

Bouallegue Lina
Date : 30-09-2025

Compte Rendu – TP MERN Semaine 1

Cours : Fondations du Back-end avec Node.js et Express

Objectifs du TP:

L'objectif de ce TP est de découvrir les bases du développement back-end avec Node.js et Express.js, de comprendre le concept d'architecture MERN et l'approche "API-First", puis de créer une première API simple que l'on testera avec Postman.

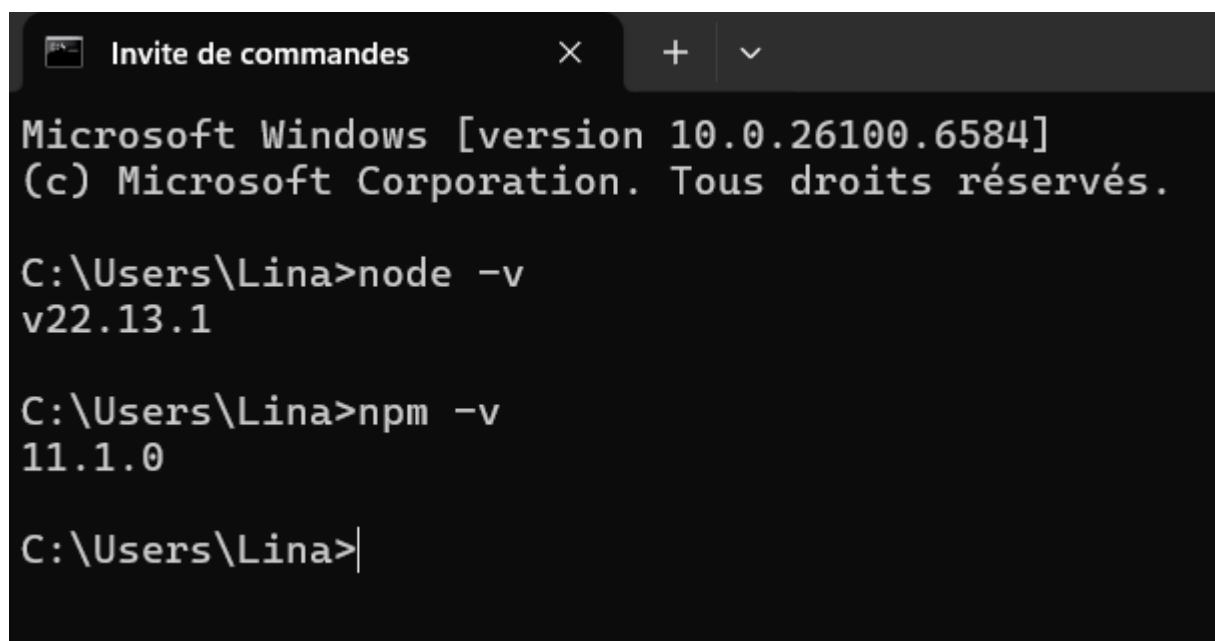
À la fin, je veux :

- Créer un projet Node.js avec NPM
- Utiliser Express pour créer un serveur
- Gérer des routes GET et POST
- Tester des endpoints avec Postman

Étape 1 : Configuration de l'Environnement de Développement

1-Installation de Node.js (version LTS)

Vérification :



```
Microsoft Windows [version 10.0.26100.6584]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Lina>node -v
v22.13.1

C:\Users\Lina>npm -v
11.1.0

C:\Users\Lina>
```

2-Installation de VS Code avec les extensions :

- ESLint
- Prettier
- DotENV
- GitLens

3-Installation de Postman pour tester les requêtes HTTP.

Étape 2 : Démarrage du Projet et Dépendances

Création d'un nouveau dossier :

```
Invite de commandes
C:\Users\Lina>mkdir Mon-Api-Blog
Un sous-répertoire ou un fichier Mon-Api-Blog existe déjà.

C:\Users\Lina>mkdir mon-api-blog
Un sous-répertoire ou un fichier mon-api-blog existe déjà.

C:\Users\Lina>cd mon-api-blog

C:\Users\Lina\mon-api-blog>npm init -y
Wrote to C:\Users\Lina\mon-api-blog\package.json:

{
  "name": "mon-api-blog",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

Installation des dépendances :

npm install express

npm install nodemon --save-dev

