

Mission

Transformation Express-Template en modèle MySQL

Rédigé par

David ROUMANET
Professeur BTS SIO



Changement

Date	Révision

Sommaire

A Modifications.....	1
A.1 Fichier d'origine.....	1
A.2 Fichier destination.....	1

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Modifications

L'objectif est de rendre ce projet compatible avec une base MariaDB/MySQL alors qu'il est initialement prévu pour MongoDB

A.1 Fichier d'origine

Le fichier à modifier ne concerne que les données et particulièrement la manière d'écrire ces données dans une base de données.

Ici, la base utilise MongoDB (une base de données noSQL).

A.2 Fichier destination

Le fichier **MedecinModel.js** doit utiliser le framework **mysql2-promise** pour remplacer MongoDB. Il faut donc installer ce framework avec npm :

```
npm mysql2-promise
```

Puis il faut importer cette librairie dans le code, au début :

```
const mysql = require('mysql2/promise')
```

Ensuite, il faut préparer la connexion. Il est recommandé d'utiliser un pool de connexions, pour pouvoir utiliser plusieurs requêtes simultanément (plusieurs clients).

```
const pool = mysql.createPool({
  host: 'localhost',
  user: 'votre_utilisateur',
  password: 'votre_mot_de_passe',
  database: 'votre_base_de_donnees',
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0
})
```

Puis il faut modifier les requêtes comme getMedecins() :

```
async function getmedecin() {
  const [rows] = await pool.query('SELECT * FROM medecins')
  return rows
}
```



Note : la fonction est de type asynchrone, car les doit attendre que la requête transmise retourne un résultat, grâce au **await** qui s'applique sur une promesse (pool.query).

Pour effacer un médecin, la fonction suivante sera utilisée :

```
/**
 * @description DELETE : Supprimer un médecin
 * @param {number} medecinID
 * @returns {Object} Résultat de la suppression
 */
async function deletemedecin(medecinID) {
  const [result] = await pool.execute('DELETE FROM medecins WHERE id = ?', [medecinID])
  return result
}
```

Ici, on exécute la requête en passant l'ID du medecin à supprimer.

L'intérêt du modèle dans le MVC, est que la description des fonctions reste la même, malgré le changement de moteur de base de données.

La fonction d'édition d'un médecin `editmedecin(newMedecin)` est l'une des plus complexes, voici un exemple original, basé sur la transmission des éléments, via un tableau :

```
async function editmedecin(newMedecin) {
  const query = `
    UPDATE medecins
    SET nom = ?, specialite = ?, ville = ?, CP = ?,
        telephone_fixe = ?, telephone_mobile = ?
    WHERE id = ?
  `
  const values = [
    newMedecin.nom.toUpperCase(),
    newMedecin.specialite,
    newMedecin.ville.toUpperCase(),
    newMedecin.CP,
    newMedecin.telephone.fixe,
    newMedecin.telephone.mobile,
    newMedecin.id
  ]

  const [result] = await pool.execute(query, values)
  return result
}
```