Updated on 01/09/2026

Register

Jetbrains WebStorm Training

3 days (21 hours)

Overview

JetBrains WebStorm is a JavaScript/TypeScript IDE designed to accelerate front-end and Node.js development with intelligent auto-completion, code navigation, and integrated debugging. Ideal for making React/Vue/Angular projects more reliable, standardizing team workflows, and saving time on quality and testing.

This training aims to make WebStorm immediately operational in your projects: environment configuration, understanding inspections, refactorings, dependency management, and integration of modern tools (linters, formatters, tests). The goal is to produce code that is more readable, more secure, and faster to maintain.

The approach is 100% practical: guided demonstrations, workshops on an existing project, debugging and refactoring exercises. You will leave with a reproducible WebStorm configuration (settings, shortcuts, templates), an integrated Git workflow, and a tool checklist (ESLint/Prettier/tests) tailored to your stack.

Objectives

- Configure WebStorm for a JavaScript/TypeScript project.
- Navigate efficiently through a code base and its dependencies.
- Debug a front-end and Node.js application with breakpoints.
- Apply safe refactorings and automate quality (ESLint/Prettier).
- Industrialize the workflow with Git, tests, and npm tasks.

Target audience

- Front-end developers (React, Vue, Angular).
- Node.js/full-stack developers.
- Tech leads wishing to standardize tools.

Prerequisites

- Good foundation in JavaScript and basic knowledge of TypeScript.
- Knowledge of a package manager (npm or yarn).
- Basic knowledge of Git (commit, branch, merge).
- Understanding of build/test scripts (npm scripts).

Technical prerequisites

- WebStorm installed (latest stable version recommended).
- OS: Windows, macOS, or Linux.
- RAM: 8 GB minimum, 16 GB recommended.
- Node.js LTS installed + access to a terminal.
- Access to a Git project repository (local or internal) for workshops.

Jetbrains WebStorm training program

[Day 1 - Morning]

Getting started with WebStorm and setting up the environment

- Install WebStorm and activate the license (JetBrains Account, offline if necessary)
- Configure the project: Node.js, npm/yarn/pnpm, TypeScript, ESLint/Prettier
- Overview of the interface: Project, Editor, Terminal, Tool Windows
- Productivity: navigation, search, basic refactorings, Live Templates
- Hands-on workshop: Create a JavaScript/TypeScript project and apply an ESLint + Prettier configuration.

[Day 1 - Afternoon]

Smart editing, inspections, and refactoring

- Understanding inspections: errors, warnings, quick fixes, and intentions
- Key refactorings: Rename, Extract (variable/function), Move, Inline, Safe Delete
- Auto-imports, module management, navigation between symbols and usages
- Documentation and code reading: types, JSDoc/TS, structure, and breadcrumbs
- Hands-on workshop: Refactoring an existing module (function extraction, renaming, reorganization) using inspections.

[Day 2 - Morning]

Debugging and execution: Node.js, scripts, and configurations

- Run/Debug Configurations: Node, npm scripts, TypeScript (ts-node), env vars
- Debugger: breakpoints, watches, call stack, step over/into/out
- Asynchronous debugging: promises, async/await, exception handling
- Built-in tools: Terminal, HTTP Client, logs, and console
- Hands-on workshop: Debugging a Node/Express API with breakpoints, environment variables, and object inspection.

[Day 2 - Afternoon]

Testing, quality, and automation in WebStorm

- Configuring and running tests: Jest, Vitest, Mocha (run, debug, watch)
- Code coverage and test result analysis
- ESLint/Prettier integration: quick fixes, formatting on save
- Tasks and scripts: npm/yarn/pnpm, macros, custom actions and shortcuts
- Hands-on workshop: Setting up a test suite + coverage and fixing linting issues via quick fixes.

[Day 3 - Morning]

Git and collaboration: efficient workflow in the IDE

- Configuring Git: remotes, branches, .gitignore, line ending management
- Daily workflow: commit, amend, stash, interactive rebase, cherry-pick
- Resolving conflicts with merge tools and advanced comparison
- Code review: diff, annotations, history, searching in history
- Hands-on workshop: Implementing a feature in a branch, interactive rebase, conflict resolution, and preparing a clean history.

[Day 3 - Afternoon]

Productivity optimization and advanced customization

- Keymap and actions: advanced navigation, multi-cursors, structural search/replace
- Templates and generation: Live Templates, File Templates, postfix completion
- Performance management: indexing, exclusions, caches, large monorepos
- Useful plugins and team settings: configuration sharing, EditorConfig
- Hands-on workshop: Build a "team setup" (templates, keymap, EditorConfig, inspections) and apply it to a mini-project.

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.