AMBIENT-IT

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

Updated 07/08/2025

Sign up

Unit & Mockito Unit Mastery training

2 days (14 hours)

Presentation

Unit & Mockito Unit Mastery is a training course dedicated to mastering unit testing in Java projects.

It offers a complete overview of JUnit for structuring your tests and Mockito for isolating your business dependencies. The emphasis is on practice, with the implementation of robust testing strategies and continuous integration in your modern projects.

This training course will enable you to professionalize your quality approach and automate the control of your applications, while introducing you to advanced simulation and Test-Driven Development (TDD) techniques.

Thanks to practical workshops, you'll be able to write, organize and industrialize your unit tests using the market's benchmark tools.

As with all training courses, this one uses the latest stable version of Mockito v5.18.0.

Objectives

- Understand the principles and benefits of unit testing in Java
- Use JUnit to structure and organize tests
- Master Mockito to simulate and isolate behaviors
- Optimize code coverage and quality
- Integrate unit tests into CI/CD pipelines

Target audience

- Java developers

- Testers
- Tech leads
- QA engineers

Prerequisites

- Basic knowledge of Java
- Experience in application development desirable

Unit & Mockito Unit Mastery

Understanding unit testing and the Java environment

- Definition and challenges of unit testing
- Differences between unit, integration and functional tests
- Introduction to the JUnit ecosystem
- Setting up and configuring a Java project for testing
- Best practices for organizing tests within a project
- Workshop: Initializing a Maven or Gradle project with JUnit

Mastering the fundamentals of JUnit

- Discover JUnit annotations (Before, After, Test...)
- Create and run simple unit tests
- Using assertions to validate behavior
- Structuring efficient test classes and methods
- Handling exceptions and negative test scenarios
- Workshop: Writing unit tests on a Java business service

Going further with JUnit: parametric tests and coverage

- Implementing parametric tests
- Automating scenarios with different input data
- Measuring and improving code coverage
- Tools for visualizing coverage (JaCoCo...)
- Reporting and analyzing test results

Introduction to Mockito and mocking concepts

- Why use a mock? Principle and usefulness
- Discovering Mockito and its main features
- Creating and injecting mocks into tests
- Differentiating between mock, spy and stub
- Organizing dependencies in unit tests

- Workshop: Isolating a business service with mocks in JUnit

Mastering the art of testing with Mockito

- Configuring expected behavior with when and thenReturn
- Simulating exceptions and edge cases
- Verify interactions with verify
- Using ArgumentMatchers for flexible testing
- Structuring readable and maintainable tests

Industrializing tests and integrating them into CI/CD

- Introduction to Test-Driven Development (TDD)
- Automating test execution in a CI/CD chain
- Configuring pipelines on GitHub Actions, Jenkins...
- Results analysis and regression management
- Assessment and checklist of best practices
- Workshop: Integrating unit tests and mocks into a CI pipeline

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT 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 enabling 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 forthcoming course, 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 training: 60% hands-on, 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 is used to check the correct acquisition

skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.