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

Updated on July 15, 2026

Sign Up

Contract Testing Training with Pact

3 days (21 hours)

Overview

Pact is a contract testing solution focused on consumer-driven contracts, designed to secure interactions between API consumers and providers. It allows you to validate HTTP interactions and messages without increasing the number of resource-intensive integration tests, while making contracts more reliable and maintainable.

Our Contract Testing with Pact training course will help you go beyond the fundamentals to master the essential practices of contract testing in a microservices and CI/CD context.

You'll learn how to design relevant consumer tests, define robust matchers, manage provider states, and organize provider-side verification using a clear and reproducible approach.

By the end of the training, you'll be able to reduce integration regressions, improve the scalability of your APIs, ensure more reliable communication between teams, and build a validation strategy compatible with production.

This training also covers industrialization with Pact Broker, integration into pipelines, best practices for structuring contracts, and key considerations to avoid tests that are too strict or too fragile.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand the principles of contract testing and the consumer-driven contracts model.

- Write effective consumer tests with Pact for HTTP APIs and message exchanges.
- Use matchers, provider states, and provider-side validations reliably.
- Automate the process within a CI/CD pipeline, including contract publication and verification.
- Reduce fragile integration tests and improve collaboration between product and platform teams.

Target Audience

- Backend developer
- Frontend developer
- QA Engineer
- Automation Tester
- Software Architect

Requirements

- Proficiency in the fundamentals of automated testing and assertion logic.
- Knowledge of a REST API and HTTP communication.
- Experience with a development language of your choice, such as JavaScript, TypeScript, Java, Python, or Go.
- Understanding of CI/CD principles and build environments.
- Experience working with microservices architecture is a plus.

Technical Requirements

- A recent computer with an up-to-date operating system.
- A stable internet connection to install and run dependencies.
- A development environment with Node.js, Java, Python, or Go, depending on the workshop’s focus.
- A code editor such as Visual Studio Code or an equivalent IDE.
- Access to a terminal, Git, and—if the session is remote—a microphone and headset.

Agenda for our Contract Testing with Pact Training

[Day 1 - Morning]

Fundamentals of Contract Testing

- Understanding the role of contract testing in a distributed architecture.
- Distinguish between consumer, provider, and exchange contract.
- Identify the limitations of traditional integration testing when dealing with microservices.
- Discover the concept of consumer-driven contracts and its benefits.
- Analyze use cases suitable for Pact within a product team context.

- Hands-on workshop: Map an exchange flow between two services and identify the contract points that need to be secured.

[Day 1 - Afternoon]

First consumer tests

- Set up an initial consumer test using the Pact DSL.
- Define a simple HTTP interaction, a request, and an expected response.
- Understand how the contract file is generated from the test.
- Structure readable and maintainable scenarios on the consumer side.
- Avoid overly strict tests that hinder API evolution.
- Hands-on workshop: Write your first Pact test for an API client and generate the associated contract.

[Day 2 - Morning]

Matchers and provider states

- Use matchers to validate the structure without locking in unnecessary values.
- Choose between exact match, type, regular expression, and common formats.
- Describe variable data without making the contract fragile.
- Set up provider states to manage preconditions on the provider side.
- Organize contract scenarios that better reflect real-world usage.
- Hands-on workshop: Enhance a contract with matchers and provider states tailored to various business scenarios.

[Day 2 - Afternoon] Provider-

Side Verification

- Understand the principle of provider verification based on published contracts.
- Set up the test environment to run verifications automatically.
- Link provider states to the data required for each interaction.
- Interpret discrepancies between the expected contract and actual behavior.
- Distinguish between the responsibilities of contract testing and functional testing.
- Hands-on workshop: Implement a supplier verification and correct a detected contract discrepancy.

[Day 3 - Morning]

Industrialization and Pact Broker

- Discover the role of the Pact Broker in sharing contracts and results.
- Publish contracts and centralize verifications across teams.
- Integrate Pact into a CI/CD pipeline to automate checks.

- Understand the concepts of versioning, validation, and compatibility.
- Develop a phased deployment strategy using reliable contracts.
- Hands-on workshop: Simulate a pipeline that publishes a contract, runs the verification, and acts on the result.

[Day 3 - Afternoon]

Best Practices and Production Deployment

- Consolidate a contract testing strategy tailored to the organization.
- Avoid over-specified contracts and scenarios that are overly covered by integration tests.
- Manage API changes without breaking existing consumers.
- Establish collaboration guidelines between product, QA, and development teams.
- Prepare a roadmap to roll out Pact across multiple services.
- Hands-on workshop: Design a Pact adoption plan for a multi-service environment, including priorities, risks, and milestones.

Target Audience

This training is intended for both individuals and companies—large or small—that wish to train their teams in a new advanced IT technology or to acquire specific domain knowledge or modern methodologies.

Assessment at the Start of Training

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency level across various types of technologies, as well as their expectations and personal goals 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 pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Hands-On, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

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