# Playwright Language Comparison

TypeScript vs Java vs Python vs .NET (C#)

A Comprehensive Guide for Test Automation Engineers

Created By

## NAVEEN AUTOMATIONLABS

QA Training | Test Automation | Tools Development

### Connect With Us

| Platform | Link |
|---|---|
| YouTube | youtube.com/@naveenautomationlabs |
| LinkedIn | linkedin.com/in/naveenkhunteta |
| Website | naveenautomationlabs.com |

### Our Tools

| Tool | Link |
|---|---|
| LocatorLabs | locator-labs.com |
| QA Daily | qadaily.dev |

# 1. Core Basics

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Primary Support | ✓ First-class | ✓ Official | ✓ Official | ✓ Official |
| Feature Release | First to get | Slightly delayed | Slightly delayed | Slightly delayed |
| Documentation | Best & detailed | Good | Good | Good |
| Community Size | Largest | Growing | Large | Smaller |
| Package Manager | npm / yarn | Maven / Gradle | pip | NuGet |

# 2. Setup & Configuration

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Installation | npm init playwright | Maven/Gradle dep | pip install playwright | dotnet add package |
| Setup Complexity | Very Easy | Medium | Easy | Easy |
| Project Init | Auto-generates all | Manual config | Simple | Simple |
| Config File | playwright.config.ts | XML/Java config | pytest.ini / conftest | .runsettings |
| Browser Install | npx playwright install | Extra step needed | playwright install | pwsh script |

# 3. IDE & Developer Experience

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Best IDE | VS Code | IntelliJ IDEA | VS Code / PyCharm | Visual Studio |
| Auto-Complete | 5/5 | 5/5 | 4/5 | 5/5 |
| IntelliSense | Best-in-class | Strong | Moderate | Strong |
| Error Detection | Before execution | At compile time | Runtime only | At compile time |
| Refactoring | Excellent | Excellent | Limited | Excellent |
| Code Navigation | Excellent | Excellent | Good | Excellent |

# 4. Language Features

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Typing | Static (strong) | Static (strong) | Dynamic | Static (strong) |
| Async Handling | Native async/await | Complex | Native async/await | Native async/await |
| Syntax Simplicity | Clean & concise | Verbose | Very clean | Moderate |
| Learning Curve | Easy-Medium | Steep | Easy | Medium |
| Boilerplate Code | Minimal | Heavy | Minimal | Moderate |
| OOP Support | Full | Full | Full | Full |

# 5. Lines of Code (Same Test)

| Language | Lines of Code | Boilerplate Level |
|---|---|---|
| TypeScript | ~10 lines | Minimal |
| Python | ~8 lines | Minimal |
| Java | ~35 lines | Heavy |
| C# (.NET) | ~30 lines | Moderate |

# 6. Test Runner & Framework

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Default Runner | Playwright Test | JUnit / TestNG | pytest | NUnit / xUnit |
| Built-in Runner | ✓ Yes | ✗ External | ✗ Need pytest | ✗ External |
| Parallel Execution | Built-in & easy | Manual config | pytest-xdist | Manual config |
| HTML Report | Built-in | Extra setup | Need plugins | Extra setup |
| Trace Viewer | Built-in | Available | Available | Available |
| Video Recording | Built-in config | Available | Available | Available |

# 7. Reporting Capabilities

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Built-in HTML Report | ✓ Yes | ✗ No | ✗ No | ✗ No |
| Trace Viewer | ✓ Native | ✓ Available | ✓ Available | ✓ Available |
| Third-party Reports | Allure, etc. | Allure, Extent | Allure, pytest-html | Allure, Extent |
| Screenshot on Fail | One-line config | Manual setup | Manual setup | Manual setup |
| Video Recording | One-line config | Manual setup | Manual setup | Manual setup |

# 8. CI/CD Integration

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| GitHub Actions | 5/5 Easiest | 4/5 | 4/5 | 4/5 |
| Jenkins | 4/5 | 4/5 | 4/5 | 4/5 |
| Azure DevOps | 4/5 | 4/5 | 4/5 | 5/5 Best |
| Docker Support | Excellent | Good | Good | Good |
| Official CI Examples | Most examples | Limited | Good | Limited |

# 9. Async/Await Handling

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Async Model | Native async/await | Sync by default | Both sync & async | Native async/await |
| Code Readability | Clean | Complex | Clean | Clean |
| Error Handling | Simple try-catch | Complex | Simple try-except | Simple try-catch |
| Parallel Friendly | Very | Moderate | Very | Very |

# 10. Team & Ecosystem Fit

| Best For | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Frontend Teams | ✓ Perfect | ✗ Mismatch | ■ Okay | ✗ Mismatch |
| Java Backend Teams | ■ Learning needed | ✓ Perfect | ■ Okay | ■ Learning needed |
| Python/Data Teams | ■ Learning needed | ✗ Mismatch | ✓ Perfect | ✗ Mismatch |
| .NET Enterprise | ■ Learning needed | ■ Okay | ■ Okay | ✓ Perfect |
| Startups | ✓ Perfect | ■ Overkill | ✓ Good | ■ Okay |
| Quick Prototyping | ✓ Fast | ✗ Slow | ✓ Fastest | ■ Moderate |

