INTRODUIRE EXPRESSJS ET NODEJS

RÉALISÉ PAR :

Chbali chayma

Kadri ayoub

Tizgui Anouar

Arbani Salah Eddine

Hatouchi Abdellah

Plan

01. Rappeler le concept des APIS REST et les méthodes du protocole http

02. définir l'écosystéme NODEJS

03 Configurer L'envirenement de Develepoment

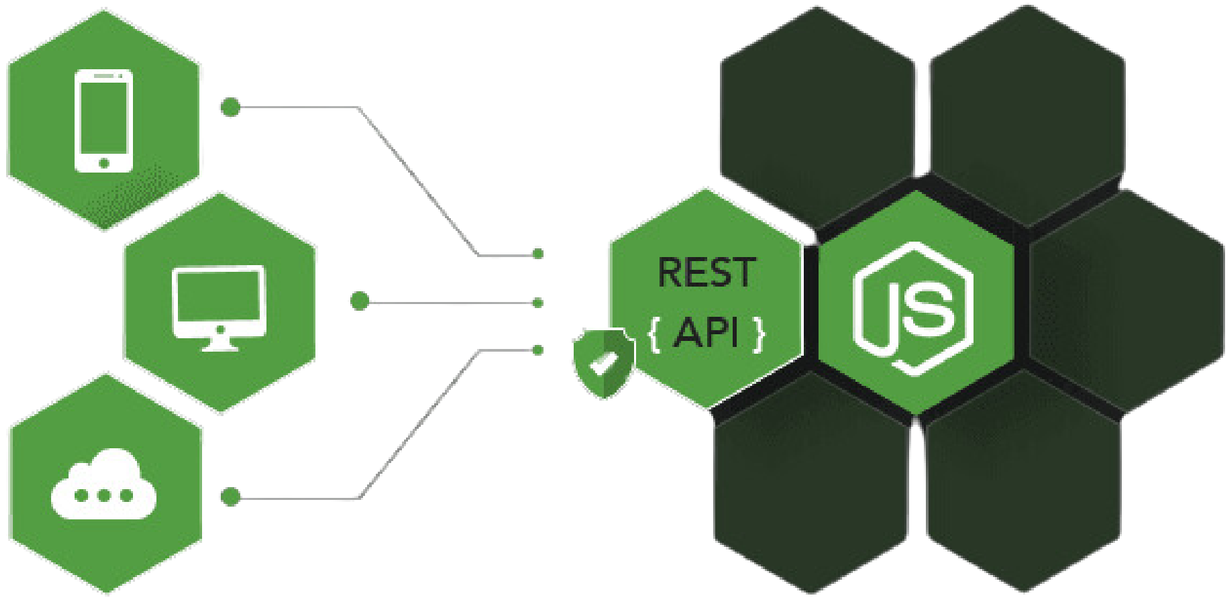
04. Recenser L'essentiel de NodeJs

Rappeler le consept des *APIS REST*

Les APIs REST (Representational State Transfer) sont un style d'architecture pour les services Web qui utilisent les protocoles HTTP et les formats de données tels que JSON et XML pour permettre aux applications de communiquer entre elles via des interfaces standardisées

Les API REST utilisent des méthodes HTTP telles que GET, POST, PUT et DELETE pour effectuer des opérations sur des ressources représentées sous forme de ressources web identifiées par des URIs (Uniform Resource Identifiers). Les clients peuvent demander des ressources en envoyant des requêtes HTTP aux serveurs, qui répondent avec des réponses HTTP contenant des données au format JSON, XML ou un autre format de données.

Les API REST sont largement utilisées dans le développement de services Web pour permettre l'interaction entre différentes applications et services. Elles sont souvent utilisées dans les applications mobiles, les sites web et les applications d'entreprise pour fournir des fonctionnalités telles que l'authentification des utilisateurs, l'accès aux données et la gestion des ressources. Les API REST sont également considérées comme un choix populaire pour les microservices, car elles peuvent être déployées de manière indépendante et facilement intégrées à d'autres services.



