Présentation Node.js

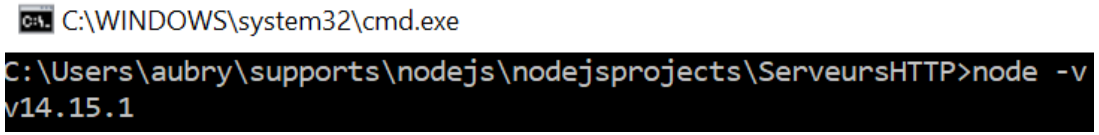
08/12/2020

Introduction

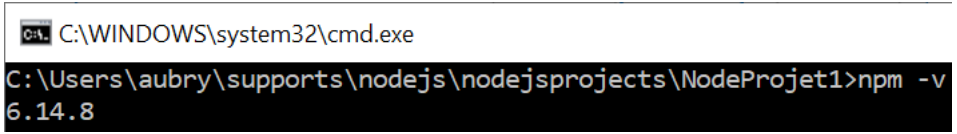
- Node.js est un ensemble de logiciels qui va permettre d'exécuter du code JavaScript côté serveur.
- Node.js a été créé par Ryan Dahl en 2009.
- Node.js est écrit et s'utilise en JavaScript.
- Permet l'exécution de tâches asynchrones.
- Permet de créer un serveur HTTP.

Installation de Node.js

- Télécharger Node.js sur nodejs.org, extraire le fichier .msi, terminer l'installation.
- A la fin de l'installation, on tape « node -v » dans l'invite de commandes (touche Windows+R, puis cmd) pour vérifier la version de Node.js installée.



- On peut taper « npm -v » pour connaître la version de npm (Node.js Package Manager, gestionnaire de paquets de Node.js).



Exemple d’exécution asynchrone

exemple01.js > ...

```
1   setTimeout(function() {
2   console.log('Au revoir');
3   }, 5000);
4   console.log('Bonjour');
```

Dans ce code *exemple01.js*, on utilise la méthode setTimeout() qui prend en argument une fonction anonyme et le délai en millisecondes.

On exécute le code en tapant : « *node exemple01* » dans le répertoire qui le contient.
Lors de l’exécution du programme, « Bonjour » s’affiche en premier suivi de « Au revoir ».

C:\WINDOWS\system32\cmd.exe

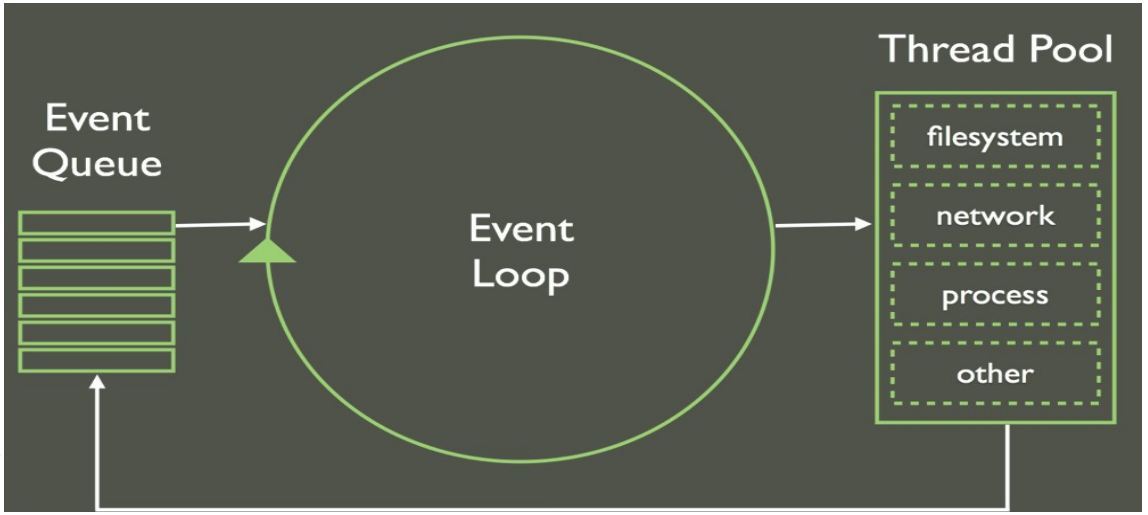
```
C:\Users\aubry\supports\nodejs\nodejsprojects\NodeProjet1>node exemple01
Bonjour
Au revoir
```

La méthode setTimeout() permet d'exécuter du code de manière asynchrone.
Cette méthode n’est pas bloquante.

Boucle des événements

L'exécution du code déclenche une « file d'attente » d'événements.

Le système prend les événements de la « file d'attente » dans la boucle des événements et les traite.



Si le système ne peut pas traiter l'événement, il l'envoie vers le Thread Pool pour être traité

Après cela, la suite du code est ajoutée à la « file d'attente »

Le code n'est pas exécuté en parallèle car il est mis en file d'attente, mais il ne bloque pas l'exécution du code depuis lequel il a été appelé.

Exemple de serveur

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 8081;

http.createServer(function (requete, reponse) {
    reponse.writeHead(200, {'Content-Type': 'text/html; charset=UTF-8'});
    reponse.write('Bonjour');
    reponse.end();
}).listen(port, hostname);
```

On charge la bibliothèque '**http**' qui permet de créer un serveur web avec la méthode require().
On déclare les constantes **hostname** et **port**.

On crée un serveur avec **http.createServer()** qui écoute à l'adresse 127.0.0.1 sur le port 8081.
Lorsque le serveur est démarré, on exécute la fonction anonyme.
La fonction anonyme prend une **requête** et une **réponse** en argument et va écrire la réponse.

```
C:\WINDOWS\system32\cmd.exe - node  exempleServeurHTTP01
C:\Users\aubry\supports\nodejs\nodejsprojects\ServeursHTTP>node exempleServeurHTTP01
```

On lance le serveur avec la commande :
node exempleServeurHTTP01

Résultat dans un navigateur :