Introduction to Cucumber Testing Essentials - TT3650

Gain a solid understanding of practical BDD concepts using Cucumber and Gherkin to drive consistent functional and non-functional tests as part of the regular build process.

Duration: 1 Day
Skill Level: Introductory
Available Format: Instructor-Led Online

What You'll Learn

Overview

This one-day hands-on training course that will provides students with a solid understanding of practical BDD concepts using Cucumber and Gherkin to drive consistent functional and non-functional tests as part of the regular build process. Students will learn the core API, tools, and how to work with them together to create powerful testing harnesses.

BDD is a process that defines features and executable specifications that are expressed using Cucumber and Gherkin. To properly showcase the practical application of BDD throughout this course, implementations of the features and behaviors must be developed and tested. This course leverages Java implementations and testing to transform the features into implemented functionality. As such, it is helpful if incoming students know Java. There is no training on Java in this course. If students do not have a background in Java, they will run the solutions to verify implementation and walk through the practical use of BDD.

Objectives

This **skills-centric** course is about 50% hands-on lab and 50% lecture, designed to train attendees in core Cucumber and web testing skills, coupling the most current, effective techniques with the soundest industry practices. Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises and review.

Working in a hands-on learning environment, led by our expert practitioner, students will learn:

- Test-Driven and Behavior-Driven Testing concepts
- The concept of refining requirements into executable specifications
- How BDD, Cucumber, and Gherkin work together
- Transform requirements into features and feature files
- Creating step definitions
- Using Cucumber and Gherkin to test services
- How to manage Cucumber and improve the organization of scenarios

Audience

This is an introductory-level course is geared for experienced Test Engineers, Quality Assurance Engineers or others needing to learn Cucumber.

Pre-Requisites

Prior experience with basic Java syntax (to work with Cucumber) would be helpful. Participants without a coding background are welcome to follow along in the labs, copying lab solutions as needed.

Agenda

1) Lesson: BDD and Testing

This lesson introduces Behavior-Driven Development (BDD) as an extension of Test-Driven Development (TDD), highlighting how it shifts the focus from testing code implementation to verifying application behavior. Students will learn how requirements can be transformed into features, stories, and executable acceptance criteria expressed in plain language. Through examples and discussion, the lesson shows how BDD improves collaboration between developers, testers, and business stakeholders, while reducing the brittleness of traditional unit tests. Participants will also see how BDD practices set the foundation for automation using Cucumber and Gherkin.

- Understanding BDD
- Transforming requirements into BDD features
- Transforming Features into Stories
- Executable Acceptance Criteria
- Automating tests

2) Cucumber Overview

This lesson introduces Cucumber as a framework that connects business-readable specifications to executable tests. Students will learn how features and scenarios are described in feature files, and how these files serve as both living documentation and test scripts. The session explains how step definitions act as the glue between natural language steps and executable code, and how Cucumber integrates with tools such as JUnit and IDEs like IntelliJ. By the end of this lesson, participants will understand the role of Cucumber in automating acceptance criteria and running tests across different environments

- Feature Files
- Step Definitions
- Cucumber and JUnit
- Implementing and Testing Features
- Tutorial: IntelliJ 2021 Quickstart
- Lab: Creating and Using a Features File

3) Executable Specifications

This lesson introduces Gherkin as the language used to create business-readable specifications that double as automated tests. Students will learn how to structure features, scenarios, and steps using keywords like Feature, Background, Scenario, Given, When, Then, and And. The lesson emphasizes how Gherkin provides a common language for stakeholders, ensuring clarity without exposing implementation details. Participants will also explore how step definitions connect Gherkin steps to executable code, making specifications both living documentation and functional tests.

- Gherkin Keywords
- Feature, Background, Scenario
- Given, When, and other Gherkin concepts
- A detailed look at Step Definition
- The use of Step Definition Files
- Lab: Cucumber Step Definition
- Lab: BDD for a simple Service

4) Step Definitions
This lesson explains how step definitions serve as the bridge between natural language scenarios and executable code. Students will learn how Cucumber expressions and regular expressions are used to map feature file steps to methods, how to handle parameters and optional text, and how to work with data tables and scenario outlines. The session also addresses handling different result states such as success, undefined, failed, pending, and ambiguous steps. Finally, participants will explore techniques for organizing step definitions to keep test suites scalable and maintainable

- Steps and Step Definitions
- Capturing Groups and Arguments
- Returned Results: Success to Ambiguous
- Resolving Collisions
- Organizational Techniques
- Lab: Using Capturing Groups and Arguments

5) Managing Cucumber
This lesson focuses on organizing and controlling test execution as Cucumber projects grow. Students will learn how to use tags to group features and scenarios, filter tests, and manage execution subsets. The session also covers hooks for setup and teardown at different stages, along with advanced options like conditional and global hooks. Additional topics include running dry-runs, customizing generated step definitions, and extending Cucumber with plugins for reporting and identifying unused steps. By the end of this lesson, participants will understand how to manage Cucumber effectively in larger, real-world test suites

- Cucumber Runtime Options
- Working with Plugins
- Filtering Scenarios
- Lab: Using Plugins and Filters
- Lab: Cucumber and Selenium

Additional Topics: Time Permitting

These topics will be included in your course materials but may or may not be presented during the live class depending on the pace of the course and attendee skill level and participation.

1. **Cucumber reporter plugins**

This lesson explores how Cucumber generates reports to provide insight into test execution and application stability. Students will learn how to configure and use built-in reporter plugins such as pretty, HTML, JSON, and JUnit XML, as well as explore the Cucumber Reports Service for publishing and sharing results online. The session also covers other specialized formatters like progress, rerun, summary, and usage, along with guidance on creating custom plugins and formatters for advanced reporting needs. By the end of this lesson, participants will understand how to configure reporting to support team collaboration and continuous integration workflows

- Cucumber reports service
- Built-in reporter plugins
- Defining formatters
- Lab: Generating reports

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.