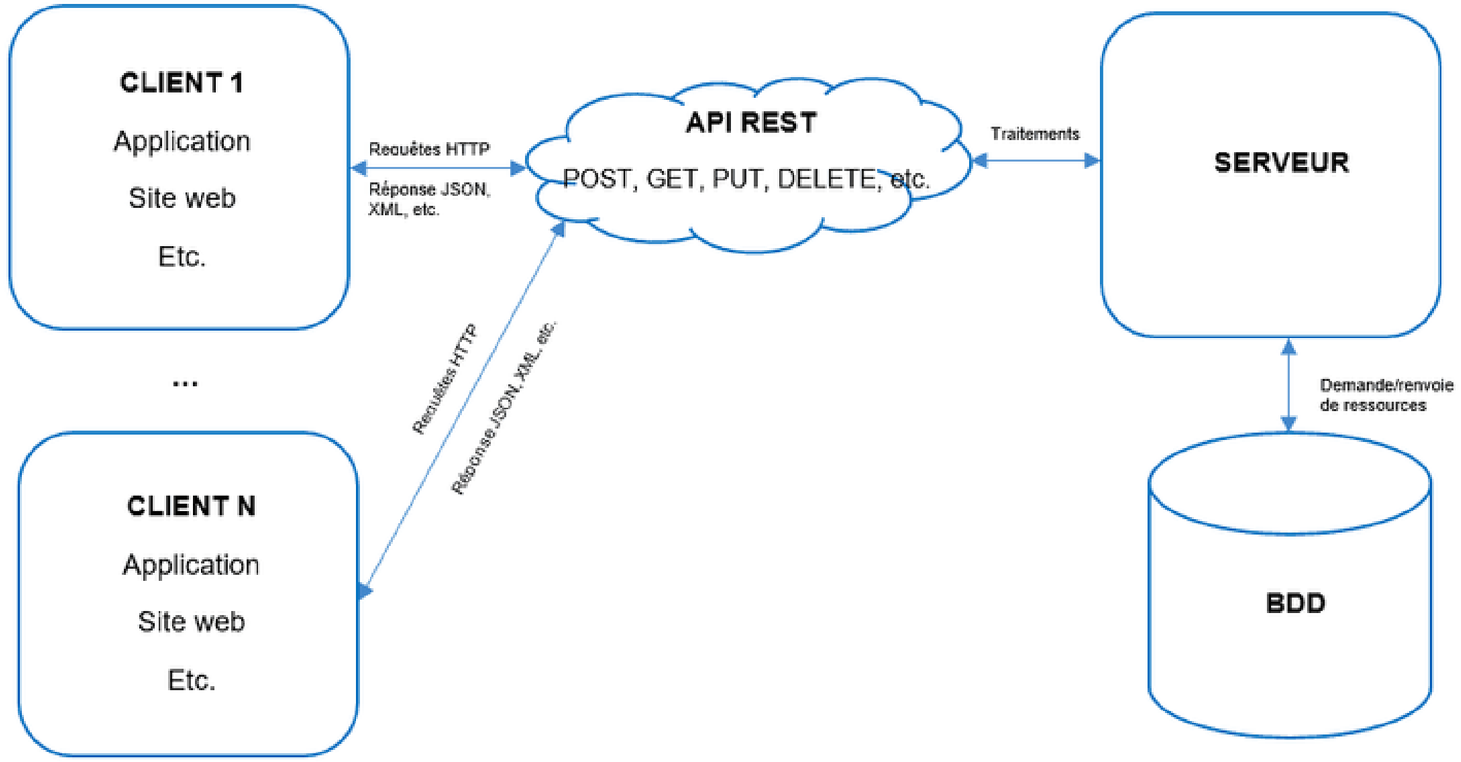
principales méthodes HTTP utilisées dans la création d'API REST

