# Selenide + TestNG for UI Automation: A Beginner's Guide

## 1. Introduction

- Overview of Selenide  
- Importance of TestNG in UI automation  
- Benefits of combining Selenide and TestNG  
- Introduction to the tools: Selenium Grid, Owner Library, Lombok, TestNG Listeners

## 2. Setting Up the Project

- Prerequisites: JDK, Maven/Gradle, IDE (IntelliJ/Eclipse), Selenium Grid setup  
- Project Structure  
- Sample pom.xml or Gradle file with necessary dependencies:  
 - Selenide  
 - TestNG  
 - Owner Library  
 - Lombok

## 3. Selenide Basics

- Browser configuration  
- Writing your first Selenide test  
- Understanding Selenide's API:  
 - Element selectors  
 - Assertions  
 - Waits  
 - Actions (click, type, drag and drop)

## 4. TestNG Essentials

- Writing a basic TestNG test  
- TestNG annotations:  
 - @BeforeSuite, @BeforeTest, @BeforeMethod  
 - @Test, @AfterMethod, @AfterTest  
- Parameterizing tests with @DataProvider

## 5. Integrating Selenide with Selenium Grid

- Configuring Selenide to run tests on Selenium Grid  
- Sample configuration for running tests on different browsers and devices  
- Troubleshooting common issues with Selenium Grid

## 6. Using the Owner Library

- Introduction to Owner Library for managing configurations  
- Creating configuration interfaces  
- Loading properties dynamically for different environments

## 7. Lombok in Action

- Overview of Lombok annotations (@Getter, @Setter, @Builder)  
- Simplifying Page Object Model classes using Lombok

## 8. Implementing TestNG Listeners

- Overview of TestNG listeners and their use cases  
- Implementing a custom listener for:  
 - Logging test results  
 - Capturing screenshots on test failure  
 - Sending notifications or updating dashboards  
- Sample listener implementation

## 9. Framework Design

- Modularizing the framework:  
 - Page Object Model with Selenide  
 - Utilities (e.g., for logging, reporting, environment setup)  
 - Test data management  
- Extent Report integration for advanced reporting  
- Strategy for handling dynamic and flaky elements

## 10. Running Tests

- Command-line execution with Maven/Gradle  
- Parallel execution in TestNG  
- CI/CD integration:  
 - Running tests in Jenkins or GitHub Actions  
 - Generating and publishing test reports

## 11. Best Practices

- Tips for writing maintainable and reusable test cases  
- Debugging and optimizing Selenide tests  
- Handling dynamic web elements and waits effectively

## 12. Resources

- Official documentation and community links for Selenide, TestNG, Selenium Grid, Owner Library, and Lombok  
- Troubleshooting guide