

Rust auf dem Server: Ressourceneffiziente Microservices mit Rust

Ziel und Nutzen

Teilnehmer/innen lernen die Rust-Grundlagen um hochperformante und effiziente Backend-Services zu entwickeln

Zielpublikum

Software Entwickler welche Rust in Projekten professionell einsetzen möchten. Sowohl Java, C#, Python, PHP, Backend, Smartphone App als auch JavaScript Entwickler werden Anschlusspunkte finden.

Inhalt

Einstieg in die kompilierte Programmierung mit Rust, einer blitzschnellen, speichersicheren Sprache mit modernen high-level Features, einem mächtigen "Ownership"-Modell, das Sicherheit gleichzeitig mit minimalem Overhead zur Laufzeit garantiert.

Der Kurs wird durchgeführt von [Letsboot](#)

- Warum Rust - Der GC gap, trade offs and die Lernkurve
- Bausteine: Datentypen und Variablen
- Kontrollfluss: Verzweigungen und Schleifen
- Funktionen & Closures: Modulare Code erstellen, sogar über Threads hinweg
- Ownership und Borrowing: Die Eckpfeiler von Rusts Speichersicherheit
- Traits: Verhalten von Typen definieren
- Kontrolliertes Fehler-Handling: Der Result-Typ
- Projekte mit Cargo: Der Rust Package Manager
- Der Borrow Checker: Der Wächter über die Speichersicherheit
- Unit Tests: Nahe vom Code und sogar in der Dokumentation
- Einführung REST API Framework (Axum)
- Einführung in die Datenbankabstraktion (Sqlx)
- Entwicklung von REST Endpoints
- Verbinden zu bestehenden REST APIs, Datendeserialisierung (Serde)
- Implementation eines Beispielprojekts mit CRUD
- Entwicklung von gut dokumentierten CLI apps (Clap)

Voraussetzungen

Erfahrung in der Entwicklung mit objektorientierten oder funktionalen Programmiersprachen sowie Grundverständnis der Entwicklung Webservices (z.B: REST APIs).