

QA

ADVANCED

5 DAYS

End-to-End Testing with Cypress (TypeScript)

Build a strong foundation in end to end test automation using Cypress and TypeScript to ensure application reliability and quality. Learn how to design, write, organize, and execute maintainable E2E tests while integrating automated testing into modern software development and delivery workflows.

Why Cypress (TypeScript)?

Type-safe E2E suites

Build Cypress tests with TypeScript for safer refactors.

Network control

Stub and assert API interactions.

Faster critical-path coverage

Automate key journeys with reliable commands.

Pipeline quality gates

Run Cypress in CI with actionable reports.

What you'll be able to do

- ✓ Gain a clear understanding of the concept, purpose, and benefits of E2E testing in software development.
- ✓ Get acquainted with the features, capabilities, and advantages of Cypress and TypeScript for E2E testing.
- ✓ Learn how to set up and configure Cypress and TypeScript for E2E testing in your development environment.
- ✓ Develop the skills to write effective and maintainable E2E tests using Cypress and TypeScript.
- ✓ Explore best practices for E2E test automation, including test design, test organisation, and test data management.
- ✓ Master advanced testing techniques in Cypress and TypeScript.
- ✓ Learn how to execute E2E tests, generate reports, and integrate E2E tests into the software development lifecycle.

Who this is for

- Software Developers
- Quality Assurance (QA) Engineers
- Test Automation Engineers
- DevOps Engineers

Curriculum

01 Cypress Foundations

- Apply cypress foundations using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Project Setup

- Apply project setup using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Selectors and Commands

- Apply selectors and commands using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Assertions

- Apply assertions using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Network Stubbing

- Apply network stubbing using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 Page Objects

- Apply page objects using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 API Testing in Cypress

- Apply api testing in cypress using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Fixtures and Data

- Apply fixtures and data using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 CI Reporting

- Apply ci reporting using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 Flake Control

- Apply flake control using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Best Practices

- Apply best practices using Cypress (TypeScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks