

Mis à jour le 04/02/2026

S'inscrire

Formation JetBrains RustRover

3 jours (21 heures)

Présentation

JetBrains RustRover vous aide à coder en Rust plus vite et plus sereinement grâce à l'analyse du code, la navigation intelligente et le débogage intégré. Cette formation cible des cas d'usage concrets : démarrer un projet Cargo, refactorer en sécurité, diagnostiquer des erreurs de borrow checker et industrialiser votre workflow.

L'objectif de notre formation est de vous rendre autonome sur RustRover pour développer, tester et maintenir des applications Rust avec un environnement outillé : édition assistée, exécution, tests, profiling et intégration Git. Vous apprendrez à configurer l'IDE pour votre contexte (workspace, toolchain, clippy, rustfmt) et à exploiter les inspections pour améliorer la qualité.

L'approche est résolument pratique : ateliers guidés, démos pas à pas et exercices de correction.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Configurer RustRover, Cargo et la toolchain Rust.
- Naviguer efficacement dans un codebase (symboles, usages, types).
- Exécuter et déboguer binaires et tests avec breakpoints.
- Appliquer rustfmt, clippy et inspections pour refactorer.
- Automatiser le workflow via run configurations et Git.

Public visé

- Développeurs Rust débutants à intermédiaires.
- Développeurs système/embarqué souhaitant outiller leur IDE.
- Développeurs back-end adoptant Rust en production.

Pré-requis

- Bases de programmation (types, fonctions, structures).
- Notions de Rust (ownership/borrow, modules, Cargo).
- Compréhension des tests unitaires.
- Pratique minimale de Git.

Pré-requis techniques

- 8 Go RAM minimum, 16 Go recommandés.
- Windows, macOS ou Linux (x64/ARM selon machine).
- Rust stable installé (rustup), Cargo, rustfmt, clippy.
- JetBrains RustRover installé, accès terminal et Git.

Programme de formation JetBrains RustRover

[Jour 1 - Matin]

Prise en main de RustRover et configuration d'un environnement Rust

- Installer RustRover, Rust toolchain (rustup) et composants (clippy, rustfmt)
- Configurer un projet Cargo : workspace, profils, variables d'environnement
- Tour d'horizon de l'IDE : Project, Editor, Run, Terminal, Search Everywhere
- Paramétrer l'analyse Rust : rust-analyzer, inspections, quick-fixes
- Atelier pratique : Créer un workspace Cargo et valider la chaîne outillage (build, fmt, clippy).

[Jour 1 - Après-midi]

Productivité au quotidien : navigation, refactoring et exécution

- Navigation efficace : symboles, usages, hiérarchie, bookmarks, structure de fichier
- Refactorings utiles : rename, extract fonction, inline, move, change signature
- Exécuter et déboguer : configurations Run/Debug, breakpoints, watch, evaluate
- Gestion des tests : exécution ciblée, filtres, couverture, logs et output
- Atelier pratique : Refactorer un module, ajouter des tests et déboguer un cas d'échec.

[Jour 2 - Matin]

Qualité de code : inspections, formatage, linting et documentation

- Mettre en place rustfmt : règles, format on save, actions de formatage
- Exploiter clippy : niveaux de lint, corrections rapides, suppression/allow ciblé
- Comprendre les inspections RustRover : erreurs, warnings, intentions et quick-fixes
- Générer et consulter la documentation : rustdoc, commentaires doc, navigation API
- Atelier pratique : Nettoyer un codebase (fmt + clippy), documenter une API et valider en CI locale.

[Jour 2 - Après-midi]

Gestion de dépendances, build avancé et intégration Git

- Maîtriser Cargo.toml : features, optional deps, dev-dependencies, patch
- Profiler le build : profils debug/release, incremental, target dir, cache
- Outils Cargo dans l'IDE : run configurations, tasks, commandes fréquentes
- Workflow Git : commit, amend, rebase interactif, résolution de conflits dans l'IDE
- Atelier pratique : Ajouter une dépendance avec features, optimiser un build et gérer une PR avec conflits.

[Jour 3 - Matin]

Débugage avancé, tests et diagnostics de performance

- Débugage avancé : breakpoints conditionnels, logpoints, attach à un process
- Tests Rust : unit/integration, tests paramétrés, exécution parallèle et filtrage
- Analyse des erreurs : backtraces, panic, affichage structuré et logs
- Premiers pas perf : profils, bench, repérage des hotspots (outils Cargo)
- Atelier pratique : Diagnostiquer un bug intermittent avec breakpoints conditionnels et stabiliser via tests.

[Jour 3 - Après-midi]

Industrialisation : templates, CI locale et bonnes pratiques d'équipe

- Standardiser un projet : templates Cargo, structure de crate, conventions de modules
- Automatiser la qualité : hooks, tâches (fmt/clippy/test), checks avant commit
- Préparer une CI "portable" : scripts, commandes reproductibles, variables d'environnement
- Bonnes pratiques IDE en équipe : settings partagés, exclusions, inspections recommandées
- Atelier pratique : Construire un squelette de projet prêt CI (fmt + clippy + tests) et le partager au groupe.

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.