

Spring Boot Software Testing

Course Duration: 32 Hours

Course Code : SBST-302

1. Course Overview

The **Spring Boot Software Testing** course is designed to equip learners with the skills to write, automate, and execute tests for Spring Boot applications. The course covers different testing strategies, including **unit testing, integration testing, and end-to-end testing** using frameworks like JUnit, Mockito, and Spring Boot Test. Learners will also explore test automation, test-driven development (TDD), and continuous integration practices to ensure high-quality and bug-free Spring Boot applications.

2. What You'll Learn?

- Fundamentals of software testing in Spring Boot applications.
- Writing **unit tests** with JUnit and Mockito.
- Performing **integration testing** with Spring Boot Test framework.
- Testing **REST APIs** and database layers.
- Implementing **Test-Driven Development (TDD)** in Spring Boot projects.
- Using **MockMvc** and WebTestClient for testing controllers.
- Automating testing as part of a **CI/CD pipeline**.
- Best practices for writing maintainable and effective test cases.

3. Target Audience

- Java developers working with **Spring Boot applications**.
- QA engineers and test automation professionals.
- Software engineers aiming to master **unit and integration testing**.
- Students and professionals preparing for roles in **Java full-stack development**.
- Teams looking to improve application quality through **automated testing**.

4. Pre-Requisites

- Basic knowledge of **Java programming**.
- Familiarity with **Spring Boot fundamentals**.
- Understanding of **REST APIs** and database integration (helpful but not mandatory).

5. Course Content

Module 1: Introduction to Testing in Spring Boot

- Importance of testing in software development
- Types of testing: Unit, Integration, E2E
- Testing tools and frameworks for Spring Boot

Module 2: Unit Testing with JUnit and Mockito

- JUnit basics and annotations
- Writing unit tests for service classes
- Mocking dependencies with Mockito
- Assertions and test lifecycle management

Module 3: Integration Testing with Spring Boot

- Setting up Spring Boot Test framework
- Writing integration tests for REST APIs
- Testing data repositories with an in-memory database (H2)
- Using `@SpringBootTest` and `@DataJpaTest`

Module 4: Testing Controllers and REST APIs

- Using `MockMvc` for REST API testing
- Testing with `WebTestClient`
- Validating JSON responses
- Error handling and exception testing

Module 5: Test-Driven Development (TDD)

- Introduction to TDD and Red-Green-Refactor cycle
- Writing tests before implementation
- Developing Spring Boot features with TDD approach

Module 6: Database Testing

- Configuring test databases (H2/PostgreSQL/MySQL)
- Testing JPA repositories
- Handling transactions in tests
- Testcontainers for database testing in Docker

Module 7: Test Automation and CI/CD

- Automating tests with Maven/Gradle
- Integrating tests into CI/CD pipelines (Jenkins/GitHub Actions)
- Generating test reports
- Best practices for automated Spring Boot testing

Module 8: Advanced Testing Strategies

- Testing microservices in Spring Boot
- Contract testing with Spring Cloud Contract
- Performance and load testing basics
- Security testing in Spring Boot applications

