+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 01/09/2026

Register

Jetbrains RubyMine Training

3 days (21 hours)

Overview

JetBrains RubyMine is a comprehensive IDE that accelerates Ruby and Ruby on Rails development with autocompletion, smart navigation, and refactoring tools. The training focuses on practical use cases: getting started with an existing project, improving quality, and everyday productivity.

This training teaches you how to configure RubyMine for your stack (Ruby, Rails, Bundler) and how to use its key features: code inspection, advanced search, execution and debugging, testing, and Git integration. The goal is to reduce the time spent on diagnosis and maintenance, while making your deliveries more reliable.

The approach is decidedly practical: guided workshops on a Rails application, refactoring demos, and debugging exercises. You will leave with a reproducible IDE configuration, efficient shortcuts and workflows, and a checklist of tools (tests, linters, run configurations) applicable to your projects.

Objectives

- Configure RubyMine for Ruby/Rails, Bundler, and SDKs.
- Navigate efficiently through a code base (symbols, usages, searches).
- Debug an application and analyze runtime errors.
- Run and industrialize tests (RSpec/Minitest) from the IDE.
- Apply safe refactorings and leverage code inspections.

Target audience

- Ruby/Ruby on Rails developers.
- Full-stack developers working on a Rails codebase.
- Tech leads wishing to standardize IDE practices.

Prerequisites

- Basic knowledge of Ruby.
- Understanding of Ruby on Rails (MVC, routes, console).
- Familiarity with Git and the terminal.
- Understanding of automated testing (principles).

Technical prerequisites

- RubyMine installed (recent version) and license/access activated.
- OS: macOS, Linux, or Windows (WSL2 recommended).
- RAM: 8 GB minimum, 16 GB recommended.
- Ruby + Bundler installed, and a working Rails project (provided or internal).
- Internet access to retrieve dependencies (gems) and plugins.

Jetbrains RubyMine training program

[Day 1 - Morning]

Getting started with RubyMine and configuring the Ruby environment

- Install RubyMine and configure the Ruby SDK (rbenv, RVM, asdf) and Bundler
- Create/import a project: structure, Gemfile, dependency management
- Configure the editor: inspections, formatting, file templates, essential shortcuts
- Configure execution: Run/Debug configurations, environment variables, arguments
- Hands-on workshop: Initialize a Ruby project, configure the SDK, and run a script with Bundler.

[Day 1 - Afternoon]

Daily productivity: navigation, smart editing, and refactoring

- Quick navigation: classes, methods, files, symbols, structured search
- Assisted editing: completion, intentions, code generation, live templates
- Ruby refactorings: rename, extract method/variable, inline, safe delete
- Quality: inspections, quick fixes, Ruby conventions, warning management
- Hands-on workshop: Refactor an existing Ruby module by improving readability and conventions without breaking tests.

[Day 2 - Morning]

Ruby debugging and execution analysis

- RubyMine debugger: breakpoints, step into/over/out, evaluate expression
- State inspection: variables, watches, call stack, frames, and scopes
- Debugging Bundler scripts and Rake tasks: configuration and common pitfalls
- Analysis tools: logs, Ruby console, exception handling, and backtraces
- Hands-on workshop: Diagnosing and fixing a bug (nil/NoMethodError) using breakpoints and expression evaluation.

[Day 2 - Afternoon]

Automated testing and quality tools (RSpec/Minitest)

- Configuring and running RSpec or Minitest from RubyMine (run, rerun, focus)
- Debugging tests: breakpoints in specs/tests, isolating a scenario
- Coverage and feedback: reading results, slow tests, organizing test suites
- Linting/formatting: RuboCop, auto-corrections, integration with inspections
- Hands-on workshop: Set up an RSpec suite, add RuboCop, and correct violations using quick fixes.

[Day 3 - Morning]

Ruby on Rails in RubyMine: execution, console, and productivity

- Configuring a Rails project: SDK, Bundler, databases, environment variables
- Launch server, console, and tasks: rails server/console, rake/rails tasks, runners
- Rails navigation: routes, controllers, views, models, associations, helpers
- Generators and templates: scaffolding, snippets, conventions, and structure
- Hands-on workshop: Create a Rails feature (resource + validations) and navigate efficiently between routes, controllers, and views.

[Day 3 - Afternoon]

Git integration, code review, and workflow optimization

- Git in RubyMine: commit, amend, stash, interactive rebase, conflict resolution
- Comparison tools: diff, history, annotate/blame, search history
- Local code review: inspections, TODO, bookmarks, quality checklist before PR
- Advanced customization: keymap, useful plugins, project settings vs. IDE settings
- Hands-on workshop: Perform a mini code review, resolve a Git conflict, and prepare a clean commit (message, scope, lint/tests).

Target companies

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

wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Placement 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 on different types of technologies, 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.