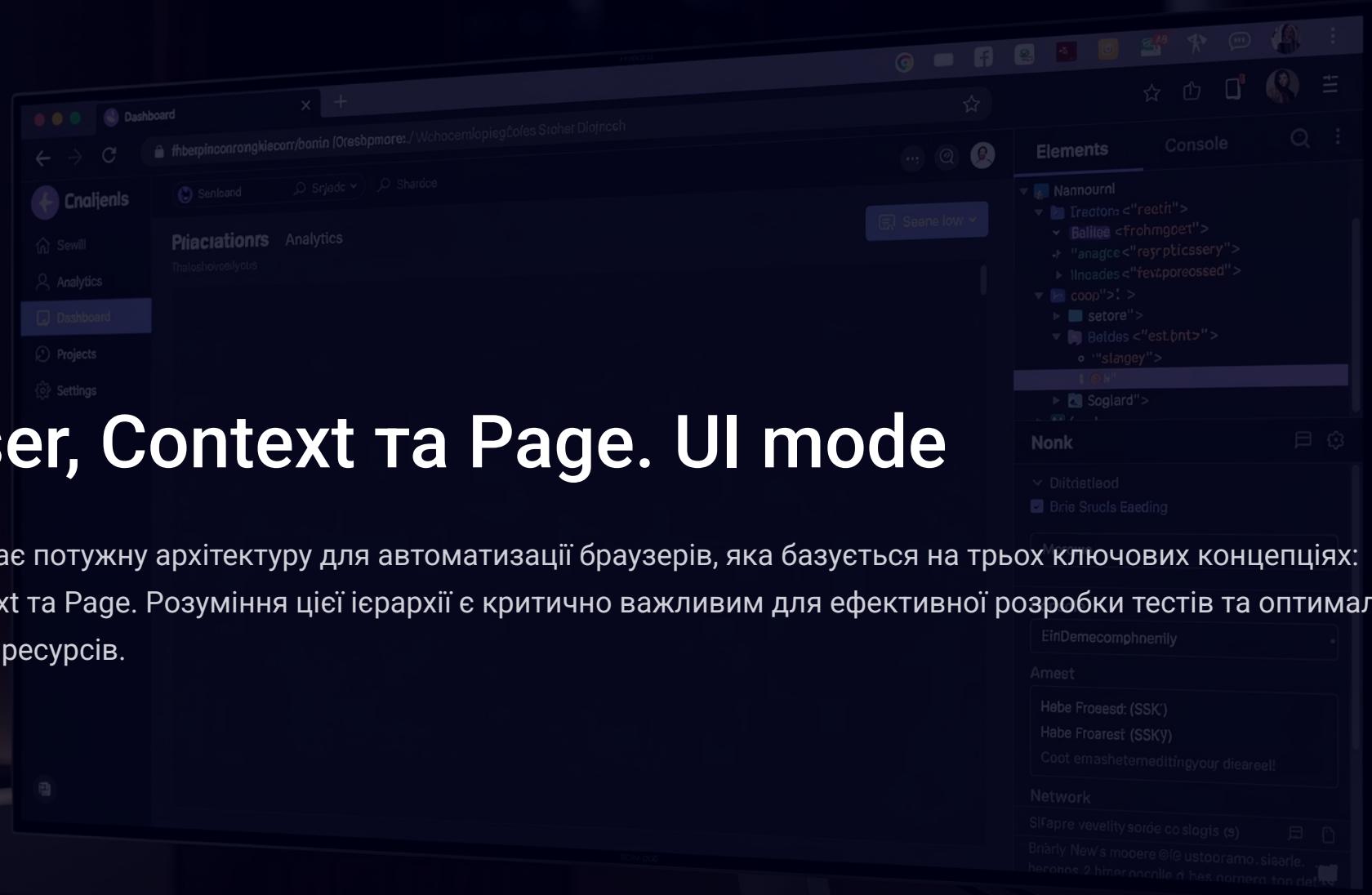
