

# Nodemon et DotEnv

## Nodemon

Lorsque l'on développe une application node js, il y a de forte chance que cette application soit ce que l'on appelle un `daemon` : **C'est un programme qui démarre et qui n'as pas de fin.**

Du coup, il est parfois difficile de modifier le code source d'un daemon sans avoir à le redémarrer. Il existe pour ça un programme s'occupant du redémarrage automatiquement pour nous, c'est `nodemon`.

L'idée est très simple, redémarrer automatiquement à chaque changement de code notre programme.

Pour mettre en place ce `nodemon`, il suffit tout d'abord de l'installer :

```
npm i -D nodemon
```

Une fois l'installation faite, il suffit de le paramétrer dans le script de notre package.json :

```
"start:dev": "ts-node src/index.ts",  
"start:daemon": "nodemon -e .ts,.js,.json -w src -x \"npm run start:dev\""
```

## Configuration et de variables d'environnements

Pour mettre en place de la configuration pour notre application, il faut utiliser des **variables d'environnement**. C'est un format universelle pour toutes application "moderne". Afin de créer de la configuration pour notre application nous devons créer un fichier `.env` à la racine de notre projet. Ce fichier contiendra toute la configuration de notre application.

Voici un exemple de fichier `.env`

```
# Le dièse représente un commentaire. Ensuite c'est un fichier  
# simple contenant des clefs en majuscule et des valeurs :  
DATABASE_URL="mongodb://....."  
DEFAULT_RESULT_NUMBER=25  
USERNAME=root
```

## Lire et récupérer des variables d'environnement

Pour pouvoir récupérer et lire des variables d'environnement en nodejs il faut tout d'abord installer `dotenv` :

```
npm i dotenv
```

Maintenant il faut changer notre script de démarrage (`npm run start` et `npm run start:dev`) :

```
"start": "node -r dotenv/config dist/index.js",  
"start:dev": "ts-node -r dotenv/config src/index.ts"
```

Nous pouvons accéder à n'importe quelle configuration n'importe où dans notre code en utilisant :

```
process.env.NOM_DE_LA_VARIABLE_ENVIRONNEMENT
```

## Attention au versionnement !

Il est **formellement déconseillé voir même très dangereux** de partager sur github votre fichier de configuration `.env`. Il est donc conseillé de s'organiser de cette manière :

1. On ignore le fichier `.env` dans le `.gitignore`
2. On crée un fichier **d'exemple** `.env.dist` contenant les exemples de toutes les valeurs de configuration. Attention aucune informations sensible ne doit être présent dans ce fichier `.env.dist`
3. Lors de l'installation du projet, il nous faudra copier `.env.dist` en `.env` et éditer les valeurs de configuration