

Updated 05/28/2024

Sign up

Hexagonal Architecture with JAVA

3 days (21 hours)

Presentation

Dive into the heart of Hexagonal Architecture with JAVA and master the fundamental principles of this innovative architectural approach. Design software systems that are modular, flexible and easy to maintain.

During our Hexagonal Architecture with JAVA training course, we'll explore in detail the key concepts, design principles and best practices for developing robust, scalable applications.

You'll discover how to structure your code in such a way as to isolate the functional core of your application from external dependencies, thus facilitating [automated testing](#) and change management.

We'll also look at the various facets of Hexagonal Architecture, such as ports, adapters, use cases, and how to implement them efficiently using the JAVA programming language.

At the end of this Hexagonal Architecture with JAVA training course, you'll have a solid understanding of Hexagonal Architecture and be able to apply it to your JAVA projects to better manage software complexity.

Objectives

- Understand the principles of Hexagonal Architecture and Clean Architecture in Java
- Master test-driven development (TDD) techniques
- Acquire advanced Domain-Driven Design (DDD) skills

Target audience

- Technical Leaders
- Backend developers
- Full Stack Developers
- Technical architects

Prerequisites

Prior knowledge of JAVA programming would be an asset.

Software requirements

- Java and Docker installed
- An IDE
- A database

Program of our Hexagonal Architecture with JAVA training course

Day 1: Foundations and Practices

Introduction

- Introducing Hexagonal Architecture
- Introducing Clean Architecture
- Demonstrate the concepts and principles of architectures in Java
- Starting live coding

TDD

- Introduction to TDD
- Full use of TDD
- Benefits
- Refactoring
 - TDD
 - Clean Architecture

Day 2: Infrastructure and advanced

Domain-Driven Design testing

- Introduction to SDD

- The main strategic patterns
 - Bounded Contexts
 - Context Mapping
- The main tactical patterns
 - Aggregates
 - Entities/Value Objects
 - Repositories
- Sizing aggregates
- Influence of "competition" aspects
- Implementation of Domain-Driven Design concepts

Day 3 :

Continuity of live coding

- New management rules
- Use Cases
- Agile project management approach

Database

- Database integration with Hibernate ORM
- Demonstrating integration testing with Test Containers

Spring

- Adding a REST controller (Spring-Boot)
- Demonstrating end-to-end testing with Spring MockMVC

API

- Add a Google Place-type API
 - Secondary adapter
- Demonstration of "InMemory" usage
- Real dependencies" usage

Further information

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.