

1. What is CI/CD and how do automated tests run in this process?

CI/CD (Continuous Integration / Continuous Delivery or Deployment) is a practice where code changes are frequently integrated into a shared repository and automatically built, tested, and deployed.

In this process, automated tests run as part of the CI pipeline after each code commit or pull request. For example, Playwright end-to-end tests are executed automatically to verify that critical application functionality works correctly before changes are merged or deployed. This helps detect issues early and ensures application stability.

2. What is the difference between E2E, Smoke, and Regression testing?

- **End-to-End (E2E) testing** validates complete user workflows from start to finish, ensuring that all integrated components work together as expected.
- **Smoke testing** is a small subset of critical tests executed to verify that the main functionality of the application works after a build or deployment.
- **Regression testing** ensures that new changes have not broken existing functionality by re-running previously passed test cases.

3. Why are waits important in automation and what is the correct way to use them in Playwright?

Waits are important because modern web applications are asynchronous, and elements may not be immediately available for interaction.

In Playwright, the correct approach is to rely on built-in auto-waiting mechanisms and assertions such as `toBeVisible()` and `toHaveURL()` instead of using hard waits like `waitForTimeout`. This makes tests more stable, reliable, and less flaky.