



Express

Développé par **TJ Holowaychuk** et sa communauté.

La première version est sortie en **2010**.



Francis Munabeno

Pourquoi Express

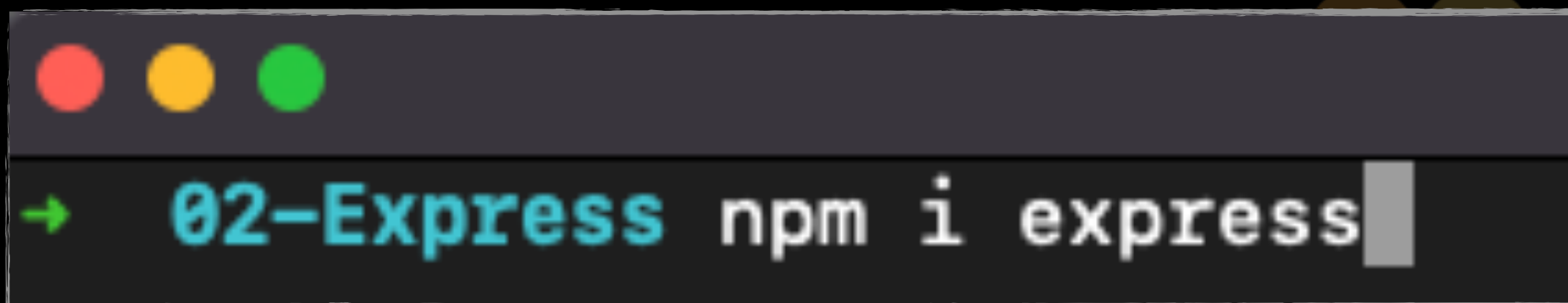
Développer un serveur pur **Node** est possible,
Mais cela vous prendra beaucoup beaucoup de temps.

Il vous faudra analyser manuellement
chaque demande entrante de votre **serveur**.

Express est un framework qui simplifie ces tâches
En nous permettant de déployer une **API**
Facilement et rapidement.

Installer Express

Pour installer Express il faut ajouter le package grâce à la commande : `npm i express`



```
→ 02-Express npm i express
```

App.js

Créer un fichier app.js qui contiendra l'application **Express**.

Ensuite il vous faudra importer le package **express** pour créer votre application.

Le **module.exports** nous permettra d'exporter notre **application** express afin de pouvoir l'utiliser dans notre **serveur node**.

```
02-Express > JS app.js > [⌘] <unknown>
1  const express = require('express')
2  const app = express()
3
4  module.exports = app
```


Server.js

La méthode `set()` de notre app, spécifie à express le port à utiliser pour le fonctionnement du `serveur`.

`CreateServer()` `return` notre application Express.

Et dans cette application on pourra utiliser la méthode `use()` de Express.

La méthode `use()` reçoit aussi deux objets `request` & `response` en arguments, Exactement comme la précédente.

```
1  const http = require('http');
2  const app = require('./app');
3
4  const PORT = process.env.PORT || 3000 // PORT
5
6  app.set('port', PORT);
7
8  const server = http.createServer(app);
9
10 server.listen(PORT, () => {
11   console.log(`Listening at http://localhost:${PORT}`);
12 });
```

App.js

Si on effectue une requête à notre **serveur**, on doit récupérer un objet JSON contenant la **response** spécifié.

C'est pour cela que nous utilisons la méthode **res.json({ })**

```
02-Express > JS app.js > ...  
1   const express = require('express')  
2   const app = express()  
3  
4   app.use(( req, res ) => {  
5     res.json({ reponse: "Hello world 🤖" })  
6   })  
7  
8   module.exports = app
```

Middlewares

Une application Express est fondamentalement une série de fonctions appelée middleware.

Chaque élément de middleware reçoit les objets `request` & `response`.

Le middleware Express reçoit également la méthode `next`, qui permet à chaque middleware de passer au middleware suivant.

Un middleware est un bloc de code qui traite les `requêtes` et `réponses` de votre application.

```
4 app.use((req, res, next) => {  
5   console.log("First middleware")  
6   next()  
7 })  
8  
9 app.use((req, res, next) => {  
10  console.log("Seconde middleware")  
11  next()  
12 })
```