

Updated on 31/01/2025

Sign up

# Vitest training: Automate your tests

2 days (14 hours)

## PRESENTATION

Our Vitest training course will enable you to improve the quality of your code and facilitate the automation of your unit tests. You'll be able to easily test your Java, Kotlin and Android applications to guarantee the quality of your code.

In this training course for developers, you'll learn how to use Vitest's main functions.

Such as unit test creation, test execution and report generation.

You'll also discover how to integrate Vitest with other technologies such as development frameworks and continuous integration tools to improve code quality and accelerate application delivery.

This course will introduce you to the full range of Vitest functionalities, with a particular focus on best practices in [unit testing](#), code validation continuous integration.

You'll also learn how to configure your unit tests for specific environments, such as Android development or Spring-based applications.

As with all our training courses, this one will introduce you to the latest stable version: [Vitest 0.29](#).

## Objectives

- Configure and use Vitest in a JavaScript project.
- Write unit, asynchronous and integration tests.
- Use mocks to simulate modules, functions and dates.

- Work with snapshots to validate complex user interface or object responses.
- Organize tests using hooks and cleaning methods

## Target audience

- Developers
- Technical architects
- Project managers

## Prerequisites

- Basic knowledge of JavaScript and HTML/CSS

## Vitest training program

### Introduction to testing

- Why test?
- Types of testing: unit, integration, E2E
- Overview of different test frameworks (Jest, Mocha, Vitest)
- Vitest benefits: Fast, integrated with Vite, native support for asynchronous tests

### Setting up Vitest

- Installing and configuring Vitest in a JavaScript project
- Structure of a test project
- Vitest configuration (vitest.config.ts file)

### The describe(), test() and it() functions

- What is describe? Logical test structure
- test() and it(), how to use them to define test scenarios
- Practical exercise: creating an initial test suite for a simple function

### The expect() function and matchers

- Introducing expect()
- Various Vitest matchers: toBe(), toEqual() toHaveBeenCalled(), etc.
- Exercise: Testing simple functions with assertions

### Asynchronous testing

- Testing asynchronous functions with `async/await`
- Using `vitest` to manage promises and deadlines
- Exercise: Testing an asynchronous function returning a promise

## Preparation and cleaning

- Preparation and cleaning hooks: `beforeAll()`, `afterAll()`, `beforeEach()`, `afterEach()`
- Why and when to use them in a JavaScript project
- Exercise: Setting up hooks to initialize and clean data

## Mock Functions

- Introduction to simulated functions in Vitest
- Using `vi.fn()` to create simulated functions
- Exercise: Simulating a callback function in a test

## Mocking Modules

- How to mock external modules (e.g. `axios`, `localStorage`, etc.).
- Using `vi.mock()` to replace a module in tests
- Exercise: Mocking an API call with `axios`

## Mocking dates

- How to simulate dates with Vitest
- Use `vi.setSystemTime()` to set a date and use it in tests
- Exercise: Testing a module that uses dates and times

## Snapshots for user interface testing

- Introduction to snapshots for validating complex responses (e.g. user interfaces or objects)
- Managing and updating snapshots with Vitest
- Exercise: Testing a dynamic user interface and capturing a snapshot

## Integration testing in a JavaScript project

- Testing a module or service in a JavaScript project with multiple dependencies
- Exercise: Testing a dynamic form with multiple components and states

## End-to-end testing with Vitest

- Simulate a complete usage scenario in a JavaScript application
- Mock external services (e.g., send emails, call external APIs)
- Exercise: Testing a complete user flow on a web page

## Closing the course

- Review of concepts covered
- Questions/answers and tips for going further

## 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, 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.

approval.

Ambient IT 2015-2025. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg