

# Training: Integration Testing in .NET 8

**Author:** Rolf Rodrigo Krueger

**E-mail:** [rodrigo.krueger@philips.com](mailto:rodrigo.krueger@philips.com)

## Table of Contents

Basic Topics.....	3
Module 1: Introduction to Integration Testing.....	3
1.1 What is Integration Testing? .....	3
1.2 Setting Up a Testing Environment in .NET 8 .....	3
Module 2: ASP.NET Core Integration Testing Fundamentals .....	3
2.1 Working with ASP.NET Core TestHost.....	3
2.2 WebApplicationFactory in .NET 8 .....	3
2.3 Configuring Test Environments and Dependencies.....	3
Module 3: Dependency Injection and Service Mocking .....	3
3.1 Introduction to Dependency Injection in .NET 8 .....	3
3.2 Using Moq to Mock Dependencies .....	4
3.3 Advanced Dependency Injection.....	4
Module 4: HTTP Client Testing and Route Verification .....	4
4.1 Testing HTTP Endpoints with HttpClient.....	4
4.2 Route Testing with Minimal APIs and Route Groups .....	4
4.3 Testing Query Parameters and Request Headers.....	4
Module 5: Error Handling, Logging, and Diagnostics.....	4
5.1 Error Handling in Integration Tests .....	4
5.2 Structured Logging in Tests .....	4
5.3 Using Diagnostics and Tracing for Debugging.....	4
Advanced Topics.....	5
Module 6: Introduction to Behavior-Driven Development (BDD) and Cucumber .....	5
6.1 BDD Concepts .....	5
6.2 Cucumber and SpecFlow Overview.....	5
6.3 Setting Up a SpecFlow Project in .NET 8.....	5
Module 7: Implementing Step Definitions and Running Integration Tests .....	5
7.1 Creating a Feature File for ASP.NET Core APIs .....	5
7.2 Writing Step Definitions for API Scenarios.....	5
7.3 Handling Reusable Steps and Common Test Logic.....	5
7.4 Running Tests and Generating Reports .....	6

## Basic Topics

### Module 1: Introduction to Integration Testing

#### 1.1 What is Integration Testing?

- Differences between unit, integration, and end-to-end testing.
- Benefits and challenges of integration testing.

#### 1.2 Setting Up a Testing Environment in .NET 8

- Project structure for .NET 8 tests.
- Introduction to xUnit and NUnit.
- Introduction FluentAssertions

**Hands-On Activity:** Create a basic integration test project with .NET 8.

### Module 2: ASP.NET Core Integration Testing Fundamentals

#### 2.1 Working with ASP.NET Core TestHost

- Introduction to Microsoft.AspNetCore.TestHost.
- Creating an in-memory server for HTTP tests.

#### 2.2 WebApplicationFactory in .NET 8

- Overview of WebApplicationFactory setup.
- Using HttpClient with WebApplicationFactory.

#### 2.3 Configuring Test Environments and Dependencies

- Environment configuration for testing.
- Overriding app settings for integration tests.

**Hands-On Activity:** Set up a basic WebApplicationFactory with dependency injection and test API endpoints.

### Module 3: Dependency Injection and Service Mocking

#### 3.1 Introduction to Dependency Injection in .NET 8

- Overview of DI and testing use cases.
- Service lifetimes (transient, scoped, singleton) and test implications.

### 3.2 Using Moq to Mock Dependencies

- Introduction to Moq and its usage in .NET 8.
- Creating and injecting mock services in WebApplicationFactory.

### 3.3 Advanced Dependency Injection

- Overriding DI services for specific tests.
- Testing different service configurations.

**Hands-On Activity:** Inject mock services and test multiple configurations in integration tests.

## Module 4: HTTP Client Testing and Route Verification

### 4.1 Testing HTTP Endpoints with HttpClient

- Configuring HttpClient for endpoint testing.
- Sending requests and verifying responses.

### 4.2 Route Testing with Minimal APIs and Route Groups

- Testing route matching and parameters.
- Working with route groups in Minimal APIs.

### 4.3 Testing Query Parameters and Request Headers

- Using query parameters in tests.
- Verifying headers and status codes.

**Hands-On Activity:** Set up tests for Minimal API routes, including parameterized routes and headers.

## Module 5: Error Handling, Logging, and Diagnostics

### 5.1 Error Handling in Integration Tests

- Validating HTTP error codes (404, 500, etc.).
- Testing error handling in API endpoints.

### 5.2 Structured Logging in Tests

- Configuring ILogger<T> for structured logging.
- Capturing and verifying logs during integration tests.

### 5.3 Using Diagnostics and Tracing for Debugging

- Tools and techniques for tracing request failures.
- Tips for debugging complex scenarios in integration tests.

**Hands-On Activity:** Set up error handling and logging verification for multiple endpoints.

## Advanced Topics

### Module 6: Introduction to Behavior-Driven Development (BDD) and Cucumber

#### 6.1 BDD Concepts

- What is BDD, and why is it beneficial?
- How BDD differs from traditional testing.
- Real-world scenarios where BDD is useful.

#### 6.2 Cucumber and SpecFlow Overview

- Introduction to Cucumber, Gherkin syntax, and SpecFlow for .NET.
- How Cucumber integrates with .NET through SpecFlow.

#### 6.3 Setting Up a SpecFlow Project in .NET 8

- Setting up a SpecFlow test project using .NET 8 and SpecFlow.NUnit or SpecFlow.xUnit.
- Installing required packages: SpecFlow, SpecFlow.Tools.MsBuild.Generation.

**Hands-On Activity:** Initialize a new .NET 8 project with SpecFlow and create a basic Gherkin feature file.

### Module 7: Implementing Step Definitions and Running Integration Tests

#### 7.1 Creating a Feature File for ASP.NET Core APIs

- Defining a feature for a sample ASP.NET Core API, such as a product API.
- Mapping API operations to BDD scenarios.

#### 7.2 Writing Step Definitions for API Scenarios

- Creating C# methods for each step in the Gherkin scenarios.
- Using HttpClient to make HTTP requests in steps.
- Parsing responses and making assertions.

#### 7.3 Handling Reusable Steps and Common Test Logic

- Extracting common steps to avoid duplication.
- Creating helper methods for test setup and teardown

## 7.4 Running Tests and Generating Reports

- Configuring your test project to output SpecFlow results in JSON format
- Executing tests and generating a report

**Hands-On Activity:** Write step definitions and run the scenarios, checking the expected outcomes and report.