cargo-fuzz

cargo-fuzz

> *Catalogue généré le 2026-05-11*

En une phrase

cargo-fuzz est l'outil officiel pour fuzzer un projet Rust : il bombarde tes fonctions Rust avec des entrées aléatoires pour trouver les bugs cachés, en une seule commande Cargo.

Quand l'utiliser

- Fuzzer un projet Rust qui utilise Cargo (l'outil de build standard).
- Tester un crate Rust avant publication sur crates.io.
- Tester du code Rust qui contient des blocs `unsafe` (sensibles à la mémoire).
- Tester un parseur Rust ou une fonction de désérialisation.
- Quand tu veux activer un sanitizer (AddressSanitizer) facilement avec Rust.

Comment l'invoquer

- **Slash command** : `/cargo-fuzz`
- **Phrases déclencheurs (texte)** : "fuzz Rust project", "cargo fuzz target", "Rust fuzzing"
- **Auto-invocation** : Sur demande explicite

Description détaillée

cargo-fuzz, c'est le standard de fait pour fuzzer du code Rust quand tu utilises Cargo (le gestionnaire de paquets et de build de Rust). Il s'utilise comme une sous-commande Cargo : `cargo fuzz init`, `cargo fuzz run mon_test`, et voilà. Sous le capot, il utilise libFuzzer comme moteur, mais tout est branché automatiquement avec les bons flags de compilation pour Rust.

Rust est censé être un langage "memory safe" par défaut, donc tu pourrais te dire "à quoi bon fuzzer ?". Réponse : Rust empêche les bugs mémoire classiques, mais il reste plein d'autres bugs possibles. Tes parseurs peuvent paniquer sur une entrée tordue, ta logique de validation peut avoir des trous, et si tu utilises des blocs `unsafe`, là toutes les protections sautent. cargo-fuzz t'aide à trouver tout ça.

Tu écris un petit fichier `fuzz_target!` qui reçoit des bytes aléatoires, tu lances la commande, et l'outil tourne en boucle en générant des entrées de plus en plus exotiques jusqu'à trouver un crash. Le support des sanitizers (comme AddressSanitizer) est intégré, ce qui est très utile pour le code `unsafe`.

Pour aller plus loin

Pour les exemples concrets, options de configuration et patterns avancés, voir le SKILL.md original.

Source

- **Plugin** : `trailofbits/testing-handbook-skills`
- **Nom interne** : `cargo-fuzz`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/testing-handbook-skills/1.0.1/skills/cargo-fuzz/SKILL.md`

Revision #2
Created 2026-05-11 21:19:53 UTC by thymon
Updated 2026-05-11 21:37:27 UTC by thymon