

Updated on 01/08/2026

Register

Jetbrains PhpStorm Training

3 days (21 hours)

Overview

JetBrains PhpStorm is a comprehensive IDE for developing in PHP and managing your entire workflow: intelligent navigation, refactoring, debugging, and testing. This training course will help you save time on your daily Symfony, Laravel, or native PHP projects by improving code quality. By the end of this training, you will have mastered PhpStorm as a production tool: project configuration, code understanding, task automation, and integration of PHP ecosystem tools. You will learn how to leverage indexing, code inspection, templates, advanced searches, and refactorings to deliver faster and cleaner results.

The approach is decidedly practical, with guided workshops and reproducible demos on a common project. Deliverables include an optimized IDE configuration (plugins, inspections, formatting), a debug workflow (Xdebug), test scripts, and a checklist of best practices for working in a team.

Objectives

- Configure PhpStorm for a PHP project (interpreter, composer, standards).
- Navigate and understand a code base through indexing, symbols, and searches.
- Apply safe refactorings and automate formatting.
- Debug with Xdebug and analyze execution (breakpoints, watches).
- Run and integrate tests, linters, and quality tools in the IDE.

Target audience

- PHP developers (beginners to intermediate)
- Symfony/Laravel developers
- Tech leads looking to standardize a team environment

Prerequisites

- Solid foundation in PHP (syntax, OOP)
- Understanding of Composer and dependencies
- Basic knowledge of Git
- Understanding of logs and the command line

Technical requirements

- Minimum 8 GB RAM (16 GB recommended)
- Windows, macOS, or Linux
- PhpStorm installed (recent version)
- Local PHP 8.x, Composer, and a terminal
- Xdebug available (locally or via Docker) for debugging workshops

Jetbrains PhpStorm training program

[Day 1 - Morning]

Getting started with PhpStorm and configuring a PHP environment

- Overview: interface, projects, tool windows, quick navigation
- Configuring PHP: local interpreter, PHP via Docker, versions, and extensions
- Configuring Composer: detection, autoload, scripts, dependencies
- Indexing the project: exclusions, performance, naming conventions
- Hands-on workshop: Create a PHP project (Composer) and configure the interpreter (local or Docker).

[Day 1 - Afternoon]

Daily productivity: editing, refactoring, and code quality

- Advanced editing: intentions, templates, multi-cursor, live templates
- Efficient navigation: Search Everywhere, symbols, usages, structure
- Key refactorings: rename, extract method/variable, move, safe delete
- Inspection and formatting: inspections, quick fixes, PHP CS Fixer, Code Style
- Hands-on workshop: Refactoring an existing module and applying an automated code standard.

[Day 2 - Morning]

Debugging and testing: Xdebug, PHPUnit, and targeted execution

- Configuring Xdebug: local, Docker, path mappings, breakpoints
- Step-by-step debugging: stack, variables, watches, evaluate expression
- Run/Debug Configurations: scripts, parameters, environment variables
- Testing: PHPUnit integration, file/method execution, coverage
- Hands-on workshop: Diagnosing a bug with Xdebug, then writing and running a reproducible PHPUnit test.

[Day 2 - Afternoon]

Frameworks, databases, and APIs: PhpStorm integrations

- Framework support: Symfony/Laravel (routing, services, templates, CLI)
- HTTP Client and REST: requests, environments, variables, simple tests
- Database: connection, SQL console, introspection, favorite requests
- ORM and queries: entity navigation, usage search, basic generation
- Hands-on workshop: Calling an API via HTTP Client and validating data in the database with the SQL console.

[Day 3 - Morning]

Git and collaboration: workflow, review, and conflict resolution

- Configuring Git: remotes, keys, ignore, hooks, and external tools
- Workflow: branches, rebase/merge, stash, cherry-pick, tags
- Conflict resolution: merge tool, history, annotations (blame)
- Review and quality: advanced diff, pre-commit inspections, commit templates
- Hands-on workshop: Implementing a feature in a branch, opening an MR/PR, resolving a conflict, and finalizing a clean rebase.

[Day 3 - Afternoon]

Automation and customization: tools, local CI, and IDE optimization

- Tasks and external tools: Composer scripts, Makefile, npm, custom commands
- Static analysis: PHPStan/Psalm, configuration, execution, and reading reports
- Continuous quality: linters integration, pre-commit, save inspections
- Customization: keymap, useful plugins, scopes, favorites, file templates
- Hands-on workshop: Setting up a local pipeline (tests + static analysis + formatting) triggered before commit.

Target companies

This training is intended for both individuals and companies, small or large.

wishing to train its teams in advanced new IT technology or to acquire specific professional knowledge or modern methods.

Positioning at the start of training

The positioning at the start of training complies with Qualiopi quality criteria. Upon final registration, learners receive 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 reflection sessions, and group work.

Validation

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.