

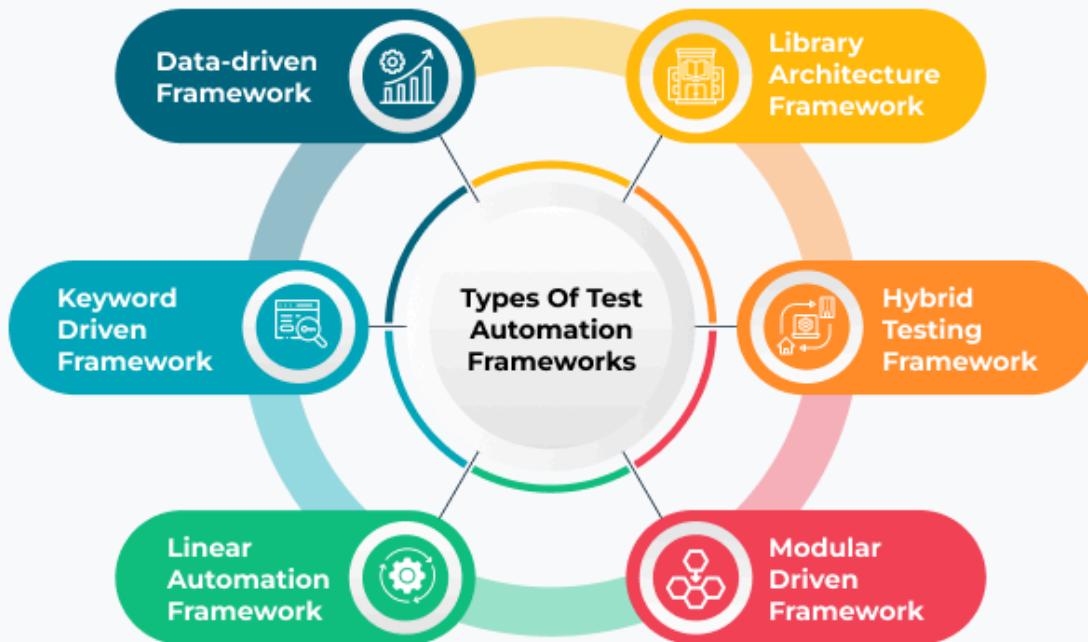
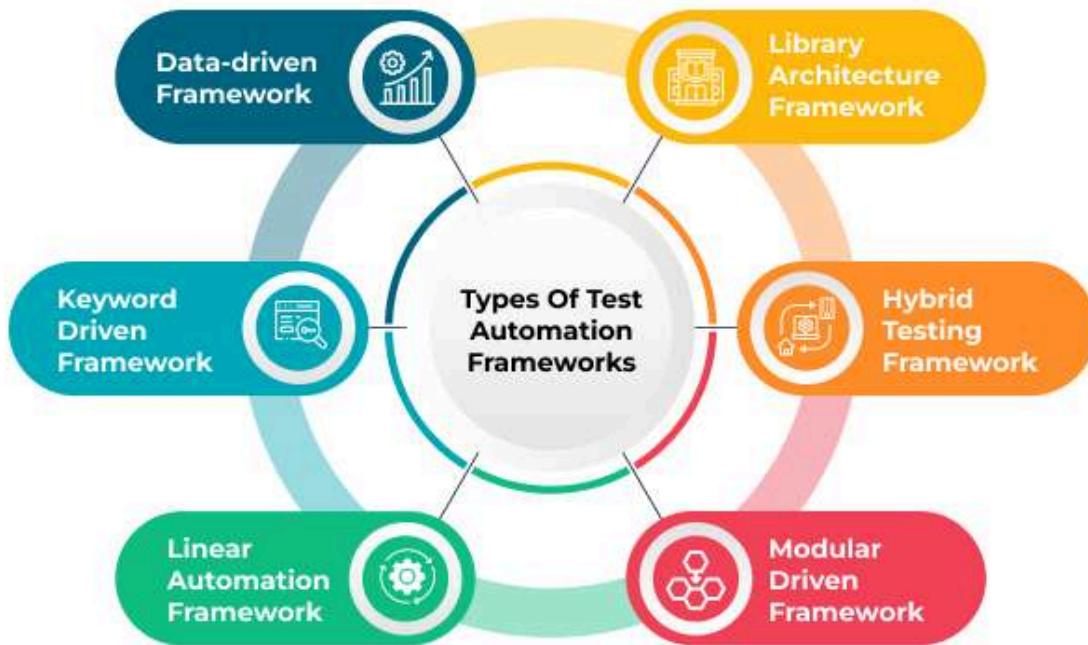
Topics: All Topics ▾

WEB TESTING

Modular Based Framework (Automation)

Anisur 18 Nov 2024 2 331 0

Share



my approach but also inspired me to explore automation framework strategies in greater depth. I'm eager to continue learning and growing in this field and would love to hear your thoughts on modular frameworks—or any other strategies that have strengthened your automation planning! Thanks for reading.

