

Updated on 01/08/2026

Register

Jetbrains IntelliJ IDEA Training

3 days (21 hours)

Overview

JetBrains IntelliJ IDEA is the leading IDE for efficient development in Java and the JVM ecosystem (Kotlin, Spring). This training course will help you increase your productivity through refactoring, advanced debugging, and the integration of build and testing tools. Ideal for industrializing your workflow on backend and microservices projects.

The goal of our training is to master the key features of IntelliJ IDEA to code faster, navigate a codebase better, secure changes, and diagnose problems. You will learn how to configure the IDE, leverage static analysis, manage dependencies, and automate recurring tasks.

The approach is decidedly practical: guided demonstrations, workshops on a common project (Spring/Maven or Gradle), refactoring exercises, and debugging sessions. Deliverables: reproducible IDE configuration, essential shortcuts, quality checklist, and a project ready to compile, test, and run.

Objectives

- Configure IntelliJ IDEA for a Java/JVM project (SDK, plugins, inspections).
- Quickly navigate and understand a code base (search, diagrams, uses).
- Refactor safely (rename, extract, move) with impact preview.
- Debug and profile efficiently (breakpoints, watches, threads, heap).
- Integrate build, tests, and VCS (Maven/Gradle, JUnit, Git) into a smooth workflow.

Target audience

- Java/Kotlin developers
- Spring backend developers
- Tech leads and quality assurance specialists

Prerequisites

- Basic knowledge of Java or Kotlin
- Understanding of OOP and unit testing
- Familiarity with Git
- Basic knowledge of Maven or Gradle

Technical prerequisites

- Computer with at least 8 GB RAM (16 GB recommended)
- Windows, macOS, or Linux
- IntelliJ IDEA installed (Community or Ultimate depending on context)
- Recent JDK installed (LTS recommended) + access to a terminal

Jetbrains IntelliJ IDEA training program

[Day 1 - Morning]

Getting started with IntelliJ IDEA and configuring the environment

- Install IntelliJ IDEA (Community/Ultimeate) and understand the differences
- Configure the JDK, project SDK, and environment variables
- Explore the interface: Project, Editor, Tool Windows, quick navigation
- Configure the IDE: keymap, themes, essential plugins, saving settings
- Hands-on workshop: Create a Java/Gradle project and validate the JDK configuration.

[Day 1 - Afternoon]

Daily productivity: editing, navigation, and refactoring

- Master navigation: Search Everywhere, Go to Class/File/Symbol, bookmarks
- Speed up editing: live templates, intentions, inspections, auto-import
- Safe refactorings: rename, extract method/variable, move, change signature
- Code quality: formatting, code style, static analysis, and quick fixes
- Hands-on workshop: Refactor a mini-project by applying inspections and quick fixes.

[Day 2 - Morning]

Build and dependencies: Maven/Gradle, project structure, and execution

- Import/open a Maven or Gradle project and understand synchronization
- Managing dependencies, scopes/configurations, and repositories (local/remote)

- Configure Run/Debug Configurations (main, tests, profiles)
- Use built-in tools: terminal, Gradle tasks, Maven lifecycle
- Hands-on workshop: Add a dependency, create a task, and run an application via dedicated configuration.

[Day 2 - Afternoon]

Advanced testing and debugging

- Run and organize JUnit tests: suites, tags, targeted execution
- Debug efficiently: conditional breakpoints, log breakpoints, evaluate expressions
- Analyze execution: frames, variables, watches, smart step into
- Code coverage and identification of untested areas
- Hands-on workshop: Diagnosing a bug with conditional breakpoints and completing tests to secure the fix.

[Day 3 - Morning]

Git in IntelliJ: workflow, review, and conflict resolution

- Configuring Git, managing remotes and branches from the IDE
- Using commit, amend, stash, rebase/cherry-pick with visualization
- Compare and review code: diff, annotations (blame), history
- Resolving conflicts with the merge tool and securing the result
- Hands-on workshop: Create a feature branch, rebase, and resolve conflicts in a guided case study.

[Day 3 - Afternoon]

Optimization, best practices, and advanced customization

- Speed up the IDE: indexing, exclusions, memory, performance tips
- Automate: macros, actions, custom keymaps, favorites, and scopes
- Quality and conventions: custom inspections, code style profiles, import order
- Useful tools: HTTP Client, Database (if Ultimate), scratch files, diagrams as needed
- Hands-on workshop: Build a "productivity setup" (keymap + inspections + templates) and export it for the team.

Target companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in new advanced IT technology 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.

Certification

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