



Advanced API Testing with Postman and REST Assured Training Course

27 June – 8 July 2027
Johannesburg

Advanced API Testing with Postman and REST Assured Training Course

Ref: 747_156965 **Date:** 27 June – 8 July 2027 **Location:** Johannesburg **Fees:** 6400 Euro

Overview

Advanced API Testing with Postman and REST Assured Training Course is a ten-day advanced course for API testers, quality engineers, automation engineers, developers, DevOps teams, and technical leads who leave with a Production-Ready API Test Automation Framework. Participants apply Postman test automation, REST Assured framework design, API contract validation, Newman pipeline execution, and API test governance to expose defects, organize reusable assets, and control release evidence. Agile Leaders Training Center develops advanced API testing capability.

Who Should Attend

- Quality functions responsible for REST API validation and release evidence
- Automation teams responsible for reusable test suites and execution architecture
- Development teams responsible for service behavior and defect diagnosis
- DevOps functions responsible for pipeline gates and test reporting
- Technical leads responsible for API quality controls and test governance

The course assumes participants can read REST requests and responses and use basic Java or scripting, and leaves out API design, application development, performance engineering, and penetration testing.

Departments and Industries

The course supports departments and industries that rely on stable service integrations and repeatable API release checks.

- Software quality and test automation
- Application engineering and DevOps
- Financial services and digital commerce
- Telecommunications and logistics platforms
- Healthcare and professional-service technology teams

Learning Objectives

By the end of this course, participants will be able to:



- Build Postman collections with variables, scripts, assertions, and data sets
- Apply authentication, negative, boundary, schema, and contract checks
- Use mock servers and request chaining for dependent workflows
- Build reusable REST Assured specifications and test components
- Diagnose failures through logs, reports, and pipeline evidence
- Build a Production-Ready API Test Automation Framework

Course Agenda

Day 1: REST Test Strategy and Request Design

- REST Resource and Operation Test Map
- HTTP Request Component Checklist
- Response Behavior and Status Matrix
- API Risk-Based Test Coverage Model
- Test Data and Environment Boundary Plan

Day 2: Postman Environments and Collections

- Postman Workspace and Collection Structure
- Environment and Collection Variable Model
- Request Parameterization Pattern
- Authentication Configuration Matrix
- Collection Documentation and Naming Rules

Day 3: Postman Scripts and Assertions

- Pre-Request Script Control Flow
- Post-Response Assertion Library
- JSON Path Extraction Pattern
- Dynamic Variable and Token Handling
- Assertion Failure Diagnostic Record

Day 4: Negative, Boundary, and Data-Driven Testing

- Negative API Test Case Matrix
- Boundary Value Request Catalogue
- Data-Driven Collection Runner Design
- Stateful Request Chaining Workflow
- Error Response Consistency Checklist



Day 5: Contracts and Service Virtualization

- OpenAPI Operation Coverage Map
- JSON Schema Validation Rules
- Request and Response Contract Checklist
- Postman Mock Server Scenario Design
- Contract Drift Defect Register

Day 6: Newman Execution and Reporting

- Newman Command-Line Execution Profile
- Environment and Data File Control Procedure
- Collection Run Failure Thresholds
- Newman Reporter Configuration
- Execution Evidence Retention Plan

Day 7: REST Assured Test Architecture

- Java API Test Project Structure
- REST Assured Request Specification
- REST Assured Response Specification
- Serialization and Deserialization Pattern
- Reusable Endpoint Client Layer

Day 8: REST Assured Validation and Diagnostics

- Hamcrest Response Assertion Pattern
- REST Assured JSON Schema Validator
- Authentication and Session Filter Design
- Conditional Request and Response Logging
- Failure Triage and Defect Evidence Template

Day 9: CI Integration and Test Governance

- API Test Pipeline Stage Design
- Parallel Execution and Test Isolation Plan
- Quality Gate Decision Matrix
- Flaky Test Classification Register
- API Automation Ownership and Review Model

Day 10: API Automation Framework Practice

- Exercise: Build a Postman Collection Test Suite
- Exercise: Validate Contracts and Negative Scenarios
- Exercise: Implement Reusable REST Assured Components
- Exercise: Diagnose Newman and Pipeline Failures
- Capstone: Production-Ready API Test Automation Framework

Practical Exercises

The course uses suggested activities drawn from financial services, digital commerce, telecommunications, logistics, and healthcare technology.

- Suggested activity: create a data-driven Postman collection with authentication, chaining, and failure diagnostics.
- Suggested activity: compare live responses with OpenAPI operations and JSON schemas, then record contract drift.
- Suggested activity: construct reusable REST Assured request, response, serialization, and logging components.
- Suggested activity: run API tests in a delivery pipeline and evaluate gate evidence, isolation, and ownership.

FAQs

Who suits advanced API testing with Postman and REST Assured, and what does it assume?

Quality, automation, development, DevOps, and technical leadership functions suit the course; it assumes participants can interpret REST exchanges and use basic Java or scripting.

How does advanced API testing differ from introductory Postman training?

Advanced API testing combines risk-based coverage, scripts, data-driven execution, contracts, mocks, REST Assured architecture, pipeline integration, diagnostics, and governance rather than basic request sending.

How should Postman collections support maintainable API testing?

Postman collections should use controlled environments, scoped variables, reusable scripts, explicit assertions, stable naming, separated data, documented dependencies, and diagnostic failure messages.

How can REST Assured test components remain reusable?

REST Assured components remain reusable through shared request and response specifications, endpoint clients, serializers, authentication filters, assertion helpers, configuration boundaries, and isolated test data.



What belongs in a Production-Ready API Test Automation Framework?

A Production-Ready API Test Automation Framework includes coverage rules, collections, schemas, mocks, Java components, test data, environment controls, pipeline execution, reports, diagnostics, quality gates, ownership, and maintenance procedures.

Conclusion

Participants take back a Production-Ready API Test Automation Framework supported by Postman collections, contract checks, REST Assured components, pipeline controls, reports, and governance records. It changes scattered API checks into repeatable release evidence with defined ownership. The framework provides a maintainable basis for detecting defects, diagnosing failures, and controlling API quality decisions.