Les méthodes HTTP sont des actions que les clients et les serveurs peuvent utiliser pour communiquer entre eux via le protocole HTTP (Hypertext Transfer Protocol.
voici une brève explication des principales méthodes HTTP utilisées dans la création d'API REST :

- **GET** : La méthode GET est utilisée pour récupérer des informations auprès d'un serveur. Elle est généralement utilisée pour récupérer des ressources telles que des pages web, des images ou des fichiers.
- **POST** : La méthode POST est utilisée pour envoyer des données à un serveur. Elle est généralement utilisée pour soumettre des formulaires web ou pour ajouter des données à une base de données.
- **PUT** : La méthode PUT est utilisée pour mettre à jour des données existantes sur un serveur. Elle est généralement utilisée pour modifier une ressource existante telle qu'un fichier ou une entrée dans une base de données.

principales méthodes HTTP utilisées dans la création d'API REST

- **DELETE** : La méthode DELETE est utilisée pour supprimer des données sur un serveur. Elle est généralement utilisée pour supprimer des ressources telles que des fichiers ou des entrées dans une base de données.
- **PATCH** : La méthode PATCH est utilisée pour mettre à jour une partie des données existantes sur un serveur. Elle est généralement utilisée pour apporter des modifications mineures à une ressource existante.



Définir l'écosystéme NODEJS

Node.js est un environnement d'exécution JavaScript côté serveur qui permet aux développeurs de créer des applications réseau évolutives et hautes performances. Il utilise le moteur JavaScript V8 de Google pour exécuter du code JavaScript hors navigateur.

Node.js a été initialement développé en 2009 par Ryan Dahl et est devenu très populaire parmi les développeurs en raison de sa capacité à gérer de grandes quantités de connexions simultanées avec une utilisation minimale des ressources système

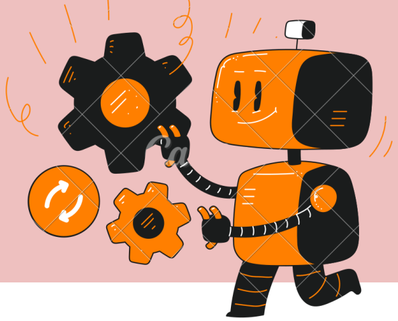
Node.js est souvent utilisé pour créer des applications web, des serveurs d'API, des outils en ligne de commande, des applications de streaming en temps réel et bien plus encore. Sa grande communauté de développeurs et de bibliothèques de modules en font un choix populaire pour les projets de développement d'applications modernes.

définir l'écosystéme NODEJS

une fonctionnalité clé de Node.js

Node.js utilise un modèle événementiel non bloquant pour les entrées-sorties (E/S), ce qui signifie qu'il peut traiter de nombreuses connexions simultanément sans bloquer le thread principal, ce qui le rend très efficace pour les applications en temps réel. Il est également doté d'un grand écosystème de modules open-source, appelé npm, qui permet aux développeurs d'ajouter rapidement des fonctionnalités à leurs applications.

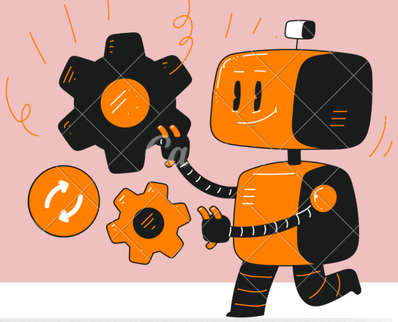
Configurer l'environnement de développement



étapes à suivre pour configurer l'environnement de développement et installer Node.js et Express.js :

1. Téléchargez la dernière version de Node.js à partir du site officiel de Node.js : https://nodejs.org/
2. Suivez les instructions d'installation pour votre système d'exploitation.
3. Ouvrez un terminal ou une ligne de commande et exécutez la commande "node -v" pour vérifier que Node.js est installé correctement. Cette commande affichera la version de Node.js que vous avez installée.

Configurer l'environnement de développement



1- initialisation de projet :

- Créer un nouveau répertoire de projet .
- Initialiser le projet .