# 11. Performance

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Test Execution Speed | Fast | Fast | Moderate | Fast |
| Startup Time | Fast | Slow (JVM warmup) | Fast | Moderate |
| Memory Usage | Low | High | Low | Moderate |
| Parallel Performance | Excellent | Good | Good | Good |

# 12. Debugging Experience

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Debug in IDE | Excellent | Excellent | Good | Excellent |
| Playwright Inspector | ✓ Best support | ✓ Available | ✓ Available | ✓ Available |
| Trace Viewer | ✓ Native | ✓ Available | ✓ Available | ✓ Available |
| Console Logs | Easy | Verbose | Easy | Moderate |
| Breakpoints | Seamless | Seamless | Good | Seamless |

# 13. Codegen Support

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| playwright codegen | ✓ Default output | ✓ Supported | ✓ Supported | ✓ Supported |
| Code Quality | Clean | Verbose | Clean | Moderate |
| Ready to Use | Production-ready | Needs cleanup | Production-ready | Needs cleanup |

# 14. Cucumber / BDD Support

| Aspect | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| BDD Framework | Cucumber-JS | Cucumber-JVM | Behave | Reqnroll (SpecFlow EOL) |
| Gherkin Syntax | ✓ Full Support | ✓ Full Support | ✓ Full Support | ✓ Full Support |
| Integration Ease | 5/5 Easy | 4/5 Moderate | 3/5 Complex | 4/5 Moderate |
| Community Resources | 5/5 | 5/5 | 3/5 | 4/5 |
| IDE Plugin | VS Code Cucumber | IntelliJ Cucumber | PyCharm Gherkin | VS Reqnroll |
| Playwright-BDD Lib | ✓ playwright-bdd | ✗ Manual setup | ✗ Manual setup | ✗ Manual setup |
| Parallel Execution | Built-in | TestNG/JUnit | behave-parallel | NUnit/xUnit |
| Allure Reports | ✓ Easy setup | ✓ Easy setup | ✓ Available | ✓ Available |
| Async Handling | Native | Sync by default | Requires async | Native |
| Learning Curve | Easy | Medium | Medium-Hard | Medium |

## BDD Package Installation

| Language | Package / Installation Command |
|---|---|
| TypeScript | npm install @cucumber/cucumber |
| Java | Maven: io.cucumber:cucumber-java:7.x |
| Python | pip install behave |
| .NET (C#) | NuGet: Reqnroll, Reqnroll.NUnit |

## BDD Framework Notes

• TypeScript: playwright-bdd package runs Cucumber with Playwright test runner natively

• Java: Cucumber-JVM with Maven/Gradle — mature ecosystem, excellent IntelliJ support

• Python: Behave (Cucumber-style) — requires async handling with Playwright

• C# (.NET): Reqnroll (SpecFlow successor) — SpecFlow EOL Dec 2024, migrate to Reqnroll

# 15. Final Scorecard

| Criteria | TypeScript | Java | Python | .NET (C#) |
|---|---|---|---|---|
| Ease of Setup | 5/5 | 3/5 | 5/5 | 4/5 |
| Learning Curve | 4/5 | 2/5 | 5/5 | 3/5 |
| IDE Support | 5/5 | 5/5 | 4/5 | 5/5 |
| Code Readability | 5/5 | 3/5 | 5/5 | 4/5 |
| Built-in Features | 5/5 | 3/5 | 4/5 | 4/5 |
| Reporting | 5/5 | 3/5 | 4/5 | 4/5 |
| Community Support | 5/5 | 4/5 | 5/5 | 3/5 |
| Enterprise Ready | 4/5 | 5/5 | 4/5 | 5/5 |
| OVERALL | ■ Best Overall | Enterprise Choice | ■ Quick & Easy | MS Ecosystem |

# 16. Recommendations

| Your Situation | Recommended Language |
|---|---|
| Starting fresh / New project | TypeScript |
| Frontend / Full-stack team | TypeScript |
| Existing Java backend team | Java |
| Quick automation / scripting | Python |
| Data / ML team | Python |
| Microsoft / .NET enterprise | C# (.NET) |
| Want best Playwright experience | TypeScript |

## Key Takeaways

• TypeScript offers the best overall Playwright experience with first-class support

• Python is perfect for quick prototyping and teams with Python background

• Java suits enterprise teams with existing Java infrastructure

• C# (.NET) integrates seamlessly with Microsoft ecosystem

• TypeScript gets new Playwright features first

• All languages support core Playwright functionality

Created by Naveen AutomationLabs

YouTube: Naveen AutomationLabs | Tool: LocatorLabs