AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 16/05/2023

S'inscrire

Formation Deno

3 jours (21 heures)

Présentation

Deno est un runtime en ligne de commande pour exécuter du code JavaScript et Typescript. Comme NodeJS, Deno est construit sur le moteur JavaScript de Chrome : V8. Il est écrit entièrement en Rust (Nodejs est écrit en C et C++) ce qui promet pas mal de performances. Enfin pour garantir de l'I/O asynchrone l'event loop utilisée est Tokio. C'est celle de Rust, l'équivalent de libuv en NodeJS.

Deno existe depuis bientôt 1 an. Il est en développement pour le moins intensif. Il a pour objectif de devenir un environnement de développement productif et sécurisé en JavaScript avec un support natif de Typescript. Ca ressemble énormément à NodeJS dans l'utilisation. Un NodeJS réécrit en entier avec des technologies moins contraignantes. De base il évite tous les problèmes qu'on ne peut plus résoudre avec NodeJS aujourd'hui. Et bien sûr il y a son lot de nouveautés.

Comme toujours, nous vous enseignerons la dernière version en date du programme à savoir Deno 1.11.

Objectifs

- Savoir utiliser Deno et NodeJS dans les dernières versions
- Développer une application web avec Deno

Public visé

- Développeurs web

Pré-requis

- Connaissance du langage JavaScript et connaissance d'un framework côté client seraient un plus ou connaissance d'un autre langage de programmation orienté objet (java, php, etc.)

Pour aller plus loin

- Suivez notre Formation Node.JS

Programme de la formation Deno

- Introduction
- Chapitre Philosophy
  - Goals
  - Non-goals

- Chapitre Setup
  - Binary Install
  - Build from source
    - Prerequisites
    - Other useful commands

- Chapitre API reference
  - deno types
  - Reference websites

- Chapitre Examples
  - An implementation of the unix "cat" program
  - TCP echo server
  - Inspecting and revoking permissions
  - File server
  - Permissions whitelist
  - Run subprocess
  - Linking to third party code
  - Using external type definitions
  - Testing if current file is the main program

- Chapitre Command line interface
  - Flags
  - Environmental variables
  - Shell completion
  - V8 flags
  - Bundling
  - Installing executable scripts

- Chapitre Proxies
- Chapitre Import maps

- Chapitre Internal details
  - Deno and Linux analogy
    - Resources
    - Metrics
  - Schematic diagram
  - Profiling
  - Debugging with LLDB
  - Deno Core
  - Updating prebuilt binaries
  - Continuous Benchmarks
  - Logos

- Chapitre Contributing
  - Submitting a pull request
  - Changes to third_party
  - Adding Ops (aka bindings)
  - Documenting APIs
    - Utilize JSDoc

Comparaison avec Node.js

Deno et Node.js sont tous deux des environnements d'exécution construits sur le moteur JavaScript V8 de Google, identique à celui utilisé dans Google Chrome. Ils disposent tous deux de boucles d'événements internes et fournissent des interfaces de ligne de commande pour l'exécution de scripts et un large éventail d'utilitaires système.

En attendant, Deno s'écarte principalement de Node.js dans les aspects suivants:

- Utilise ES Module comme système de module par défaut, au lieu de CommonJS.
- Utilise des URL pour le chargement de dépendances locales ou distantes, similaires aux navigateurs.
- Inclut un gestionnaire de paquets intégré pour la récupération des ressources, donc pas besoin de NPM.
- Prend en charge TypeScript prêt à l'emploi, à l'aide d'un compilateur TypeScript instantané doté de mécanismes de mise en cache.
- Vise une meilleure compatibilité avec les navigateurs avec une large gamme d'API Web.
- Permet de contrôler l'accès au système de fichiers et au réseau afin d'exécuter du code en bac à sable.
- API redéfinie pour utiliser les fonctionnalités Promises, ES6 et TypeScript.
- Réduit la taille de l'API principale tout en fournissant une grande bibliothèque standard sans dépendances externes.

Sociétés concernées

Cette formation s'adresse aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.