Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Test-Driven Development with Blazor and ASP.NET Core APIs

Class Duration

21 hours of live training delivered over 3 days.

Student Prerequisites

- HTML, CSS, and JavaScript programming experience
- C# programming experience
- Experience with Blazor and ASP.NET Core APIs (Minimal APIs or controller-based Web APIs) is highly recommended

Target Audience

Developers building Blazor apps and ASP.NET Core APIs who want to work test-first and run their tests in CI/CD. Teams who want to learn the underlying stack first can start with *Full-Stack Web Apps with Blazor and ASP.NET Core APIs*.

Description

This course teaches test-driven development on the current .NET testing stack. Students write xUnit v3 tests on Microsoft.Testing.Platform and use mocks, stubs, and fakes for isolation. Assertions use open-source libraries such as AwesomeAssertions and Shouldly. Students drive Razor components test-first with bUnit and test-drive Minimal API and controller endpoints. Integration tests run in memory with WebApplicationFactory and against real databases with Testcontainers. The course closes with Playwright for .NET browser tests, code coverage, and tests in GitHub Actions or Azure DevOps.

Learning Outcomes

- Apply the red-green-refactor cycle and the principles of TDD
- Organize xUnit v3 test suites on Microsoft.Testing.Platform
- Isolate code under test with mocks, stubs, and fakes
- Write readable assertions with AwesomeAssertions or Shouldly
- Drive Razor components test-first with bUnit

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Test-drive Minimal API and controller endpoints
- Write in-memory integration tests with WebApplicationFactory
- Test data access against real databases with Testcontainers
- Automate browser tests with Playwright for .NET
- Run tests and code coverage in GitHub Actions or Azure DevOps

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install the .NET 10 SDK and the .NET 11 SDK, plus Visual Studio 2026 or Visual Studio Code with the C# Dev Kit, a container runtime (Docker Desktop or Podman), and Node.js on their computers. Also, students will need permission to install NuGet Packages. If students are unable to configure a local environment, a cloud-based environment can be provided.

Training Topics

Getting Started with TDD

- What is Test-Driven Development (TDD)?
- Benefits of TDD
- Challenges of TDD

Principles of TDD

- Three Laws of TDD
- Clean Tests
- One Concept Per Test
- Five Rules: FIRST
  - Fast
  - Independent
  - Repeatable
  - Self-Validating
  - Timely
- Red, Green, Refactor Technique
- Given-When-Then

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Behavior-Focused Test Names

Kinds of Testing

- Unit Tests
- Integration Tests
- E2E Testing
- The Test Pyramid
- Automated vs. Manual Testing
- Testing & DevOps

Testing Parts

- Tests
- Test Suites
- Assertions
- Setup/Teardown
- Mocks, Fakes, Stubs
- Arrange, Act, Assert
- Test Frameworks
- Test Runners
- Code Coverage

.NET Testing Tools

- Testing Frameworks
  - xUnit v3
  - MSTest
  - NUnit
- Microsoft.Testing.Platform vs. VSTest
- Test Runners
  - dotnet test
  - Visual Studio
  - Visual Studio Code
- Testing Libraries
  - Mocking with NSubstitute or Moq
  - AwesomeAssertions and Shouldly
  - Fluent Assertions Licensing
  - bUnit
  - Testcontainers

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Playwright for .NET

xUnit v3

- What is xUnit?
- Facts vs. Theories
- Theory Data
- Fixtures and Shared Test Context
- Test Parallelism
- Test Context and Cancellation
- Dynamic Skipping
- Debugging Tests in Visual Studio
- Debugging Tests in Visual Studio Code

Test-Driving Minimal APIs

- What to Test in a Minimal API
- A REST Service with a Minimal API, Test-First
- Test Projects in a Solution
- Unit Testing Route Handlers and TypedResults
- Testing Validation
- Testing Asynchronous Validation Rules
- Testing Endpoint Filters
- Integration Testing with WebApplicationFactory
- Replacing Services and Authentication in Tests
- Verifying the Generated OpenAPI Document

Testing Controller-Based Web APIs

- What to Test in a Controller API
- Testing Controllers and Action Results
- Testing Filters and Model Binding
- Shared Integration Tests for Both API Styles

Test-Driving Razor Components with bUnit

- What to Test in a Razor Component
- What is bUnit?
- Using bUnit with xUnit
- Defining Tests in C# or Razor Syntax
- The Semantic HTML Comparer
- Interacting with and Inspecting Components

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Triggering Event Handlers
- Providing Cascading Values
- Injecting Services
- Faking HttpClient and API Calls
- Mocking IJSRuntime
- Testing Forms and Validation
- Testing Persistent State
- Testing Required Component Parameters
- Testing Navigation and Not Found Pages
- Testing Authorization and Authentication State
- Snapshot Testing

Testing Data Access

- Test Doubles vs. Real Databases
- EF Core In-Memory and SQLite Providers
- Pitfalls of In-Memory Providers
- Testcontainers for SQL Server and PostgreSQL
- Database Reset and Test Isolation
- Testing Dapper Queries

End-to-End Testing with Playwright for .NET

- What is End-To-End Testing?
- Browsers, Contexts, and Pages
- Locators and Auto-Waiting
- Web-First Assertions
- Page Objects
- Hosting the App Under Test
- Testing Interactive Render Modes
- Testing Circuit Reconnection
- Passkey Sign-In with Virtual Authenticators
- Trace Viewer
- Debugging Flaky Tests

Testing JavaScript Interop Code

- When Blazor JavaScript Needs Its Own Tests
- Vitest for Collocated JavaScript Modules
- Mocking IJSRuntime vs. Testing the JavaScript

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Code Coverage

- What is Code Coverage?
- What can be understood from Code Coverage?
- What are the limitations of Code Coverage?
- Microsoft.Testing.Platform Coverage Extensions
- Coverlet
- Coverage Reports for ASP.NET Core Code

CI/CD Integration

- Test Suites in GitHub Actions or Azure DevOps
- Containers for Integration Tests in CI
- Playwright Tests in CI
- Test Results and Coverage as Pipeline Gates

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved