Playwright Full Cheat Sheet: UI Assertions & API Integration

🎯Installation

```
# Initialize Node.js project
npm init -y

# Install Playwright
npm install -D @playwright/test

# Install browser binaries
npx playwright install
```

🧨UI Assertions

```
const { test, expect } = require('@playwright/test');

test('UI assertions example', async ({ page }) => {
  await page.goto('https://example.com');

  // Element visibility
  await expect(page.locator('h1')).toBeVisible();

  // Element text
  await expect(page.locator('h1')).toHaveText('Example Domain');

  // Input value
  await page.fill('input[name="username"]', 'myuser');
  await expect(page.locator('input[name="username"]')).toHaveValue('myuser');

  // Button state
  await expect(page.locator('button#submit')).toBeEnabled();

  // Count elements
  await expect(page.locator('ul li')).toHaveCount(3);
});
```

Advanced Assertions

```
await expect(page.locator('h1')).toHaveText(/Example/);
await expect(page.locator('h1')).not.toBeEmpty();
await expect(page.locator('text=Loaded')).toBeVisible({ timeout: 5000 });
await expect(page).toHaveURL('https://example.com');
await expect(page).toHaveTitle(/Example Domain/);
```

API Integration

```
const { test, expect } = require('@playwright/test');

test('API testing example', async ({ request }) => {
  const response = await request.get('https://jsonplaceholder.typicode.com/posts/1');

  // Status code
  expect(response.status()).toBe(200);

  // JSON response
  const data = await response.json();
  expect(data.id).toBe(1);
  expect(data.userId).toBe(1);

  // Header validation
  expect(response.headers()['content-type']).toContain('application/json');
});
```

POST, PUT, DELETE

```
// POST
const postResponse = await request.post('https://jsonplaceholder.typicode.com/posts', {
  data: { title: 'foo', body: 'bar', userId: 1 }
});
expect(postResponse.status()).toBe(201);

// PUT
const putResponse = await request.put('https://jsonplaceholder.typicode.com/posts/1', {
  data: { title: 'updated title' }
});
expect(putResponse.status()).toBe(200);
```

```
// DELETE
const deleteResponse = await request.delete('https://
jsonplaceholder.typicode.com/posts/1');
expect(deleteResponse.status()).toBe(200);
```

UI + API Integration

```
test('UI reflects API data', async ({ page, request }) => {
  const apiResponse = await request.get('https://jsonplaceholder.typicode.com/
posts/1');
  const data = await apiResponse.json();

  await page.goto('https://example.com/posts/1');

  // Assert UI displays API data
  await expect(page.locator('h1')).toHaveText(data.title);
  await expect(page.locator('p')).toHaveText(data.body);
});
```

Common Commands

```
# Run all tests
npx playwright test

# Run tests in a specific browser
npx playwright test --project=chromium

# Run tests with reporters
npx playwright test --reporter=list,json,allure-playwright

# Generate code from browser interactions
npx playwright codegen
```

Tutorial References

- **JavaScript Tutorial**: Playlist
- **TypeScript Tutorial**: Playlist
- **Playwright Tutorial**: Playlist
- Advanced Playwright
- Playwright JS

- **Jenkins Tutorial**: Link
- **AWS Cloud Tutorial**: Link

Cheat sheet by Swati Salunkhe – Playwright UI & API Integration Quick Reference