

# TP N 1

## Introduire ExpressJS et NodeJS

**ISTA:** tinghir

**Realisé par:** Groupe 1

**le:** 9 avril 2023

**Module:** M206

Création d'une application Cloud native

### Partie QCM

**Instructions :** Choisissez la bonne réponse pour chaque question.

**Qu'est-ce qu'une API REST ?**

1. Une API qui utilise les verbes HTTP pour communiquer avec les clients
2. Une API qui utilise des connexions TCP pour communiquer avec les clients
3. Une API qui utilise des connexions UDP pour communiquer avec les clients

**Que signifie l'acronyme NodeJS ?**

1. New Open Development Environment JavaScript
2. Node Open Development Environment JavaScript
3. Node JavaScript

**Comment installer Node.js ?**

1. En utilisant un navigateur web
2. En téléchargeant et en installant un fichier d'installation depuis le site officiel de Node.js
3. En utilisant un gestionnaire de paquets comme npm

**Quel est le module npm utilisé pour créer un serveur web avec Node.js ?**

1. http
2. express
3. nodemon

**Comment redémarrer automatiquement le serveur Node.js lorsqu'un fichier est modifié ?**

1. En utilisant la commande npm run start
2. En installant le module nodemon et en utilisant la commande nodemon server.js
3. En utilisant la commande node server.js à chaque fois que les fichiers sont modifiés

### **Comment exporter une fonction ou un objet depuis un module Node.js ?**

1. En utilisant la syntaxe `module.exports = maFonction;`
2. En utilisant la syntaxe `exports = maFonction;`
3. En utilisant la syntaxe `export maFonction;`

### **Comment importer une fonction ou un objet depuis un module Node.js ?**

1. En utilisant la syntaxe `require('./monModule');`
2. En utilisant la syntaxe `import maFonction from './monModule';`
3. En utilisant la syntaxe `require('./monModule').maFonction;`

### **Comment gérer les fonctions asynchrones avec JavaScript ?**

1. En utilisant des callbacks
2. En utilisant des promesses
3. En utilisant des fonctions `async/await`

### **Comment gérer les erreurs dans une application Node.js ?**

1. En utilisant un bloc `try-catch`
2. En utilisant un middleware qui renvoie une erreur 500 pour toutes les erreurs
3. En utilisant un middleware spécifique pour chaque type d'erreur

## **Partie Pratique**

### **Tache 1**

Installer NodeJS sur votre ordinateur.

### **Tache 2**

Installer NPM (Node Package Manager) sur votre ordinateur.

### **Tache 3**

Installer Express via NPM.

### **Tache 4**

Installer Nodemon via NPM pour un développement plus efficace

### **Tache 5**

Configurer un environnement de développement pour NodeJS en utilisant un éditeur de texte tel que VS Code

## **Tache 6**

Créer un fichier `index.js` pour initialiser le serveur Express

## **Tache 7**

Créer un exemple d'API REST simple en utilisant NodeJS et Express