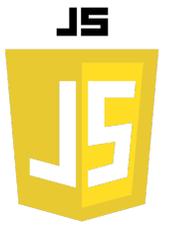




# CONCEVOIR UN API AVEC NODEJS



## Étape 1 : Installation de Node.js et NPM

Avant de commencer, assurez-vous d'avoir Node.js et NPM (Node Package Manager) installés sur votre machine. Dans le cas contraire, il faudra télécharger et installer nodejs et npm.

## Étape 2 : Création d'un nouveau projet

Créez un nouveau dossier pour votre projet et naviguez vers ce dossier dans votre terminal ou invite de commande. Vous pouvez le faire avec les commandes suivantes :

```
mkdir mon-projet-node  
cd mon-projet-node
```

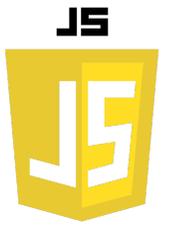
## Étape 3 : Initialisation du projet

Initialisez votre projet Node.js avec la commande suivante. Cela créera un fichier package.json qui contiendra les informations sur votre projet et les dépendances.

```
npm init -y
```



# CONCEVOIR UN API À PARTIR D'EXPRESS



## Étape 4 : Installation d'Express.js

Installez Express.js, un framework web pour Node.js, avec la commande suivante :

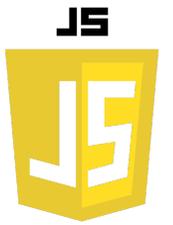
```
npm install express
```

## Étape 5 : Création de votre application

Créez un nouveau fichier appelé app.js dans votre dossier de projet. Dans ce fichier, nous allons créer une application web simple avec Express.js. Vous pouvez utilisé l'IDE de votre choix.



# CONCEVOIR UN API À PARTIR D'EXPRESS



```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Bonjour, bienvenue sur mon application Node.js !');
});

app.listen(port, () => {
  console.log(`L'application est en écoute sur le port ${port}`);
});
```

## Étape 6 : Exécution de votre application

Vous pouvez maintenant exécuter votre application avec la commande suivante :

```
node app.js
```

Ouvrez un navigateur web et allez à l'adresse **http://localhost:3000**. Vous devriez voir le message "Bonjour, bienvenue sur mon application Node.js !".



# CONCEVOIR UN API À PARTIR D'EXPRESS



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

Cette ligne importe le module Express dans votre application. Express est un framework pour Node.js qui fournit des fonctionnalités utiles pour développer des applications web.

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

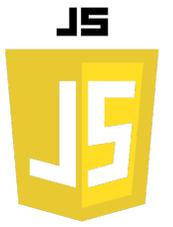
Ici, nous initialisons une nouvelle application Express. L'objet app a des méthodes pour les routes et les performances HTTP (GET, POST, PUT, DELETE, etc.), pour spécifier le moteur de rendu des vues, et pour spécifier où se trouvent les vues.

```
const port = 3000;
```

Nous définissons le port sur lequel notre application sera exécutée. Vous pouvez choisir n'importe quel numéro de port qui n'est pas déjà utilisé par une autre application.



# CONCEVOIR UN API À PARTIR D'EXPRESS



```
app.get('/', (req, res) => {  
  res.send('Bienvenue sur mon API !');  
});
```

Ici, nous définissons une route pour l'URL de base ("/") de notre application. Lorsqu'un client envoie une requête HTTP GET à cette URL, la fonction de rappel que nous fournissons est appelée. Cette fonction prend deux arguments : un objet req (requête) qui contient des informations sur la requête HTTP, et un objet res (réponse) qui est utilisé pour définir ce que nous renvoyons au client. Dans ce cas, nous renvoyons simplement le texte "Bienvenue sur mon API !".

```
app.listen(port, () => {  
  console.log(`L'application est en écoute sur le port ${port}`);  
});
```

Enfin, nous disons à notre application de commencer à écouter les requêtes HTTP sur le port que nous avons défini. La fonction de rappel est appelée une fois que l'application a commencé à écouter sur ce port. Dans ce cas, nous écrivons simplement un message dans la console pour nous informer que l'application est en cours d'exécution.

## Diagramme de séquence de l'API :

