

Atelier 04

Objectifs d'apprentissages :

- Création d'une API Rest avec express
- Test de l'API avec Postmen
- Utilisation du module fs, uuid

Le module uuid :

Uuid est l'acronyme de **Universally Unique Identifier**, il permet d'identifier un objet.

Ce package doit être installé avant l'utilisation :

```
npm install uuid
```

Pour l'utiliser dans un programme node, il faut importer le module uuid :

```
const { v4: uuid } = require("uuid");
```

Le code suivant permet de récupérer un id :

```
const id = uuid();
```

Le module fs :

Node fs, module nodejs pour "File System", permet aux développeurs de créer et gérer des fichiers pour y stocker ou lire des fichiers dans un programme Node.

Le module fs est livré nativement avec Node js. Pour l'inclure dans votre projet, utilisez simplement la fonction require :

```
const fs = require("fs");
```

Les fonctions que nous allons utiliser dans ce module :

- **existsSync** : Cette méthode est utilisée pour vérifier de manière synchrone si un fichier existe déjà dans le chemin donné ou non. Il retourne une valeur booléenne qui indique la présence d'un fichier.

La syntaxe d'utilisation est : fs.existsSync(path)

- **mkdirSync** : Cette méthode permet de créer un répertoire.

La syntaxe d'utilisation est : fs.mkdirSync(path)

- **writeFileSync** : utilisée pour écrire dans un fichier
- **readdirSync** : utilisée pour lire un contenu d'un dossier.
- **readFileSync** : utilisée pour lire un contenu d'un fichier.
- **unlink** : utilisée pour supprimer un fichier.

Travail a Réaliser :

L'objectif principale de cet atelier est de créer une API pour gérer la liste de produits d'une entreprise commerciale.

1. Créer un dossier avec le nom atelier04-express.
2. Installer dans ce dossier les packages suivants :
 - a. Express
 - b. Uuid
 - c. Cors
 - d. Nodemon
3. Créer le programme index.js dans le dossier de votre projet.
4. Créer les routes suivantes :

Route	Méthode	Description
/livre	POST	<p>- Créer un dossier nommé "data" s'il n'existe pas</p> <p>- Créer un ID unique, puis créer un fichier avec le nom "ID.txt" dans le dossier "data"</p> <p>- Le fichier texte crée, sera rempli par le JSON reçu dans le corps de la requête. Voici un exemple d'un JSON qui peut être envoyé dans le corps de la requête :</p> <pre>{ "nom": "produit 1", "description": "Description du produit 1", "famille": "Epicerie", "image": "image1.jpg", "prix": 120.3 }</pre>
/produits/all	GET	<p>Retourner la liste de tous les produits sous format JSON :</p> <pre>[{ "id": "23c402ee-1d17-41fd-a7a7-9d93fc353bc2", "nom": "produit 1", "description": "Description du produit 1", "famille": "Epicerie", "image": "image1.jpg", "prix": 120.3 }, ...]</pre>
/produits/:id	GET	Retourner un objet JSON du produit dont l'ID est égale a ID passé en paramètre.
/produits/:famille	GET	Retourner un tableau d'objets JSON des produits de la famille passée en paramètre.
/produits/:id	PUT	<p>Modifie les informations d'un produit:</p> <ul style="list-style-type: none">- ID du produit est passe en paramètre.- Les nouvelles informations sont passés dans le corps de la requête.
/produits/:id	DELETE	Supprime le fichier texte du produit dont ID est passe en

		paramètre.
--	--	------------

5. Tester votre API avec postmen.