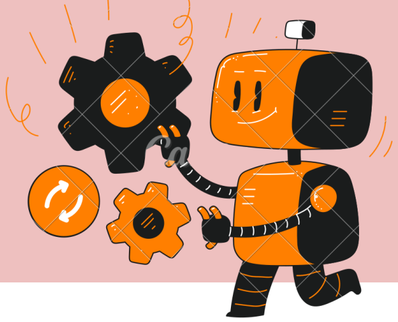
```
Express — -bash — 83×24
[salahs-MacBook-Pro:BACK-END salah$ cd Express/
[salahs-MacBook-Pro:Express salah$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
[package name: (express)
[version: (1.0.0)
```

2- Installer Express.js :

- Exécutez la commande "npm install express"

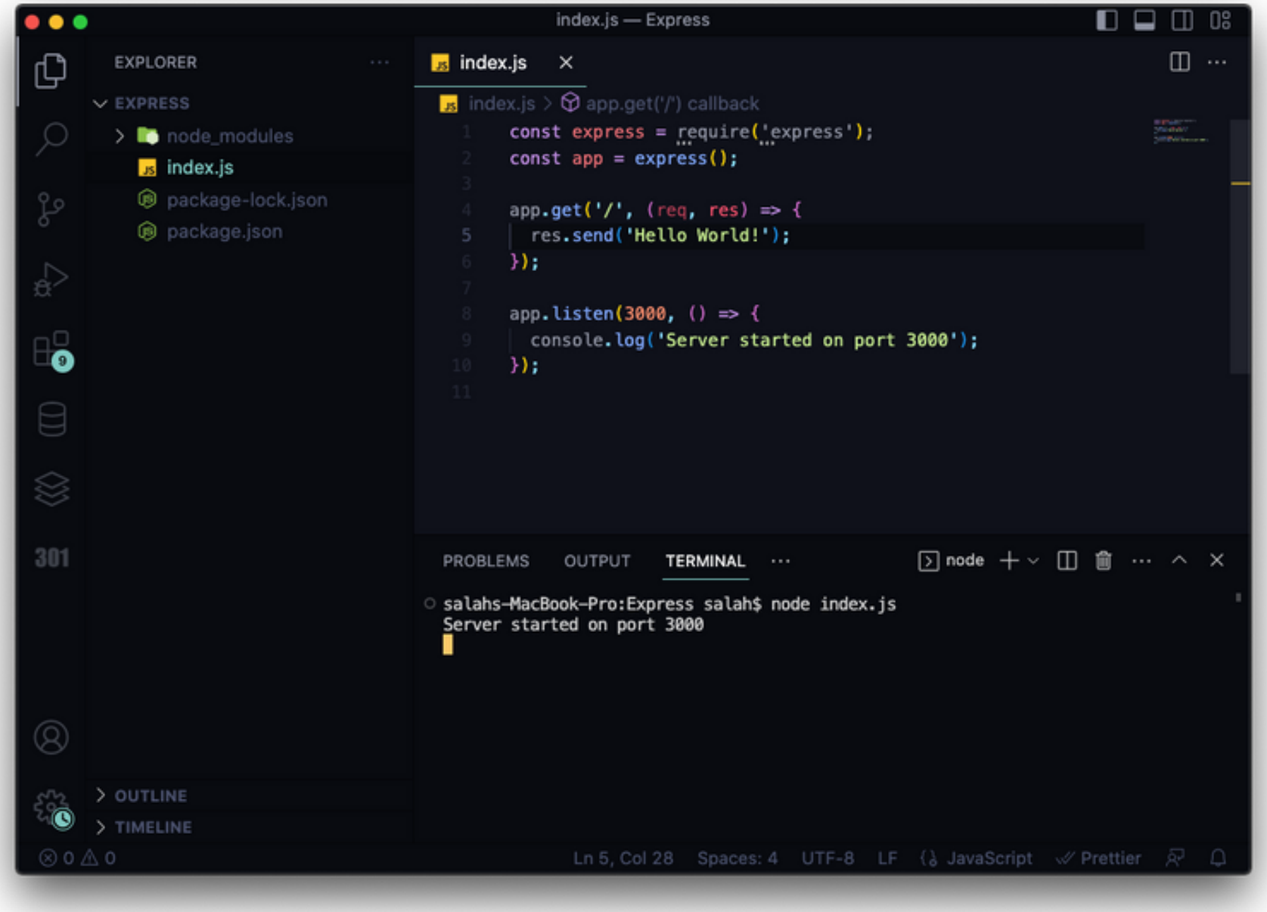
```
Express — -bash — 83×10
npm notice
[salahs-MacBook-Pro:Express salah$ npm install express

added 57 packages, and audited 58 packages in 12s

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

found 0 vulnerabilities
salahs-MacBook-Pro:Express salah$
```