```
C:\Users\Lina\mon-api-blog>npm install express

up to date, audited 96 packages in 983ms

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

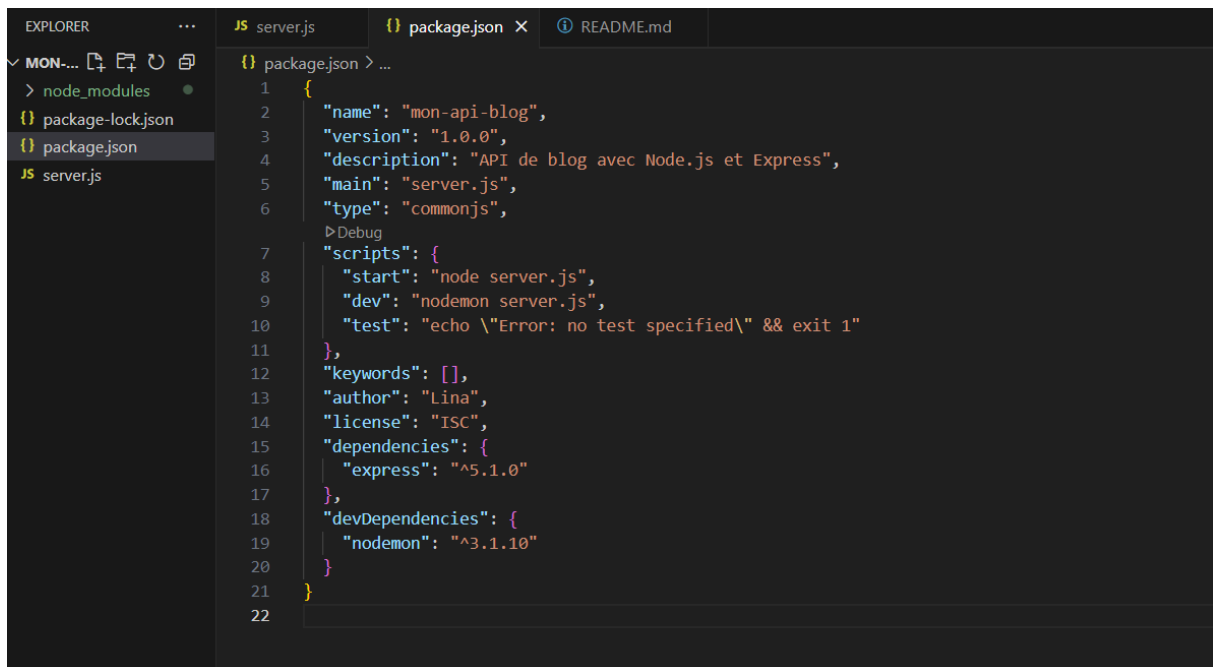
C:\Users\Lina\mon-api-blog>npm install nodemon --save-dev

up to date, audited 96 packages in 1s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Ajout des scripts dans package.json :

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named 'MON-...' with a folder 'node_modules' and files 'package-lock.json', 'package.json', and 'server.js'. The main editor area displays the 'package.json' file. The file contains the following JSON structure:

```
1 {
2   "name": "mon-api-blog",
3   "version": "1.0.0",
4   "description": "API de blog avec Node.js et Express",
5   "main": "server.js",
6   "type": "commonjs",
7   "scripts": {
8     "start": "node server.js",
9     "dev": "nodemon server.js",
10    "test": "echo \\\"Error: no test specified\\\" && exit 1"
11  },
12  "keywords": [],
13  "author": "Lina",
14  "license": "ISC",
15  "dependencies": {
16    "express": "^5.1.0"
17  },
18  "devDependencies": {
19    "nodemon": "^3.1.10"
20  }
21 }
```

