

Matière : SoA et Microservices**Enseignant :** Dr. Salah Gontara**Classe :** 4Info**OBJECTIF(S):**

- Mettre en place la gestion des **CORS** pour autoriser les requêtes multi-origines
- Sécuriser l'API avec un mécanisme de **Rate Limiting** (limitation du nombre de requêtes)

OUTILS UTILISÉS :

Nodejs, Express js, cors, express-rate-limit

Prérequis

- Vous disposez déjà d'une API Express fonctionnelle (celle du TP précédent avec SQLite3 et les routes CRUD sur /personnes).
- Node.js et npm sont installés sur votre machine.

Étape 1: Installation des Modules Nécessaires

1. Ouvrez un terminal et naviguez vers votre dossier.
2. Installez Cors et express-rate-limiting avec :

npm install cors express-rate-limit

Étape 2: Mise à Jour du projet

1. Voici le début de fichier index.js intégrant à la fois CORS et Rate Limiting. Ce fichier s'appuie sur votre API existante.

```

const express = require('express');
const cors = require('cors');
const rateLimit = require('express-rate-limit');
const db = require('./database');
const app = express();
const PORT = 3000;

// Middleware pour parser le JSON
app.use(express.json());

// 1. Configuration CORS : Autoriser toutes les origines
app.use(cors());

// Pour restreindre aux domaines autorisés, décommentez et adaptez la ligne suivante :
// app.use(cors({ origin: ['http://localhost:3000, 'http://localhost:4200'] }));

// 2. Configuration du Rate Limiting : 100 requêtes/15 min
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100, // Limite chaque IP à 100 requêtes par fenêtre
  message: 'Trop de requêtes effectuées depuis cette IP, veuillez réessayer après 15 minutes.'
});
app.use(limiter);

//RESTE DU PROJET

//DEFINITION DES ROUTES

```

2. Ajouter cette partie à votre projet existant.

Étape 3: Test de l'Implémentation

1. Tester CORS : Créez un petit fichier HTML avec JavaScript pour effectuer une requête vers votre API. Par exemple.
2. Dans *index.js*, importez Express, copiez ce code et exécutez avec *node index.js* :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Test CORS</title>
</head>
<body>
  <script>
    fetch('http://localhost:3000/personnes')
      .then(response => response.json())
      .then(data => console.log(data))
      .catch(error => console.error('Erreur :', error));
  </script>
</body>
</html>
```

3. Ouvrez ce fichier dans un navigateur et vérifiez dans la console que les données sont bien récupérées.
4. Tester Rate Limiting : Simulation de nombreuses requêtes :
 - Utilisez Postman pour envoyer plus de 100 requêtes en moins de 15 minutes depuis la même IP.
 - Attente d'un message d'erreur :
 - Une fois la limite atteinte, vous devriez recevoir le message défini :

```
{
  "message": "Trop de requêtes effectuées depuis cette IP, veuillez réessayer après 15 minutes."
}
```