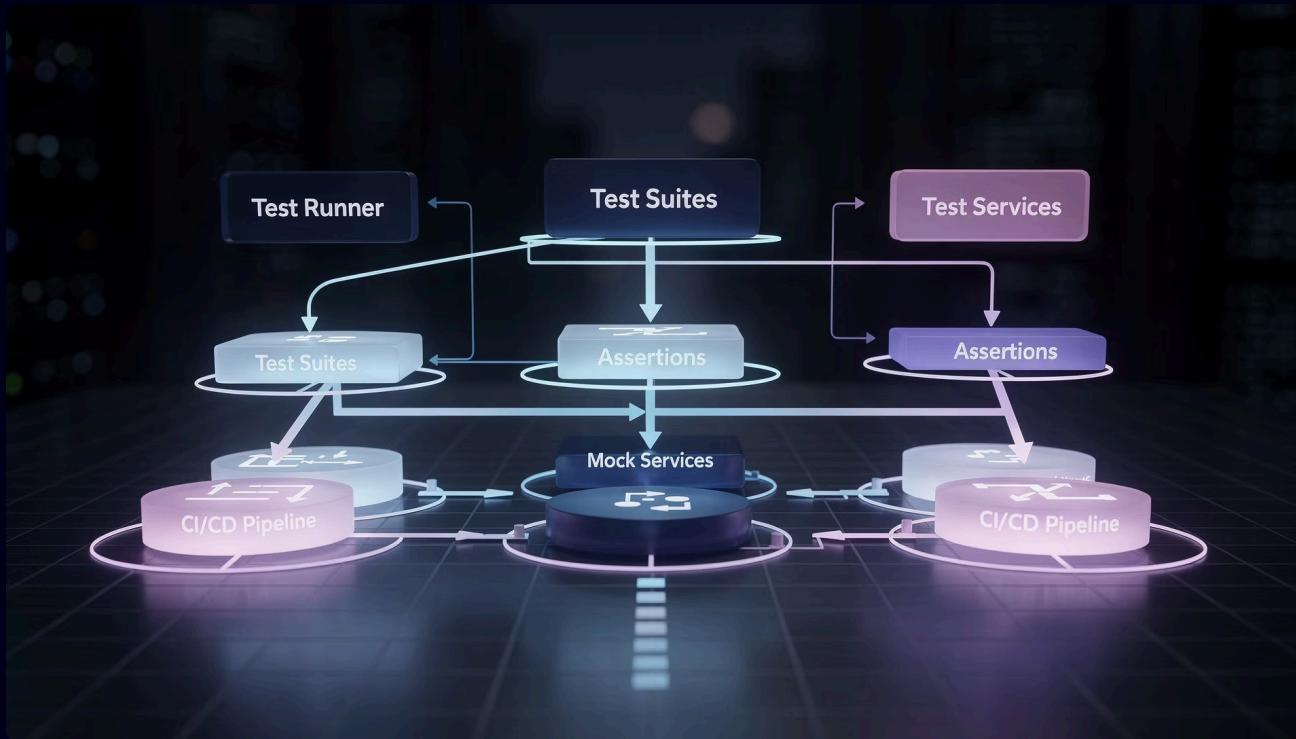