Étape 3 – Création du serveur Express

Fichier : server.js

```
MON-API-BLOG
> node_modules
{} package-lock.json
{} package.json
JS server.js

JS server.js > ...
1  const express = require('express');
2  const app = express();
3  const PORT = 3000;
4
5  // Middleware pour lire le JSON
6  app.use(express.json());
7
8  // Route d'accueil
9  app.get('/', (req, res) => {
10   | res.status(200).send('<h1>Bienvenue sur l'API de Blog !</h1>');
11   | });
12
13  // Route de test
14  app.get('/api/test', (req, res) => {
15   | res.status(200).json({ message: 'Le test a fonctionné !', success: true });
16   | });
17
18  // Route POST pour créer un article
19  app.post('/api/articles', (req, res) => {
20   | const articleData = req.body;
21   | console.log('Données reçues :', articleData);
22   | res.status(201).json({
23   |   | message: 'Article créé avec succès !',
24   |   | article: { id: Date.now(), ...articleData }
25   | | });
26   | });
27
28  // Lancer le serveur
29  app.listen(PORT, () => {
30   | console.log(`Serveur démarré sur http://localhost:${PORT}`);
31   | });
32
```

```
found 0 vulnerabilities
PS C:\Users\Lina\mon-api-blog> npm run dev

> mon-api-blog@1.0.0 dev
> nodemon server.js

> mon-api-blog@1.0.0 dev

> mon-api-blog@1.0.0 dev

> mon-api-blog@1.0.0 dev
> mon-api-blog@1.0.0 dev
> mon-api-blog@1.0.0 dev
> nodemon server.js

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Serveur démarré sur http://localhost:3000
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
[nodemon] starting `node server.js`
```

