

Updated on 05/11/2025

Register

Unit & Mockito Unit Mastery Training

2 days (14 hours)

Overview

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

It offers a comprehensive overview of JUnit for structuring your tests and Mockito for effectively isolating your business dependencies. The focus is on practical application, with the implementation of robust testing strategies and continuous integration into your modern projects.

This training 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.

Through hands-on workshops, you will be able to write, organize, and industrialize your unit tests with the industry's leading tools.

As with all training courses, this one uses the latest stable version of the [Mockito v5.18.0](#) software.

Objectives

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

Target audience

- Java developers

- Testers
- Tech leads
- QA engineers

Prerequisites

- Basic knowledge of Java
- Desired experience in application development

Unit & Mockito Unit Mastery

Understanding unit testing and the Java environment

- Definition and challenges of unit testing
- Differences between unit, integration, and functional testing
- Presentation of the JUnit ecosystem
- Installing and configuring a Java project for testing
- Best practices for organizing tests in a project
- Workshop: Initializing a Maven or Gradle project with JUnit

Mastering the fundamentals of JUnit

- Discovering JUnit annotations (Before, After, Test, etc.)
- Creating and running simple unit tests
- Using assertions to validate behavior
- Structure effective test classes and methods
- Managing exceptions and negative test scenarios
- Workshop: Writing unit tests on a Java business service

Going further with JUnit: parameterized tests and coverage

- Implementing parameterized tests
- Automating scenarios with different input data
- Measuring and improving code coverage
- Tools for visualizing coverage (JaCoCo, etc.)
- Test result reports and analysis

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 mocks, spies, and stubs
- Organizing dependencies in unit tests

- Workshop: Isolating a business service with mocks in JUnit

Mastering the art of testing with Mockito

- Configuring expected behaviors with when and thenReturn
- Simulating exceptions and edge cases
- Verify interactions with verify
- Use ArgumentMatchers for flexible testing
- Structure readable and maintainable tests

Industrialize tests and integrate them into CI/CD

- Introduction to Test-Driven Development (TDD)
- Automate test execution in a CI/CD pipeline
- Configure pipelines on GitHub Actions, Jenkins, etc.
- Analyze results and manage regressions
- Review and checklist of best practices
- Workshop: Integrating unit tests and mocks into a CI pipeline

Target companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in new advanced IT technologies or to acquire specific business knowledge or modern methods.

Positioning at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and discussion sessions, and group work.

Assessment

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

of skills.

Certification

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