

Workshop - Initiation to Node.js

Étape 1 : Installation de Node.js

Téléchargez et installez Node.js à partir du site officiel : <https://nodejs.org/>

Vérifiez que Node.js et npm (gestionnaire de paquets Node) sont installés en ouvrant une fenêtre de terminal (ou invite de commande) et en tapant les commandes suivantes :

\$ sudo dnf install nodejs (Pour installer NodeJS)

\$ node -v (Check si nodejs c'est bien installé)

Etape 1.1 : Installation de NPM

Télécharger et installer NPM à partir de : <https://www.linuxcapable.com/how-to-install-npm-on-fedora-linux/>

\$ sudo dnf install npm (Pour installer NPM)

\$ npm -v (Check si NPM c'est bien installé)

Vous devriez voir les versions de Node.js et npm s'afficher, ce qui signifie que l'installation a réussi.

Étape 2 : Initialisation de votre projet Node.js

Créez un nouveau dossier ***“WorkshopAPI”*** pour votre projet et ouvrez un terminal dans ce dossier.

Exécutez la commande suivante pour initialiser un nouveau projet Node.js:

\$ npm init -y

Cela créera un fichier package.json avec les paramètres par défaut.

Étape 3 : Installation d'Express et des dépendances

Installez Express en exécutant la commande suivante dans votre terminal :

```
$ npm install express
```

```
$ npm install cors
```

```
$ npm install dotenv
```

```
$ npm install mysql2
```

Maintenant dans votre fichier **package.json** vous pouvez voir que les dépendances sont bien installées

Étape 4 : Configuration de la base de données avec Xampp

Assurez-vous que Xampp est installé localement. Si ce n'est pas le cas, suivez les instructions d'installation

Installation de XAMPP :

```
$ sudo dnf -y update && sudo dnf -y install libnsl
```

```
$ sudo dnf -y install wget
```

```
$ wget https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/8.2.12/xampp-linux-x64-8.2.12-0-installer.run
```

```
$ chmod a+x xampp-linux-x64-*-installer.run
```

```
$ sudo ./xampp-linux-x64-*-installer.run
```

```
$ sudo /opt/lampp/lampp start
```

Pour plus d'information : <https://computingforgeeks.com/how-to-install-xampp-on-fedora/>

Ajout de la base de donnée dans XAMPP:

Aller dans la rubrique **Importer dans XAMPP** et sélectionner le **workshop.sql** disponible sur github et cliquer sur IMPORTER

Étape 5 : Création de l'API Node.js

Créez un fichier .js dans votre dossier de projet et commencez par importer les modules nécessaires :

```
const express = require('express');
```

```
const cors = require("cors")
```

Configurez votre application Express :

```
const app = express();
```

```
const PORT= 3000; // Port d'écoute de votre application
```

Initialisation du projet pour gérer les données JSON et les données de formulaire :

```
app.use(cors( {origin: "*" } ))
```

```
app.use(express.json())
```

```
app.use(express.urlencoded( { extended: false } ))
```

Créez des routes pour l'inscription et la connexion des utilisateurs :

```
app.post('/register', async (req, res) => {
```

```
//Code pour register un utilisateur dans la base de donnée
```

```
res.status(200).json({ msg: `Register route correctly call` });
```

```
});
```

```
app.post('/login', async (req, res) => {
```

```
//Code pour login un utilisateur
```

```
res.status(200).json({ msg: `Login route correctly call` });
```

```
});
```

Démarrez le serveur Express :

```
app.listen(PORT, () => console.log("Server started on port " + PORT))
```

Étape 6 : Implémentation de l'inscription et de la connexion

Dans les routes `/register` et `/login`, utilisez le modèle `User` pour interagir avec la base de donnée. Vous pouvez utiliser **bcrypt** pour sécuriser les mots de passe.

Lors de l'inscription, insérez les informations de l'utilisateur dans la base de données après avoir hashé le mot de passe.

Lors de la connexion, vérifiez les informations de l'utilisateur par rapport à celles stockées dans la base de données.

Étape 7 : Testez votre API

Utilisez un outil comme **Postman** (`curl -o- "https://dl-cli.pstmn.io/install/linux64.sh" | sh`) pour tester votre API en effectuant des requêtes POST vers les endpoints `/register` et `/login`.

Assurez-vous que les enregistrements d'utilisateurs sont stockés dans la base de données et que les utilisateurs peuvent se connecter avec succès.