NodeJS : CRUD – .env

- Modifier la configuration
- Pourquoi utiliser un fichier .env ?
- Comment utiliser un fichier .env ?
  (avec le module dotenv)
- Fichiers à modifier
  - Fichier de connexion à la DB

Modifier la configuration

- Actuellement, on utilise un fichier **dbconfig.js** pour conserver les données de connexion vers la base de données
- Ce n’est pas une bonne pratique
- On préfère utiliser un **fichier .env** qui peut améliorer la sécurité, faciliter la gestion des configurations et la collaboration entre développeurs. Et à ajouter dans le **.gitignore**
- Ces données pourront même être **stockées directement sur le serveur**

Pourquoi un fichier .env ?

- **Sécurité** : un fichier .env permet de stocker des données sensibles comme des clés API, des mots de passe et d'autres configurations en dehors du code source [lire ici]
- **Gestion facilitée** : on peut avoir plusieurs configurations pour différents environnements (développement, test, production) [lire ici]
- **Collaboration** : un fichier-type **.env.exemple** peut servir de modèle pour indiquer les variables à utiliser

Comment l’utiliser ?

- Installer le package **dotenv**

  npm install dotenv --save

- Créer un **fichier .env** à la racine de son projet et ajouter les variables d’environnement

  DB_HOST = "localhost"

  DB_USER = "root"

  DB_PASS = ""

  DB_NAME = "nom_de_la_base"

  DB_PORT = 3306

Lire le fichier .env

- **Charger** les variables d'environnement au début du fichier principal (exemple : app.js ou index.js)
- Utiliser le module dotenv et l'exploiter avec l'objet global **process** qui donne accès à ces variables (et permet aussi de gérer les processus Node.js, ses ressources, ses événements,…)
- Exemple :

```
require('dotenv').config();

console.log(process.env.DB_HOST); // Affiche
  'localhost'
```

Fichiers à modifier

- Maintenant que les données de connexion sont dans le **fichier .env**, on n’a plus besoin de **config/dbconfig.js** , on peut donc le **supprimer.**
- On doit adapter le fichier : **models/db.js**
- NB : le fichier **package.json** a déjà été modifié quand on a installé le module dotenv

Fichier db.js

- On modifie le fichier de connexion à la DB : models/db.js

```
// importer le module mysql (qui a été installé préalablement avec npm install mysql)
const mysql = require('mysql');

// importer les paramètres de connexion lus dans un fichier .env
require('dotenv').config();

// Créer la connexion avec la base de données
const connection = mysql.createConnection({
    host: process.env.DB_HOST,
    user: process.env.DB_USER,
    password: process.env.DB_PASS,
    database: process.env.DB_NAME,
    port: process.env.DB_PORT
});

// Ouvrir la connexion
connection.connect(function(error){
    if (error) throw error; // si erreur de connexion, on s'arrête ici
    console.log("Connecté avec succès à la base de données : " + process.env.DB_NAME);
});

// on exporte la connexion à la db, pour qu'elle soit disponible pour les autres modules
module.exports = connection;
```