

QA

ADVANCED

4 DAYS

Behavior Driven Development (BDD) with Cucumber

Build a strong foundation in Behavior Driven Development using Cucumber to align business requirements with automated testing. Learn how to write Gherkin feature files, implement step definitions, and test web applications and APIs through integrated BDD workflows.

Why BDD?

Improved Collaboration

BDD with Cucumber encourages collaboration between developers, testers, and business stakeholders.

Better Quality

BDD with Cucumber helps improve the software's quality by ensuring that it meets the business requirements and expected behavior. This reduces the risk of defects and improves customer satisfaction.

Reduced Time and Cost

BDD with Cucumber helps reduce software development's time and cost by detecting defects early in the development process. This reduces the rework required, thereby saving time and effort.

What you'll be able to do

- ✓ Gain a comprehensive understanding of the principles of Behavior Driven Development (BDD) and how it can improve software development processes.
- ✓ Learn how to set up Cucumber for BDD implementation and integrate it with other tools and frameworks.
- ✓ Learn how to write feature files in Cucumber using the Gherkin language, which defines the expected behavior of the software.
- ✓ Learn how to implement step definitions in Cucumber to map the feature file steps to the corresponding code.
- ✓ Learn how to test web applications using Cucumber
- ✓ Learn how to test APIs using Cucumber, including testing HTTP requests and responses and verifying API functionality.

Who this is for

- Software developers
- Testers
- Quality assurance professionals
- Product owners
- Project managers
- Business analysts
- Scrum Masters

Curriculum

01 BDD Foundations

- Apply bdd foundations using Cucumber 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 Cucumber Setup

- Apply cucumber setup using Cucumber 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 Gherkin Features

- Apply gherkin features using Cucumber 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 Step Definitions

- Apply step definitions using Cucumber 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 Data Tables and Hooks

- Apply data tables and hooks using Cucumber 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 Living Documentation

- Apply living documentation using Cucumber 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 Framework Structure

- Apply framework structure using Cucumber 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 CI Integration

- Apply ci integration using Cucumber 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 Reporting

- Apply reporting using Cucumber 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 Maintenance

- Apply maintenance using Cucumber 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 Cucumber 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