# Browser, Context та Page. UI mode

Playwright надає потужну архітектуру для автоматизації браузерів, яка базується на трьох ключових концепціях: Browser, Context та Page. Розуміння цієї ієархії є критично важливим для ефективної розробки тестів та оптимального використання ресурсів.



# Концепція фікстур тестів у Playwright



Playwright Test базується на концепції фікстур тестів – елегантному підході до організації тестового середовища. Фікстури тестів використовуються для налаштування середовища для кожного тесту, надаючи йому все необхідне і нічого зайвого. Це забезпечує чистоту та ізоляцію кожного тестового сценарію.

Фікстури тестів ізольовані між тестами, що гарантує відсутність побічних ефектів та непередбачуваної поведінки. За допомогою фікстур ви можете групувати тести на основі їхнього сенсу, а не загального налаштування середовища, що робить тестові сюїти більш логічними та зрозумілими.

# Вбудовані фікстури Playwright

Ви вже використовували фікстури тестів у своєму першому тесті, можливо навіть не усвідомлюючи всю потужність цього механізму. Розглянемо базовий приклад, який демонструє використання найважливішої фікстури – page:

```
import { test, expect } from '@playwright/test';

test('basic test', async ({ page }) => {
  await page.goto('https://playwright.dev/');
  await expect(page).toHaveTitle(/Playwright/);
});
```

Аргумент `{ page }` у функції тесту повідомляє Playwright Test налаштувати фікстуру `page` та надати її вашій функції тесту. Це приклад `dependency injection` – Playwright автоматично створює та ініціалізує об'єкт `page` перед виконанням тесту, а потім коректно очищує ресурси після його завершення.

Використання деструктуризації в параметрах функції дозволяє явно вказати, які саме фікстури потрібні конкретному тесту, що робить код більш читабельним та підтримуваним.

# Найважливіші будовані фікстури

## page

Ізольована сторінка для цього запуску тесту. Кожен тест отримує власний екземпляр page, що забезпечує повну ізоляцію.

## context

Ізольований контекст для цього запуску тесту. Фікстура page також належить до цього контексту, успадковуючи його налаштування.

## browser

Браузери діляться між тестами для оптимізації ресурсів. Це підвищує швидкість виконання тестових сюїт.

## request

Ізольований екземпляр APIRequestContext для цього запуску тесту, який використовується для надсилання API-запитів і використовує той самий context що і page.

## Ієархія фікстур

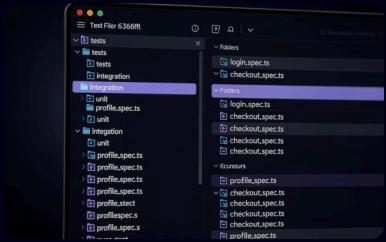
Важливо розуміти ієархічні відносини між фікстурами. Browser знаходиться на найвищому рівні і може містити кілька Context, кожен з яких може мати кілька Page. Ця архітектура забезпечує гнучкість у тестуванні різних сценаріїв.

## Оптимізація ресурсів

Playwright розумно управляє життєвим циклом фікстур: browser створюється один раз для множини тестів, context та page створюються для кожного тесту окремо, забезпечуючи баланс між продуктивністю та ізоляцією.

# UI Mode – візуальний режим виконання тестів

Щоб запустити тести з підтримкою графічного інтерфейсу виконайте команду `prx playwright test --ui`. Цей режим відкриває нові можливості для інтерактивної розробки та налагодження тестів.



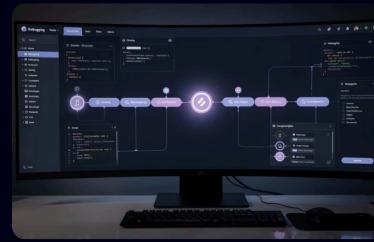
## Перегляд тестів

Після запуску режиму інтерфейсу ви побачите список усіх ваших тестових файлів у зручній деревоподібній структурі.



## Виконання тестів

Ви можете виконати всі свої тести, натиснувши іконку трикутника на бічній панелі, або запустити окремі тести вибірково.



## Налагодження

UI mode надає потужні інструменти для покрокового виконання, перегляду стану сторінки та аналізу помилок у реальному часі.

Ви також можете запустити окремий тестовий файл, блок тестів або один тест, навівши вказівник миші на назву та клацнувши трикутник поруч із ним. UI mode надає дуже багато можливостей про які можна прочитати в офіційній документації Playwright.

# Підсумок

Playwright – це потужний фреймворк для автоматизації тестування веб-додатків, який дозволяє вам створювати та запускати тести у різних браузерах. Він пропонує зручний інтерфейс для автоматизації дій у браузері та надає різні функціональності для забезпечення стабільності тестів.

Playwright підтримує роботу з Chromium, WebKit та Firefox, а також надає можливість працювати з різними мовами програмування, такими як TypeScript, JavaScript, Python, .NET, Java. Його особливості включають автоматичне очікування елементів для виконання дій, гнучкість у налаштуванні стратегії перевірки, можливість запису дій для генерації тестів.



## Стабільність

Автоматичне очікування елементів та розумні механізми ретраїв забезпечують надійність тестів



## Гнучкість

Підтримка множини браузерів та мов програмування для різноманітних сценаріїв тестування



## Інструментарій

Інспектор для візуального аналізу, генератор тестів та UI mode для зручної розробки