

Updated 10/30/2024

Sign up

Modular Monolith training

2 days (14 hours)

Presentation

Modular Monolith is a modern architectural approach to building robust, scalable software systems by combining the advantages of the monolith with modular organization.

Unlike microservice architectures, a modular monolith maintains simplicity of deployment while dividing functionality into independent modules, facilitating maintenance and scalability.

Thanks to a modular architecture, you can better manage dependencies, isolate responsibilities and improve overall performance, while avoiding the complexities associated with [microservices](#) (such as managing communications between services and multiplying databases).

Our Modular Monolith training course will guide you through the implementation of a modular architecture in a [monolithic system](#), showing you how to design reusable modules, organize dependencies, optimize performance and guarantee scalability.

You'll also discover how to migrate from a microservices architecture or a traditional monolith to a modular system while minimizing risks and technical debt.

Objectives

- Understand the basic principles and benefits of modular architecture
- Design and organize independent modules in a monolithic system
- Master scalability and performance optimization techniques in a modular monolith
- Know how to migrate from a microservices architecture or a traditional monolith to Modular Monolith
- Implement best practices for continuous maintenance and refactoring

Target audience

- Software developers and architects
- Technical project managers
- DevOps engineers

Prerequisites

- Knowledge of software architectures (monoliths, microservices)
- Experience in software development with languages such as Java, .NET, or Python
- Familiarity with CI/CD and dependency management concepts

Modular Monolith training program

Introduction to the Modular Monolith concept

- The difference between microservices and traditional monoliths
- What is a modular monolith?
- The advantages of modularity in a monolith (simplicity, lower costs, easier maintenance)
- Modular Monolith fundamentals (separation of responsibilities, single deployment, well-defined interfaces)
- Scenarios where a modular monolith is preferable to microservices
- Common challenges of microservice systems and how a modular monolith can solve them

Modular monolith architecture

- Structuring and organizing a modular project
- Design patterns for modular architecture
- Managing dependencies between modules
- Module insulation: principles and practices
- Internal communication between modules (internal APIs, explicit interfaces)
- Transaction management in a modular monolith (Saga patterns and others)

Practical implementation of a Modular Monolith

- Choose a language and framework suited to modularity (Java, .NET, Python, etc.).
- Setting up a modular project structure
- Using packages, namespaces and other tools
- Creation of independent, reusable modules
- Techniques for maintaining modularity over time (refactoring, regression testing)
- CI/CD tools

Scalability and performance in a modular monolith

- Performance optimization in a modular monolith
- Horizontal vs. vertical scaling for monolithic systems
- Load management strategies (database partitioning, caching)
- Fault management and resilience in a modular monolith
- How to ensure high availability (HA) and disaster recovery (DR)
- Monitoring and alerts

Use cases, migration and best practices

- Case study: successful and unsuccessful migrations to a modular monolith
- How to migrate from a traditional monolith or microservices system to a Modular Monolith
- Strategies for managing third-party dependencies in a modular monolith
- Best practices for maintaining modularity over the long term
- Continuous factoring and technical debt management in a monolith

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 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 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.