Configurer l'environnement de développement

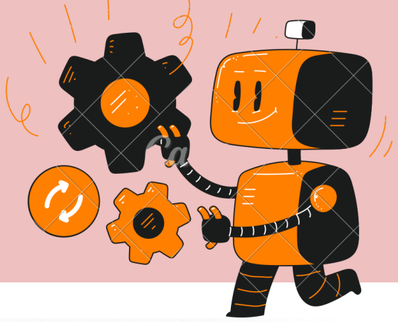


3- simple test :

- Créez un nouveau fichier appelé **index.js** dans votre répertoire de projet
- ajoutez le code suivant (image ->)
- Exécutez la commande "node index.js"(terminal)

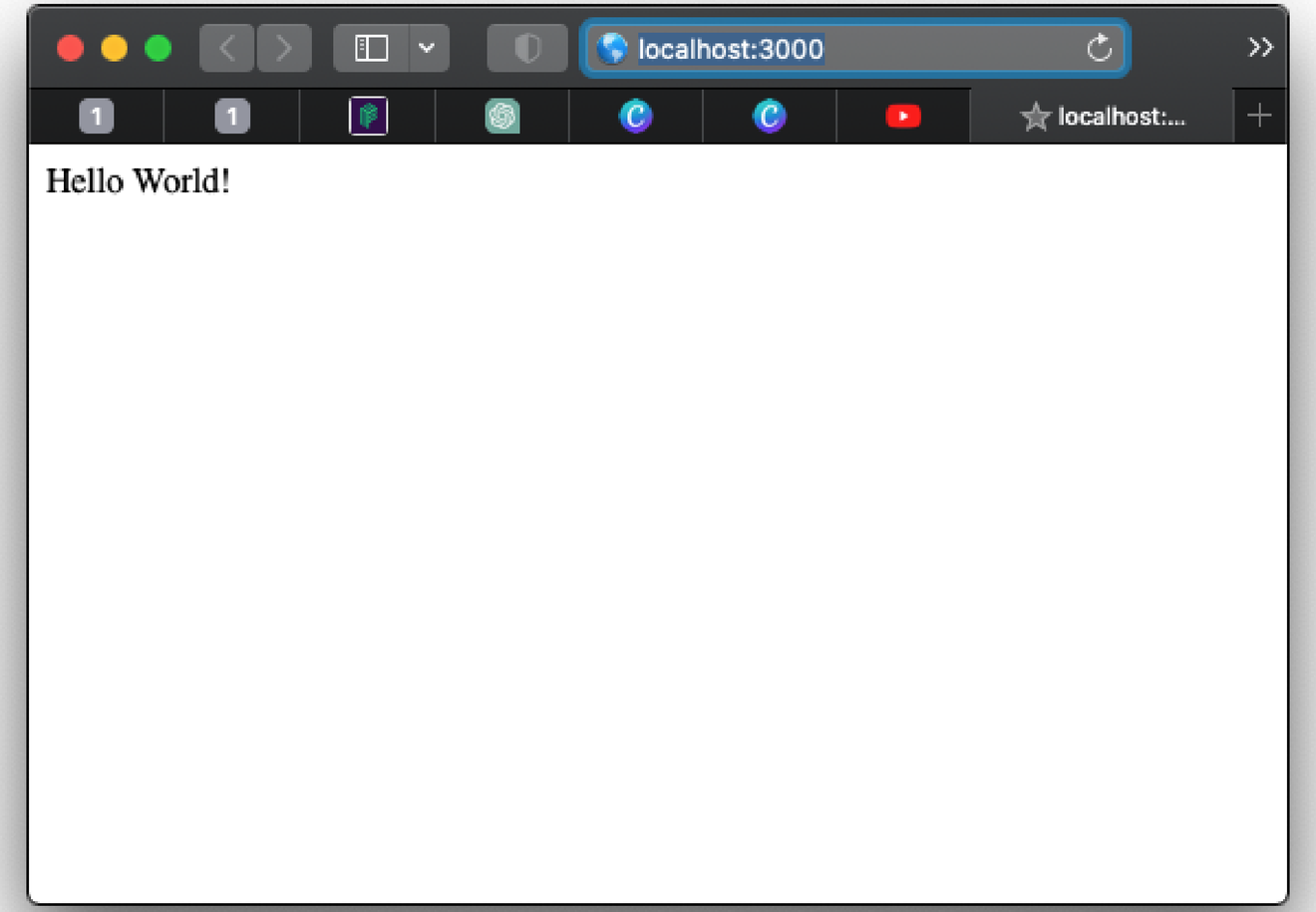


Configurer l'environnement de développement

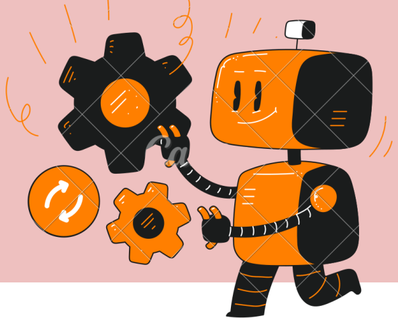


4- Tester le serveur :

