

Updated 06/12/2023

Sign up

Framework Ergo training

2 days (14 hours)

Presentation

The aim of the Ergo framework is to leverage the experience of Erlang/OTP with the performance of Golang. The framework implements the DIST protocol, ETF data format and OTP (GenServer/Supervisor/Application) design patterns.

[Ergo](#) is a framework for creating microservices using Erlang/OTP technologies and design patterns in Golang. It enables the design of complex, distributed and extremely fast solutions using technologies from the Erlang/Elixir world.

It enables you to create reliable, high-performance microservices solutions with native integration into the Erlang infrastructure. [Ergo's main advantage](#) is its speed: in network terms, it's up to 5 times faster than the original Erlang infrastructure.

In our Ergo training course, you'll learn a range of useful features that you can't find in the HTTP world. At the end of our training, you'll be able to create complex, distributed microservices solutions (machine learning, data processing pipelines, etc.).

This Ergo Framework training course will teach you the latest version of [Ergo 2.0](#).

Objectives

- Discover the basic functions of the Ergo framework
- Create high-performance microservices solutions with Erlang infrastructure
- How to implement EMPD
- Connecting Phoenix nodes with Ergo nodes

Target audience

- Developers
- Technical architects
- Project managers

Prerequisites

- Knowledge of Golang
- Knowledge of Erlang and Elixir languages

Framework Ergo training program

Introduction

- Introducing Golang
- Discover an Ergo framework

Creating microservices

- DIST protocol
- ETF data format
- OTP design patterns
 - gen.Server
 - gen.Supervisor
 - gen.Application
 - gen.Stage
 - gen.Saga
- Erlang 24 support
 - Alias
 - Remote Spawn
- Launching Erlang-type processes
- Register/unregister processes with a single atom

Ergo framework features

- The supervision chain
 - Restart
 - Monitoring stop
- Pub / Sub
 - Creating a producer/consumer architecture
 - The concept of counter-pressure
- Process monitoring
 - Monitor creation
 - Message Down
- Process linking

- Service monitoring
 - Monitoring a service node
 - Receive NODE DOWN

EMPD

- EMPD implementation
- Running the node without external processes
- Managing connection interruptions

Elixir Phoenix

- Connecting a Phoenix node to an Ergo node
- Create a pg2 gen.Server
- Register a pg2 gen.Server

Development and debugging

- Debugging options
 - Ergo.trace
 - Ergo.norecover

Companies concerned

This training 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 of proficiency in different types of technology, as well as his or her expectations and personal objectives for 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 inputs from the trainer supported by examples and

brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire is used to check that skills have been correctly acquired.

Sanction

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