichier du serveur dans Visual Code Studio (ou dans votre IDE favori).

```
code index.js
```

Afin d'y ajouter les 3 lignes ci-dessous.

```
1 /* index.js */
2
3 const express = require('express') // Appel de la dépendance
4 const app = express()             // Initialisation de
   l'application
5 const port = 1234                // Choix du port
```

Pour l'utilisation du port, utilisez un port non utilisé sur votre machine (au dessus de 1024).

```
netstat -nap | grep 1234
```

La commande ci-dessus doit renvoyer vide.

4. Etape 3 : Création de la 1ère route



4. Etape 3 : Création de la 1ère route

Toujours dans le même fichier (à la suite).

```
1 /* index.js (à la suite) */
2
3 app.get('/', (request, response) => {
4     response.send('Hello World!')
5 })
6
7 app.get('/user/:id', (request, response) => {
8     response.send(`user #${request.params.id}`)
9 })
10
11 app.listen(port, () => {
12     console.log(`Express running on port ${port}`)
13 })
```

- La route "/" (racine) affichera "Hello World!" (comme c'est original...);
- Une seconde route "/user/1" affichera "user #1" ou "/user/abc" affichera "user #abc";

5. Etape 4 : Lancer le serveur

— Un message s’affichera dans le terminal lors du démarrage du serveur.

5. Etape 4 : Lancer le serveur



Depuis votre terminal tapez la commande ci-dessous.

```
node index.js
```

Le terminal devrait afficher le message suivant "Express running on port 1234" et le serveur est accessible sur <http://127.0.0.1:1234> (ou suivant le port que vous avez défini).

```
1 netstat -nap | grep 1234
2
3 tcp6          0          0 :::1234          :::*
   LISTEN      24424/node
4 tcp6          0          0 127.0.0.1:1234  127.0.0.1:36140
   TIME_WAIT   -
```

6. Conclusion

Express.js est relativement facile à prendre en main. Je vous invite à regarder la documentation officielle d'ExpressJS disponible en [français](#).