

## A) Transitioning from Selenium to Playwright

Question	Complexity	Hints	Remarks
Why did you choose Playwright over Selenium for testing?	Medium	Consider factors like browser support, speed, and API capabilities.	Assesses decision-making skills and understanding of both tools.
How did you evaluate Playwright against other automation solutions?	High	Look for mentions of metrics like execution speed, reliability, and ease of use.	Tests analytical and comparison skills.
Challenges faced during migration from Selenium to Playwright?	High	Ask about adapting test scripts and learning new syntax.	Reveals problem-solving and adaptability skills.
Why implement Playwright in TypeScript?	Medium	Discuss the benefits of TypeScript in testing.	Tests understanding of TypeScript advantages.
Differences between Java/Python and TypeScript in test automation?	Medium	Focus on syntax, type safety, and async handling.	Highlights adaptability to new programming languages.
Unique features of Playwright not found or hard to implement in Selenium?	High	Expect mentions of auto-wait, browser contexts, etc.	Assesses depth of knowledge in Playwright.
Describe the process and timeline for migrating your team to Playwright.	High	Consider training, script conversion, and test execution.	Evaluates project management and team leadership skills.
Can you explain the Playwright architecture?	High	Expect details on the browser, context, and page model.	Tests in-depth technical understanding.

## B) Playwright Terminology and Concepts

Question	Complexity	Hints	Remarks
What is a browser context in Playwright?	Medium	Discuss isolation and use cases.	Tests fundamental knowledge of Playwright.
Is it better to create a new browser context for each test or at the suite level?	Medium	Pros and cons of each approach.	Assesses understanding of test isolation and performance.
How do you manage authentication states in Playwright?	Medium	Discuss methods like storage state.	Tests practical knowledge in test setup.
Explain the hierarchy of browser, context, and page in Playwright.	Medium	Focus on the structural relationship.	Assesses understanding of Playwright's basic components.
How to override test timeout for a specific test?	Low	Mention Playwright test configuration options.	Tests knowledge of customizing test executions.
When and how to use <code>page.pause()</code> ?	Low	Discuss debugging. Mention ensuring it's not in final code.	Tests knowledge of debugging and code quality practices.
Difference between locator and selector in Playwright?	Low	Explain the abstraction layer provided by locators.	Assesses understanding of Playwright's querying capabilities.
How do you handle multiple windows or tabs in Playwright?	Medium	Discuss methods to control multiple pages.	Tests practical skills in handling complex scenarios.

Explain handling shadow DOMs in Playwright.	High	Discuss challenges and solutions in interacting with shadow DOM elements.	Assesses ability to handle advanced web components.
Key flags in your Playwright config file and their impact?	Medium	Focus on flags like <code>headless</code> , <code>timeout</code> , etc.	Tests understanding of configuration management.
Managing sensitive data and preventing code repository exposure?	Medium	Discuss environment variables, secret management tools.	Tests knowledge of secure practices.
Utilizing Playwright's network interception capabilities?	High	Discuss use cases like mocking and monitoring network traffic.	Assesses ability to utilize advanced features for testing.
Implementing and using event listeners in Playwright?	High	Expect explanations of specific use cases.	Tests understanding of asynchronous events and handling.
Handling asynchronous operations in Playwright?	High	Discuss <code>await</code> , async functions, and their necessity.	Assesses understanding of JavaScript/TypeScript async patterns.
Strategies for clearing input fields?	Low	Discuss methods like setting value to empty.	Tests knowledge of basic DOM manipulation.
What are actionability checks in Playwright?	Medium	Discuss how Playwright ensures elements are ready for interaction.	Tests understanding of Playwright's smart waiting mechanisms.
Managing different timeouts in your framework?	Medium	Expect discussion on global vs specific timeouts.	Tests knowledge of test stability and performance tuning.
Difference between serial and parallel mode in testing?	Low	Discuss execution order and resource usage.	Tests understanding of test execution strategies.
Headed vs headless mode in CI environments?	Low	Discuss trade-offs in terms of debugging and performance.	Tests practical knowledge in CI/CD pipeline integration.

