Automated API Testing with Rest Assured

Introduction to Rest Assured

Learning objectives

- Understand what Rest Assured is and why it is used
- Explain the advantages of automated API testing
- Describe the structure of a Rest Assured test
- Identify the types of HTTP requests supported (GET, POST, PUT, PATCH, DELETE)
- Understand integration with JUnit for automated execution
- Know when and why to use Rest Assured in Spring Boot projects

What Is Rest Assured?

- Java library for testing RESTful APIs
- Provides a fluent, human-readable test syntax
- Makes HTTP request testing simple
- Works together with JUnit
- Ideal for testing Spring Boot controllers
- Popular tool for API test automation

Why Use Rest Assured?

- Removes boilerplate HTTP client code
- Easy validation of responses and JSON fields
- Suitable for integration and end-to-end testing
- Helps confirm API contracts before UI development
- Encourages cleaner test structure

Rest Assured Test Structure

- Rest Assured tests typically follow three steps:
  - Setup (headers, parameters, JSON body)
  - Execute an HTTP method
  - Validate the response
  - Designed to mimic how real API clients behave
  - Integrates naturally with JUnit test lifecycle
  - Tests become readable and expressive

Adding Rest Assured to Your Project

- To use Rest Assured in a Maven project:
  - Add the Rest Assured dependency (test scope)
  - Use JUnit 5 for running tests
  - Requires no Spring configuration
  - Works with any Java test framework, but JUnit is most common

Running Rest Assured with Spring Boot

- When testing a Spring Boot API:
  - Application can be started on a random port
  - Tests communicate with a real HTTP server
  - Allows testing the full HTTP layer
  - Works with your existing controllers, filters, and config
  - Useful complement to unit and repository tests

What Rest Assured Can Validate

- Rest Assured allows verifying:
  - HTTP status codes
  - JSON structure and fields
  - Collection sizes
  - Response content type
  - Headers and metadata
  - Behavior of all HTTP methods (GET, POST, PUT, PATCH, DELETE)

When Rest Assured Should Be Used

- Use it when:
  - Testing REST controllers
  - Verifying API endpoints end-to-end
  - Ensuring correct request/response handling
  - Validating JSON payloads
  - Confirming API behavior before frontend integration
- Not ideal for:
  - Pure business logic tests (use unit tests instead)
  - Pure repository tests (use Spring Data + H2 tests)

Key Takeaways

- Rest Assured is the standard Java tool for API testing
- Simple fluent syntax for HTTP requests and assertions
- Integrates smoothly with Spring Boot and JUnit
- Enables realistic testing of controller behavior
- Helps verify API contracts early in development
- Complements unit and repository testing techniques