

# Discord

- [Faire tourner une application node.js en permanence avec Forever](#)

# Faire tourner une application node.js en permanence avec Forever

Forever est un outil en ligne de commande permettant d'exécuter une application node.js de façon continue. Il permet de ne pas avoir à lancer `node monApp.js` en permanence et surtout de pouvoir aisément gérer l'ensemble des applications node.js qui tournent sur un serveur.

## Installation

Comme souvent dans l'univers node.js, ça s'installe avec [npm](#), le gestionnaire de paquet créé pour node.js.

On met bien l'option `-g` pour que l'installation soit globale.

```
npm install forever -g
```

## Exécuter un script en permanence

Maintenant que forever est lancé, il suffit d'utiliser la commande `forever start` pour lancer son application.

```
forever start app.js  
info:    Forever processing file: app.js
```

## Lister les application existantes

Si vous souhaitez avoir la liste de l'ensemble des applications lancées sur votre serveur, `forever list` fera votre bonheur.

```
forever list
info:    Forever processes running
data:    uid  command      script  forever pid  id logfile      uptime
data:    [0] ihf8 /usr/bin/nodejs index.js 6818    6920    /$HOME/.forever/ihf8.log
2:16:9:54.944
data:    [1] xajw /usr/bin/nodejs app.js  19646   19648   /$HOME/.forever/xajw.log
0:0:0:31.740
```

Vous avez là par exemple les deux scripts qui tournent sur mon serveur, index.js (ce blog, qui tourne sous [Ghost](#)) et app.js, une app pour un autre projet. On peut voir quelle commande est utilisée pour lancer le script (`/usr/bin/nodejs`), mais aussi l'id du processus système utilisé (pid), le chemin vers le fichier de log ainsi que le temps écoulé depuis que le script a été lancé (uptime).

Le principal problème est que si vous avez un nombre important d'applications, cela devient rapidement compliqué de les différencier : en effet, seul le nom du fichier permet de les distinguer. Une solution existe : choisir manuellement l'uid.

```
forever start --uid "myserver" server.js
```

Maintenant vous pouvez facilement repérer votre script via `forever list`.

# Arrêter et redémarrer ses scripts

Une fois lancé, les scripts peuvent être arrêtés ou relancés grâce aux commande suivantes.

## Arrêter un script

```
forever stop ID
```

Où ID peut être :

- Son id
- Son uid
- Son pid

## Arrêter tous les scripts

Il est aussi possible d'arrêter tous les scripts à la fois :

```
forever stopall
```

# Redémarrer un script

De même pour redémarrer un script.

```
forever restart ID
```

Où de nouveau, ID peut être :

- Le nom du script
- Son id
- Son uid
- Son pid

# Redémarrer tous les scripts

Et encore une fois la même chose pour tout redémarrer d'un coup.

```
forever restartall
```

# Redémarrer un script automatiquement

Il peut être très utile en phase de dev d'avoir son script automatiquement redémarrer si il est modifié. Les développeurs de Forever y ont pensé et ont ajouté une option pour cela.

```
forever start -w app.js
```

Ainsi, à la moindre modification du fichier app.js **mais aussi** de n'importe quel fichier présent dans le même dossier que app.js ainsi que dans ses sous-dossiers, app.js sera redémarrer.

# Aller plus loin

Même si forever est un outil très utile et possède des fonctionnalités intéressantes, des outils plus avancés existent, notamment [PM2](#).

Je reviendrai peut-être sur cet outil dans un futur billet.