Automated API Testing with Rest Assured

4.4 Challenge: Testing a Spring Boot API

Learning objectives

- Understand how to test nested and complex JSON structures
- Use JSONPath to select values inside arrays and objects
- Extract fields and collections into Java types
- Validate child collections and their contents

Testing Nested JSON Structures

- API responses may contain arrays inside objects
- JSONPath lets us access values at any depth
- Child lists must often be validated
- Useful matchers: hasItem, hasItems, hasSize
- Can validate structure and data at the same time
- Helps ensure relationships are returned correctly

Using JSONPath in Rest Assured

- Navigate JSON like a file path
- Index arrays using [0], [1], etc.
- Access nested fields using dot notation
- Extract values directly in assertions
- Extract values into Java variables
- Works with lists, primitives, and objects

Introducing the Demo API

- Based on Smoothe Bar API
- /categories returns nested drink lists
- Shows how to test arrays inside JSON objects
- Includes GET, POST, PUT, DELETE endpoints
- Perfect for practicing nested JSON paths

What the Demo Will Show

- Validate nested arrays in JSON
- Extract fields from lists of objects
- Extract entire child lists
- Chain GET /categories → GET /categories/{id}
- Compare list sizes and contents
- Demonstrate practical JSONPath usage

Lab Overview

- Reuse the Beach Activity project form Section 3.5 - Module 2 and test the following endpoints:
- /activities (GET, POST)
- /activities/{id} (GET, PUT, DELETE)
- /bookings/activity/{id} (POST)
- /bookings (GET)
- /bookings/{id} (GET)
- Validate response status codes, body content, and nested data
- Focus on 8 key tests to cover core functionality

Step 1: Setup

- Open provided Beach Activity project form Section 3.5 - Module 2
- Add Maven dependencies for Rest Assured
- Confirm API runs at http://localhost:8080
- Create a test class for the Rest Assured tests and handle the port used

Step 3: GET Requests

- Test /activities:
  - Validate response code 200
  - Validate bookings nested list size
  - Check at least one guest name per activity
- Test /activities/{id}:
  - Extract first activity ID
  - Validate fields and nested bookings
- Test /bookings and /bookings/{id}:
  - Validate status 200 and fields exist

Step 4: POST Requests

- Create new Activity /activities
  - Validate name, category, price, duration
- Add a booking /bookings/activity/{activityId}
  - Validate guest name, date, number of participants

Step 5: PUT Requests

- Update an existing activity /activities/{id}
  - Modify name, price, or duration
  - Validate updated values using GET

Step 6: DELETE Requests

- Delete an activity /activities/{id}
  - Confirm deletion by GET /activities/{id} → expect 500

Key Takeaways

- Rest Assured supports CRUD and nested JSON testing
- Extracting IDs enables chained requests for integration tests
- Automated tests verify API behavior reliably
- Focused lab ensures students understand core Rest Assured patterns