Notes : Api avec Node JS

# Qu’est-ce que Node.js ?

Runtime qui permet de créer toutes sortes d’applications/outils côté serveur, en JavaScript.

# Installation de Node JS

Téléchargement puis installation de la dernière version de Node depuis NodeJS.org (permet également d’installer npm pour l’installation des packages pour les projets)

# Les « core modules »

C’est une liste des modules de base fournis par Node.js :

* http : lancer un serveur web, envoyer des requêtes
* https : lancer un serveur SSL
* fs : fournit des méthodes pour interagir avec le système de fichiers
* path : fournit des fonctions pour interagir avec les chemins de fichiers
* os : fournit des méthodes et des propriétés pour interagir avec le système d’exploitation

# Créer un serveur web

Dans cette partie, nous allons créer un simple serveur web avec NodeJS (code démo)

Voir le cours NODEJSAPI/190722/app.js

# Le cycle de vie d’un programme Node

Cycle de vie = boucle d’évènements :

* permet d’effectuer plusieurs tâches
* permet de rendre les tâches plus rapides
* permet à Node.js d’effectuer des opérations d’E/S non bloquantes

Cycle de vie d’un programme Node :

1. exécution d’un fichier de nœud à l’aide de node app.js
2. exécution du script
3. analyse du script par l’analyseur en langage machine (toutes les fonctions/variables vont être enregistrées dans un emplacement mémoire)
4. exécution du non infini par le programme
5. exécution de la boucle d’événements

# L’objet Request

Headers : métadonnées ajoutées à une requête http

L’objet request représente la requête http et possède des propriétés pour la chaîne de requête (paramètres, corps, en-têtes http, etc).

# L’objet Response

L’objet response représente la réponse http qu’une application Node JS envoie lorsqu’elle reçoit une requête http.

Méthode d’objet de response :

* res.end(); : méthode utilisée pour terminer le processus de réponse

# Router la http

Le routage peut être effectué en important le module d’URL intégré et en analysant l’URL de la demande. Pour vérifier si c’est correct, on peut comparer l’URL de la requête à l’URL de la route.

Voir le cours NODEJSAPI/190722/app.js

On peut définir des en-têtes sur la réponse en utilisant res.setHeader()

# Comment récupérer les données des requêtes entrantes ?

Exercice 1 : Stocker contenu message dans fichier texte

Voir le cours NODEJSAPI/190722/app.js