## C) Framework Level Questions

Question	Complexity	Hints	Remarks
Explain your project framework with a focus on integration and handling test case challenges.	High	Expect details on reporting, CI integration, and test case optimization.	Tests ability to design and implement comprehensive testing solutions.
Why choose the Page Object Model, and its benefits?	High	Focus on maintainability, reusability, and OOP principles.	Assesses in-depth understanding of design patterns in automation.
Can Playwright.config.ts have multiple configurations?	Medium	Discuss scenarios where multiple configurations might be necessary.	Tests knowledge of advanced framework customization.
Managing environment-level data in your framework?	Medium	Discuss use of environment variables, config files.	Assesses strategies for scalable and secure test data management.
Utilizing projects in Playwright config for property overrides?	High	Discuss use cases for different browser or environment setups.	Tests advanced configuration management skills.

<b>Implementing global setup and teardown in Playwright?</b>	Medium	Discuss use cases like test data setup or environment initialization.	Assesses knowledge of efficient test workflow management.
<b>Tagging test cases and different tags used?</b>	Low	Discuss organization and filtering of tests.	Tests knowledge of test management and execution control.
<b>Customizing test reporting in Playwright?</b>	High	Discuss integration with reporting tools or custom report generation.	Assesses ability to extend framework capabilities.
<b>Video recording in CI: when to enable or disable?</b>	Medium	Discuss trade-offs in terms of resources and debugging.	Tests practical CI/CD pipeline integration knowledge.
<b>Managing test case data in your project?</b>	Medium	Discuss strategies like data-driven testing, external data sources.	Tests knowledge of scalable test data management.
<b>Use of fixtures in Playwright and their application?</b>	High	Discuss use cases for setup, teardown, and test data management.	Assesses understanding of test setup and teardown strategies.

## D) Node & NPM Related Questions

Question	Complexity	Hints	Remarks
What is Node.js and its importance in your project?	Low	Discuss JavaScript runtime environment and server-side scripting.	Tests basic understanding of JavaScript ecosystem.
What is npm and how do you utilize its capabilities in your project?	Low	Discuss package management and script execution.	Assesses knowledge of Node.js ecosystem.
How do you install and update packages using npm?	Low	Discuss commands like <code>npm install</code> and <code>npm update</code> .	Tests practical skills in package management.
Difference between package.json and package-lock.json?	Low	Discuss dependency management and version control.	Assesses understanding of npm's versioning and dependency resolution.
Using scripts in your Playwright project?	Medium	Discuss automation of tasks like test execution, linting.	Tests knowledge of enhancing development workflows with npm scripts.
Difference between npm and npx execution?	Low	Discuss use cases for running packages without installing globally.	Assesses understanding of npm command line tools.
Awareness of ESLint and its importance at the project level?	Medium	Discuss code quality, consistency, and error prevention.	Tests knowledge of code quality tools and practices.

## E) Code Snippets and Practical Application

Question	Complexity	Hints	Remarks
How do you invoke a new page in Playwright?	Low	Simple code to demonstrate opening a page.	Tests basic practical knowledge of Playwright.
Code to create a new browser, context, and page in Playwright.	Medium	Expect a script showing steps to initialize these components.	Assesses ability to handle basic Playwright operations.

<b>Handling dropdown interactions and assertions in Playwright.</b>	Medium	Write code for clicking options and verifying results.	Tests practical DOM manipulation skills.
<b>Code for handling multiple windows in Playwright.</b>	High	Script demonstrating control over multiple browser windows.	Assesses ability to handle complex scenarios.
<b>Function to check visibility of an element.</b>	Medium	Expect a function taking a selector and returning a boolean.	Tests knowledge of element state handling in Playwright.
<b>Making an API call, saving state, and reusing it in a new context.</b>	High	Code demonstrating API interaction, state management, and context initialization.	Tests advanced integration and state management skills.