

Updated on 09/12/2024

Sign up

# Axum training

2 days (14 hours)

## Presentation

Axum is a modern web framework for Rust, designed to build robust, high-performance web APIs. It relies on the Tokio runtime for asynchronous management, making it particularly well-suited to high-performance applications capable of handling large numbers of simultaneous requests without blocking.

This course will give you an in-depth understanding of web development, from route implementation to database integration, query management and application security. You'll learn how to harness the full power of Rust to create high-performance, secure and scale-ready REST APIs.

At the end of this course, you'll be able to develop scalable, high-performance web applications. You'll also be able to integrate databases, manage asynchronous concurrency, and apply best security practices to guarantee the reliability and scalability of your projects.

As with all our training courses, this one will introduce you to the latest version of Axum, [axum - v0.7.8](#).

## Objectives

- Master the basics of web development
- Creating high-performance REST APIs
- Integrating databases with sqlx
- Implementing security strategies
- Learn to manage competition and asynchronism

## Target audience

Developers.

# Prerequisites

- A solid understanding of basic Rust concepts
- Practical experience with the Rust language, ideally having completed projects or our [Rust training course](#)
- Familiarity with system and asynchronous programming concepts
- Configured Rust development environment, including Cargo and Rust build tools

# Software requirements

- Installing Docker and Docker Compose
- Installation of PostgreSQL or another relational database management system
- Configured Rust development environment, including Cargo and Rust build tools

# Axum training program

## INTRODUCTION TO AXUM

- Presentation of Axum and its position in the Rust ecosystem
- Comparison with other Rust frameworks such as Actix-Web and Rocket
- Installing and setting up an Axum project

## ROAD MANAGEMENT

- Route definitions
- HTTP method management (GET, POST, etc.)
- Using extractors to process parameters and queries
- Implementation of nested routes and middleware

## ASYNCHRONISM AND AXUM

- Understanding Tokio's role
- Use of ``sync`` and ``await`` in handlers.
- Error handling with ``result`` and error propagation

## QUERY AND RESPONSE MANAGEMENT

- Data extraction from JSON queries and forms.
- Building JSON responses with ``serde``.

- Add custom HTTP headers.

## MIDDLEWARE AND SECURITY

- Creation and use of middleware for logging, authentication and global error handling.
- Security implementation: CORS, CSRF, and secure cookie management.

## DATABASE INTEGRATION

- Introduction to `sqlx` and connecting to a database.
- Executing SQL queries for CRUD
- Connection management and performance optimization.

## TESTING AND DEPLOYMENT

- Writing unit and integration tests for endpoints
- Use of tools like `cargo test` to automate tests
- Deployment strategies for applications

## Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

## Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level on different types of technology, as well as his or her expectations and personal objectives with regard to the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

## Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

## Sanction

A certificate will be issued to each trainee who completes the course.