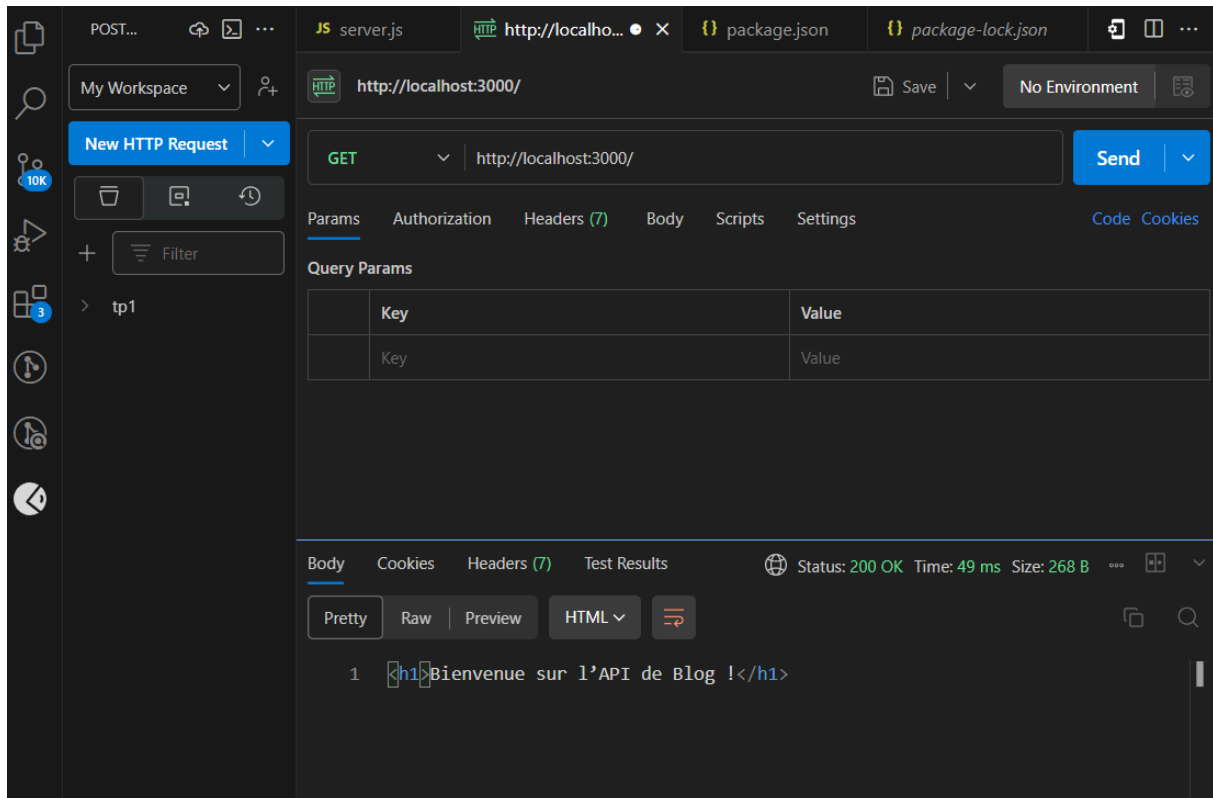
Étape 4 – Tests avec Postman

Test 1 : Route d'accueil /

Méthode : **GET**

URL : `http://localhost:3000/`

Résultat : Affiche le message *“Bienvenue sur l’API de Blog !”*

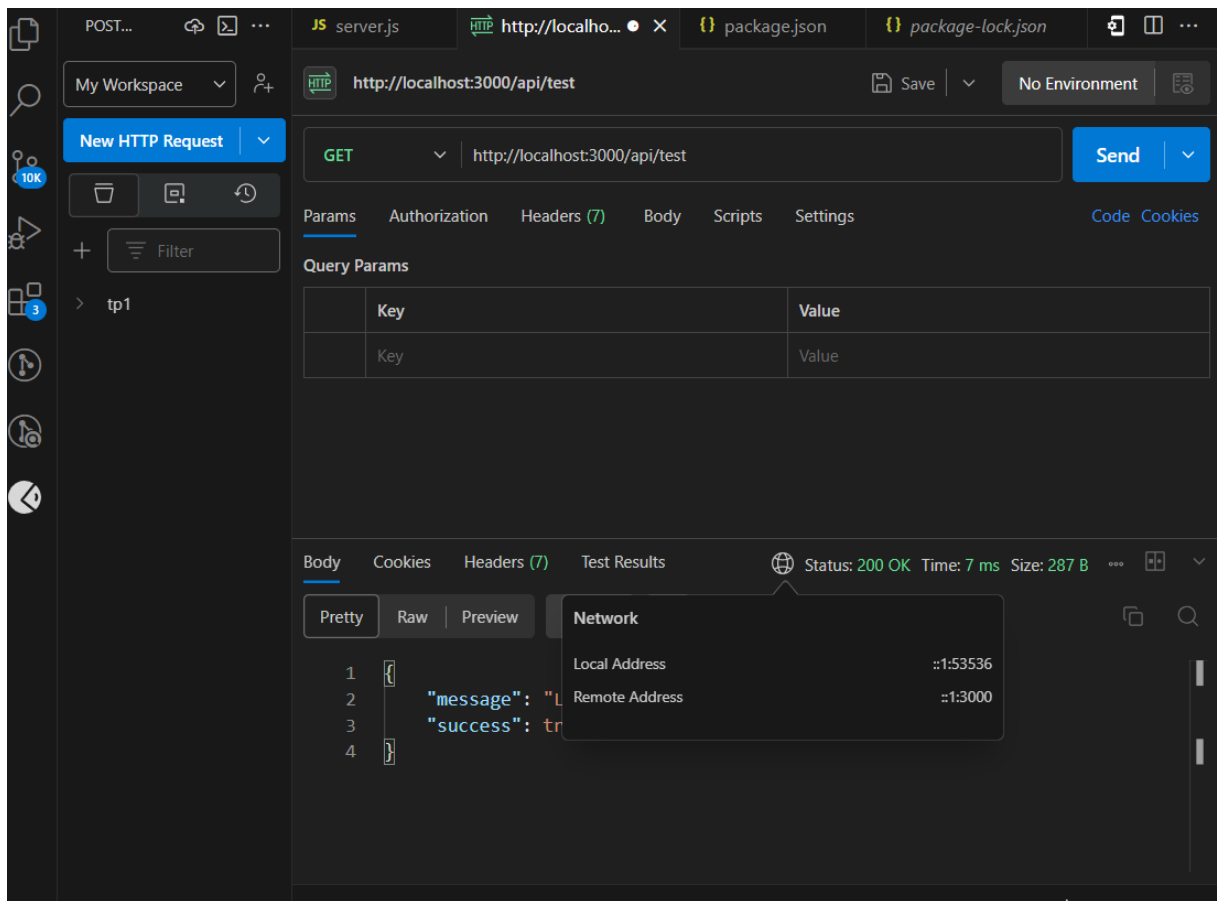


Test 2 : Route de test `/api/test`

Méthode : **GET**

URL : `http://localhost:3000/api/test`

Réponse JSON :



Test 3 : Route POST /api/articles

- Méthode : **POST**
- URL : `http://localhost:3000/api/articles`
- Onglet **Body** → **raw** → **JSON**

