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

Mis à jour le 05/12/2024

S'inscrire

Formation Rust Avancé

2 jours (14 heures)

Présentation

Notre formation Rust Avancé est conçue pour les développeurs ayant déjà une expérience de base avec Rust et cherchant à approfondir leurs connaissances pour tirer pleinement parti des capacités avancées du langage Rust.

Cette formation vous emmènera au-delà des fondamentaux pour explorer les techniques avancées, les concepts de programmation asynchrone, la concurrence, l'interopérabilité avec d'autres langages, et plus encore, vous permettant de développer des applications Rust de haute performance et sécurisées.

Vous apprendrez à utiliser à intégrer Rust grâce à l'interface de fonctions étrangères (FFI), appliquer des modèles de concurrences efficaces pour développer des applications hautement sécurisées et parallèles.

Comme dans toutes nos formations, celle-ci vous présentera la toute dernière version de Rust Programming Language, Rust 1.83.

Objectifs

- Maîtriser les concepts avancés de Rust
- Optimiser vos applications Rust pour une performance maximale
- Développer des applications web et des services avec Rust

Public visé

Développeurs.

Pré-requis

- Une compréhension solide des concepts de base de Rust
- Expérience pratique avec le langage Rust, idéalement avoir complété des projets ou notre formation Rust
- Familiarité avec les concepts de programmation système et asynchrone
- Environnement de développement Rust configuré, incluant Cargo et les outils de compilation Rust

Pré-requis logiciels

- Installer Docker et Docker Compose
- Installation de gnuplot
- Environnement de développement Rust configuré, incluant Cargo et les outils de compilation Rust

Programme de la formation Rust Avancé

TRAITS ET GÉNÉRIQUES AVANCÉS

- Approfondissement des traits et des types génériques
- Utilisation des traits comme paramètres de fonctions
- Bornes de traits et spécialisation
- Génériques et performance: monomorphisation
- Patterns de conception avec traits et génériques

PROGRAMMATION ASYNCHRONE EN RUST

- Comprendre le modèle d'exécution asynchrone
- Utiliser `async` et `await` pour des opérations non bloquantes
- Gestion des erreurs en asynchrone
- Comparaison des runtimes `tokio` et `async-std`
- Patterns de conception pour la programmation asynchrone

CONCURRENCE EN RUST

- Modèles de concurrence en Rust: threads, canaux, et `Arc<Mutex>`
- Utiliser `Rc` et `Arc` pour la gestion de la mémoire partagée
- Stratégies de passage de messages entre threads
- Sécurité des données en environnement concurrent
- Exploration des crates de concurrence populaires

UTILISATION AVANCÉE DE LIFETIMES

- Comprendre les lifetimes avancées pour la gestion de la mémoire
- Annotation des lifetimes dans les structures complexes
- Lifetimes dans les callbacks et fermetures

- Patterns pour éviter les erreurs de lifetime
- Gestion avancée des références et des emprunts

MÉMOIRE ET PERFORMANCE

- Gestion manuelle de la mémoire avec `unsafe`
- Optimisations de performance en Rust
- Profilage et débogage des applications Rust
- Utilisation des collections pour maximiser la performance
- Techniques d'allocation et libération efficaces

INTÉGRATION AVEC D'AUTRES LANGAGES

- FFI (Foreign Function Interface) pour intégrer C/C++ avec Rust
- Créer et utiliser des bibliothèques dynamiques
- Appeler Rust depuis d'autres langages
- Sécurité et pratiques recommandées dans l'utilisation de FFI

DÉVELOPPEMENT WEB AVEC RUST

- Création d'API REST avec Actix-web et Rocket
- Utilisation de Diesel pour l'accès aux bases de données
- Sécurité et gestion des sessions dans les applications web
- WebAssembly pour intégrer Rust dans le navigateur

RUST POUR LE SYSTÈME ET L'EMBARQUÉ

- Programmation `no_std` pour les systèmes embarqués
- Gestion de la mémoire sans allocation dynamique
- Interaction avec le matériel et les systèmes d'exploitation
- Crates et librairies pour le développement embarqué

VISUALISATION ET INTERFACES UTILISATEUR

- Création d'interfaces utilisateur avec `egui`, `Druid`, ou `Iced`
- Gestion des événements et du rendu graphique
- Performance et réactivité des interfaces
- Intégration avec des systèmes de fenêtrage et des outils graphiques

PROJET ET REVUE DE CODE

- Application des concepts appris à un projet concret
- Bonnes pratiques de développement en Rust
- Revue de code et feedback en groupe

- Discussion sur les cas d'usage avancés et partage d'expérience

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d’anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

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.