#TestAutomation #ModularFramework #QA #SoftwareTesting #Selenium #TestNG #AllureReports
#AutomationEngineering #LearningAndDevelopment #ContinuousImprovement
#AutomationTesting #SoftwareQuality #TechCommunity #AutomationFramework
#QualityAssurance

Building a Modular Automation Framework with Selenium, TestNG, and Allure Reports: A Journey to Scalable Testing

In the world of test automation, structure, flexibility, and readability are essential. Recently, I've been developing a modular-based automation framework that leverages Selenium in Java, TestNG, and Allure Reports, and I'd love to share both the rewards and challenges of this approach.

Why a Modular-Based Framework?

Creating a modular test framework involves organizing code into reusable components, making it easier to maintain and scale. With modularity, we break down our testing logic into distinct, manageable modules, where each module represents a feature or core functionality in the application. This structure is not only effective but aligns well with the coding approach used by developers on the team. Many development teams follow a modular structure to make their code adaptable and manageable. By using a similar framework in test automation, we create consistency across teams and a clearer picture of each feature's test coverage.

Here's how this approach has made a difference:

- 1. Readability and Maintainability:** By organizing tests and support code into focused modules, the framework remains clean and easy to navigate. Making updates becomes a breeze, especially when adding new application functionality.
- 2. Reusability:** Modularizing code allows for reusable functions, utilities, and helper classes across different test cases. This significantly reduces redundancy and time spent writing repetitive code.
- 3. Scalability:** As the application grows, new modules can be added without impacting existing ones. This makes it easier to maintain high test coverage as the system evolves.

Key Components of My Framework

Let me walk you through the core components I've set up:

- 1. Page Object Model (POM):** Using the POM pattern, I organize locators and actions for each page into dedicated classes. This approach makes updating the framework easy when UI changes occur.
- 2. Data-Driven Testing:** By using JSON files for test data, we can dynamically load data combinations to test multiple input scenarios. TestNG's @DataProvider and the Jackson library make this process efficient, especially for dynamic test data needs.
- 3. Custom Reporting with Allure:** Allure Reports offers an insightful view into test execution, with features like screenshots and logs for each test case. This transparency supports tracking and continuous improvement, as stakeholders can easily review test results.

Example: Login Test Module

Imagine we're automating the login page:

successful login, invalid login, etc.

- **Data Module:** Test data is stored in JSON and dynamically fed to tests using @DataProvider.
- **Report Module:** Each execution is captured in Allure, where failed attempts include screenshots for debugging.

This modular structure makes tests easy to follow and ensures that each function is isolated and reusable. And by following a structure similar to development, we reduce friction between teams and create a more cohesive development and testing workflow.

The Value and Challenges of a Modular Framework

While this approach offers several benefits, it's not without its challenges. Here's a balanced look at the pros and cons:

Advantages

- Enhanced test **execution time**, as modules are independent and run in parallel.
- Quick and easy **debugging**, with Allure Reports highlighting issues clearly.
- Less code **duplication**, thanks to reusable components.

Disadvantages

- Higher initial **setup time**: Designing a modular framework involves detailed planning and configuration, which can delay initial test coverage.
- Requires consistent **maintenance**: As the application grows, modules may need updates or adjustments, which can lead to extra effort if not managed well.
- Learning curve: For team members unfamiliar with the modular approach, understanding the framework's structure can take some time, requiring additional onboarding.

If you're looking to build or improve your automation framework, I highly recommend considering a **modular-based approach**. While it may require careful planning, the payoff in maintainability and scalability is significant and worthwhile. And aligning your test framework with development standards can lead to a more unified, productive testing process.

Happy Testing!

web

selenium

automation

web automation

testng

allure

framework

modularbasedframework

automationtypes

 Share your thoughts

Or

 Start discussion

Related Blogs





WEB TESTING

0 0 288

Comprehensive Guide to Testing Cloud-Based Applications and Services for Functionality...

In today's digital-first world, cloud computing has become the backbone of mod

Habiba

18 Mar 2025

WEB TESTING

0 0 317

Stress Testing in Web Testing: Ultimate Guide ↗ to Performance Optimization

Stress testing is a crucial part of web application testing that evaluates a sys

Abu Hasan

09 Mar 2025



Popular Tags

sqa

testing

qa

software testing

qabrain

testing tool

automationtesting

softwaretesting

mobiletesting

selenium

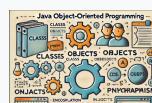
[View All](#)

Popular Post



Can a Software Tester Become a Game Tester? Here's What You Need t...

As the gaming industry continues to grow, fueled by innovations in virtual reali



Understanding Java Object-Oriented Programming (OOP) Concepts

Java is a powerful and widely used programming language known for its versatilit



Essential Bugs to Check for in Game Testing: A Guide for Beginners

Game testing is crucial to ensure a smooth, engaging, and bug-free experience fo

Popular Discussion

01 Top Software Testing Interview Questions and Expert Tips from QA Leaders

02 AI tools for QA engineer

03 What is SQL?

04 Appium, WebDriver

05 What are the most effective strategies you've found for balancing speed and...

[View All](#)

QA Brains

QA Brains is the ultimate QA community to exchange knowledge, seek advice, and engage in discussions that enhance Quality Assurance testers' skills and expertise in software testing.

QA Topics

[Web Testing](#)

[Interview Questions](#)

[Game Testing](#)

[See more →](#)

Quick Links

[Discussion](#)

[About Us](#)

[Terms & Conditions](#)

[Privacy Policy](#)

Follow Us



