

**Web APIs with Rust and ActixWeb**

**Course Number:** RUST-118
**Duration:** 3 days

**Overview**

This advanced Rust training course teaches attendees how to build REST APIs with Rust and ActixWeb. Participants learn how to create routes, handlers, and extractors and how to connect to a database and perform CRUD operations.

**Prerequisites**

This Rust course assumes prior experience with Rust. Experience with the SQL language and PostgreSQL or SQL Server is beneficial, but not required.

**Materials**

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub through documentation and extensive code samples. Students practice the topics covered through hands-on lab exercises.

**Software Needed on Each Student PC**

* A free, personal GitHub account to access the courseware
* Permission to install Rust and Visual Studio Code on their computers
* Permission to install Rust Crates and Visual Studio Extensions

If students cannot configure a local environment, a cloud-based environment can be provided.

**Objectives**

* Understand the principles of building a REST API
* Create web APIs with Rust and Actix
* Connect a REST API to a database (PostgreSQL or SQL Server)
* Configure routes
* Explore extractors, handlers, and middleware
* Publish and host a REST API

**Outline**

* Introduction
* REST Overview
	+ What is REST?
	+ What is a REST API?
	+ What is a REST Client?
	+ HTTP Methods and Resource Operations
	+ URLs, Query Strings, and Path Parameters
	+ HTTP Status Codes
	+ HTTP Headers and Body
	+ CURL and Postman
* Web APIs with Actix
	+ What is a Web API?
	+ What is Actix?
	+ How does it compare to Express, Flask, and ASP.NET MVC?
	+ Create an Actix Project
	+ Run and Debug Actix Applications with VS Code
* Routing
	+ What is Routing?
	+ Map HTTP Routes to Rust Functions
	+ Route Parameters
	+ Wire up Routes in an ActixWeb Application
* Extractors
	+ What is an Extractor?
	+ Path Extractor
	+ Query String Extractor
	+ JSON Extractor
	+ URL-Encoded Form Extractor
	+ Other Extractors
	+ Custom Extractors
* Request Handlers
	+ What is a Request Handler?
	+ Stages of Handling a Request
	+ Responder Trait
	+ Respond with a Custom Type
	+ Streaming Response Body
	+ Different Response Types
* Middleware
	+ What is Middleware?
	+ Request/Response Pipeline
	+ Logging
	+ User Sessions
	+ Error Handling
	+ Custom Middleware
* Static Files
* Database Programming
	+ What is a Database?
	+ Connect to a Database
	+ Query data from the database
	+ Modify data in the database
* Testing
	+ Integration Testing
	+ Stream Response Testing
	+ Unit Testing Extractors
	+ Unit Testing Middleware
	+ Unit Testing Responders
* Deployment
* Build a Release Version
* Public the Release Version
* Hosting Considerations
* Dockerize the Application
* Conclusion