

Automated API Testing with Rest Assured

4.3 Advanced Scenarios

Learning objectives

- ▶ Validate nested and complex JSON structures
- ▶ Extract data from API responses for further testing
- ▶ Chain multiple API requests together
- ▶ Understand different authentication methods for APIs
- ▶ Apply advanced Rest Assured techniques to real-world scenarios
- ▶ Prepare for building comprehensive integration tests

Validating Nested and Complex JSON

- ▶ APIs often return objects within objects or arrays of objects
- ▶ Validation requires navigating JSON paths to reach nested fields
- ▶ Ensures correct structure and values in the full payload
- ▶ Helps catch subtle data inconsistencies early
- ▶ Useful for verifying responses from relational data endpoints
- ▶ Prepares for testing endpoints with multiple relationships

Extracting Data from Responses

- ▶ Responses can contain dynamic data needed for further tests
- ▶ Extract values using JSON paths (e.g., IDs, tokens, names)
- ▶ Store extracted data in variables for use in subsequent requests
- ▶ Supports testing chained workflows or dependent resources
- ▶ Reduces hardcoding and increases test reusability
- ▶ Ensures tests reflect real API usage scenarios

Chaining API Requests

- ▶ Some APIs require multiple sequential calls
- ▶ Examples: create a resource → update it → verify it → delete it
- ▶ Chaining ensures proper stateful testing
- ▶ Combined with data extraction, you can simulate realistic workflows
- ▶ Supports integration tests that mimic end-to-end usage
- ▶ Improves confidence in multi-step API operations

Handling Authentication

- ▶ Many APIs require authentication or authorization
- ▶ Common types:
 - ▶ API keys in headers
 - ▶ Basic Auth (username/password)
 - ▶ Bearer tokens (JWTs)
- ▶ Rest Assured supports passing credentials with requests
- ▶ Enables testing of protected endpoints
- ▶ Essential for realistic and secure API testing
- ▶ Must validate both successful and unauthorized access

Demo Introduction (Person API)

- ▶ In this demo, we will:
 - ▶ Extract IDs for follow-up tests
 - ▶ Test multiple endpoints in sequence

Lab: Testing the Task API

- ▶ In this lab, you will:
 - ▶ Continue working on the Task API from last lab
 - ▶ Cover PATCH and DELETE
 - ▶ Extract IDs for chaining requests
 - ▶ Verify basic business rules (toggle behavior)

Lab Instructions 1

- ▶ 1. Write a test for:
 - ▶ PATCH /tasks/{id}/toggle
 - ▶ Create a new Task and extract ID from the created Task
 - ▶ Then use the extracted ID in the PATCH request
 - ▶ Validate completed changes value

Lab Instructions 2

- ▶ 2. Write a test for:
 - ▶ DELETE /tasks/{id}
 - ▶ Create a new Task and extract ID from the created Task
 - ▶ Then use the extracted ID in the DELETE request
 - ▶ Validate 204 No Content
 - ▶ Confirm subsequent GET returns 404 Not Found

Key Takeaways

- ▶ Advanced JSON validation is critical for real-world APIs
- ▶ Data extraction and request chaining allow dynamic, dependent tests
- ▶ These techniques are essential for full integration and end-to-end testing
- ▶ Rest Assured provides flexible methods to handle complex scenarios