- Ouvrez votre navigateur Web et accédez à **http://localhost:3000/**



Configurer l'environnement de développement



5- installer Nodemon (npm)

- exécutant la commande suivante dans votre terminal :

"npm install -g nodemon"

- exécutant la commande suivante dans votre terminal :

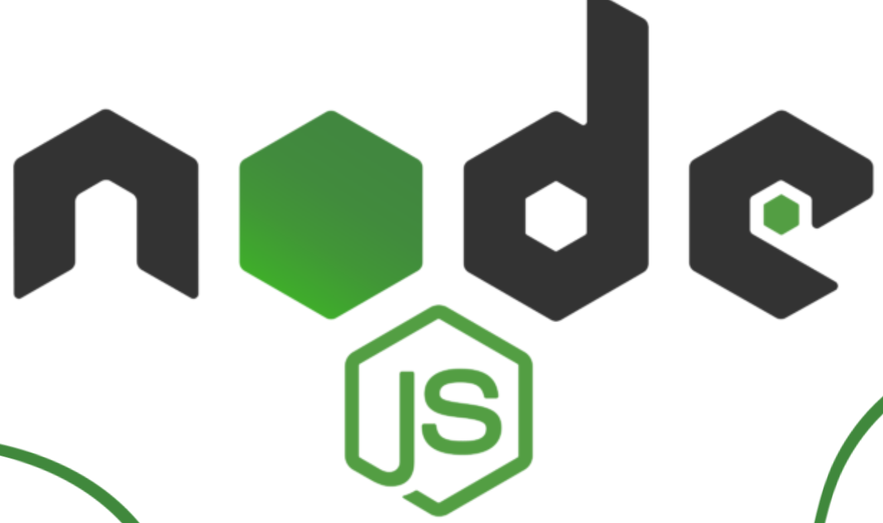
"nodemon app.js"

```
npm install -g nodemon
```

```
nodemon app.js
```

Recenser L'essentiel de NodeJs

Cette partie est pratique



Merci pour votre Attention