+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 WebAssembly

3 jours (21 heures)

Présentation

WebAssembly est essentiellement le successeur spirituel d'ASM.js, et est développé par Google, Microsoft, Mozilla entre autres. Ses principaux avantages sont des temps de chargement plus rapides pour les utilisateurs et la compatibilité du code (WebAssembly prendra en charge les anciennes plates-formes en traduisant le binaire wasm en code asm.js.

WebAssembly représente une avancée fondamentale de la plateforme web. Il permet d'exécuter du code de langages de haut niveau comme C/C++/Rust sur le Web avec des performances similaires aux applications natives.

WebAssembly est conçu pour être utilisé de pair avec JavaScript. Grâce à l'API JavaScript WebAssembly, on peut charger des modules WebAssembly au sein d'une application JavaScript et partager des fonctionnalités entre les deux. Cela permet de tirer parti des performances de WebAssembly et de la flexibilité de JavaScript, même si on ne sait pas écrire du code WebAssembly.

Ce cours vous montrera comment utiliser cette technologie pour écrire des applications de haute performance qui s'exécutent dans le navigateur.

Vous serez initié à de puissants concepts de WebAssembly qui vous aideront à écrire des applications Web légères et puissantes avec des performances natives. Apprendre WASM commence par vous familiariser avec l'évolution de la programmation Web et ce qui peut être fait avec cet outil. Vous verrez ensuite comment passer de JavaScript à asm.js en passant par WebAssembly.

Au fur et à mesure de votre progression, vous analyserez l'anatomie d'un module WebAssembly et la relation entre les formats binaires et texte, ainsi que l'API JavaScript correspondante. Comme toujours, nous vous enseignerons la dernière version en date à savoir WebAssembly 1.1.

Objectifs

- Explorer les éléments de WebAssembly
- Créer et charger un module WebAssembly
- Créer une application à partir de zéro
- Créer des applications web à page unique avec .Net ou C++, sans code JavaScript

Public visé

Développeur Web

Pré-requis

- Connaissance en programmation
- Connaissance de JavaScript, C/C++

Pour aller plus loin

- Découvrez le couplage de WebAssembly et du langage ultra-rapide Rust avec notre Formation WebAssembly avec Rust
- Pour compléter cette formation, nous vous proposons la formation sur Node.JS afin de mieux gérer l'intégration et la transition entre les deux frameworks.

Programme de notre Formation WebAssembly

Introduction à WebAssembly

- Qu'est-ce que WebAssembly ?
- Quels sont les problèmes qu'il permet de résoudre ?
- Comment fonctionne-t-il ?
- Structure d'un module WebAssembly
- Format texte de WebAssembly
- Comment le WebAssembly est-il sécurisé ?
- Quels langages puis-je utiliser pour créer un module WebAssembly ?
- Où puis-je utiliser mon module ?

WebAssembly Performance

- WebAssemmbly vs. JavaScript Performance
- Installation d'Emscripten toolkit (emcc)

Support du navigateur WebAssembly

- Support du navigateur actuel
- Roadmaps des navigateurs pour le support de WebAssembly
- Support de Parralel WASM/JavaScript (Pollyfill)

Module WebAssembly

- Créer son premier module
- La boîte à outils Emscripten
- Modules d'assemblage Web
- Options de sortie Emscripten
- Compiler du C ou du C++ avec Emscripten et utiliser le modèle HTML
- Générer le code de plomberie JavaScript par Emscripten
- Faire en sorte que Emscripten ne génère que le fichier WebAssembly
- Détection des caractéristiques : Comment tester si WebAssembly est disponible
- Utilisation de C ou C++ pour créer un module avec la plomberie Emscripten
- Utilisation de C ou C++ pour créer un module sans Emscripten
- Exemples de cas d'utilisation réels

Création de votre premier module WebAssembly

- La boîte à outils Emscripten
- Modules d'assemblage Web

Liaison dynamique : Mise en oeuvre

- Liaison dynamique : Avantages et inconvénients
- Options de liaison dynamique
- Examen des liens dynamiques
- Création des modules WebAssembly
- Ajustement de la page web

Threading : Web workers & pthreads

- Avantages des web workers
- Considérations relatives à l'utilisation des web workers
- Préchargement (prefetching) d'un module WebAssembly à l'aide d'un web worker
- Utilisation des pthreads

Modules WebAssembly dans Node.js

- Validation côté serveur (server-side)
- Travailler avec des modules construits en Emscripten
- Utilisation de l'API JavaScript de WebAssembly

Débogage & tests

- Format texte
- Créer la logique de base du jeu en utilisant le format texte WebAssembly
- Génération d'un module WebAssembly à partir du format texte
- Le module généré par Emscripten
- Création des fichiers HTML et JavaScript
- Visualisation des résultats

Personnalisation du Debug

- Ajuster le HTML
- Affichage du nombre d'essais
- Augmenter le nombre de tentatives
- Mise à jour de l'écran de synthèse
- Installer le cadre de test JavaScript
- Création et exécution des tests

Fonctions : ccall, cwrap, direct, Emscripten macros

- ccall, cwrap, Direct function calls
- Passage : Passing an array to a module
- emscripten_run_script macros
- EM_JS macros
- EM_ASM macros

Les modules de WebAssembly

- Déclaration des modules
- Exportations
- Index et méthode de démarrage
- Mémoire linéaire
- Insertion de code
- Information de débogage
- Intégration avec ECMAScript (ES6)
- Compiler C ou C++ avec Emscripten
- Création d'un modèle HTML
- Interop avec JavaScript
- Appeler vers JavaScript
- Appeler le module depuis JavaScript

Compilation et arbre syntaxiques abstraits (AST)

- Pourquoi les AST ?
- Syntaxe et analyse sémantique
- Propriétés et annotations
- AST Design
- Types de variables
- Composants Left et Right
- Identificateurs

- Modèles de conception utiles
- Sérialisation binaire des AST

Format texte de WebAssembly

- Affichage de la source sur un module WebAssembly
- Compilation d'S-Expression et assemblage en ligne
- Débogage de l'intégration des symboles
- Profilage

Création d'application à partir de zéro

- Conception d'une application
- Exigences via le serveur
- Interagir avec les services du serveur

Modules avancés de WebAssembly

- Intégration avec Node.JS
- Options de Threads
- Utilisation des WebWorkers
- Utilisation des PThreads
- Caractéristiques à venir

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.