AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated 10/25/2023

Sign up

Hexagonal Architecture and TDD training

3 days (21 hours)

Presentation

Hexagonal Architecture is not an architecture in the strict sense of the word, but a set of architectural principles put forward by Alistair in 2005 and refined by a whole host of other authors: Robert C. Martin aka Uncle Bob in 2012.

The latter has named its variant of the hexagonal "Clean Architecture", providing a number of guidelines and preventive measures.

These architectural principles deliver a whole host of benefits:

- Strong decoupling between the business part of the application and the infrastructure/technology part, making it much easier to design the brain of the application.
- Business part testability greatly improved; real TDD made possible!
- Ability to postpone infrastructure technology choices.
- Ability to change technologies effortlessly, avoiding time-consuming redesigns

At the end of this Hexagonal Architecture training course, you'll learn to master the essential principles, produce organized software and create high-performance domain models.

Objectives

- Master the principles of Hexa/Clean Architecture, such as dependency inversion
- Know how to start a project from scratch using TDD and Hexa Architecture with a "code design emergence" mindset
- Awareness of crucial global technical architecture decisions
- Know how to handle behavior-oriented TDD in a Hexagonal Architecture for increased, seamless productivity
- Know how to integrate infrastructure components such as a PostgreSQL database and Third-party partner APIs without touching the core application
- Know how to dissociate the application's business logic from the Spring-Boot framework

- Knowing all the pitfalls to avoid in Hexa/Clean Architecture
- Be fully aware of the major difference between TDD and simply writing tests
- How to order the actions to be carried out when creating a Hexagonal Architecture
- All questions answered and popular misunderstandings about practices revealed

Target audience

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

Prerequisites

- Mastery of Java or any other object-oriented language
- Notions of the main OOP concepts: Interfaces / Abstract classes / Polymorphism
- Test writing with JUnit 5 and AssertJ

Technologies used

- Java 20
- Maven 3
- Spring-Boot / Rest APIs
- Hibernate/JPA
- PostgreSQL
- JUnit 5 / AssertJ
- TestContainers (Docker)

Hexagonal Architecture and TDD Training Program

Day 1: Foundations and Practices

Analysis of classic layered architectures

- Introduction to common problems
- Demonstrating the pitfalls and difficulties by example

Understand current limits and aspire to a better approach

Diving into the reversal of dependencies

- Magnificence of the "D" in SOLID
- Practical application for optimizing code

Inverting dependencies creates a natural bridge to Hexagonal Architecture

Transition to Hexagonal Architecture

- History: origins and development
- Comparison with Clean Architecture and Onion Architecture
- Exploring key mechanisms

Once this solid foundation has been laid, how do you ensure effective implementation? TDD

Immersion in Test-Driven Development (TDD)

- Overcoming preconceptions about TDD and unit testing
- Interactive challenges to test understanding
- Introduction to different types of linings

With TDD, we can practice informed coding

Live coding workshop

- Introduction to Behavior-Driven Development (BDD)
- Java 20 coding session
- Focus on TDD test-driven refactoring
- Tips for refactoring tests

Day 2: Infrastructure and Advanced

Testing *Let's get down to the technological side*

of the project Setting up the infrastructure

- Creating controllers with Spring-Boot
- Spring Container IOC configuration
- PostgreSQL integration with Hibernate
- Using TestContainers for the database

Good infrastructure is essential, but knowing how to test it is crucial

Overview of the different types of test

- Distinctions between each type of test

Integration and End-to-End Testing Workshop

- Test-first writing with tools/frameworks
- Development of end-to-end and integration tests

Integrate third-party tools and services while maintaining core/tool decoupling

Exploiting complementary technologies

- Integration with Google Maps API

Continuation of the practical workshop

- Continued live coding

Day 3: Deepening and extending

Let's consolidate our foundations and explore collaborative techniques

Refining management rules

- Workshop to refine business rules

Enter the advanced nuances of Hexagonal Architecture

Exploring "Nested" Hexagonal Architecture

- Hexagonal architecture in an adapter

Application extension workshop

- Challenge for participants
- Live evaluation quiz

Q&A session

- Open discussion of all concepts

Further information

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.

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.