4.Videos Screenshots on Test Failure

**Introduction**

Playwright allows you to record videos during test execution, which is especially helpful for debugging failed tests. By recording videos of tests that fail, you can see the complete test flow and understand what went wrong in the context of dynamic web pages or complex user interactions.

* **Feature**: Automatically record videos when a test fails.
  + Captures the entire user interaction flow, not just the final state (like screenshots).
  + Can be configured globally or per test.
  + Supports different video formats and quality settings.

**Configuring Videos on Failure**

You can configure Playwright to capture videos when tests fail by setting the video option in the Playwright configuration file (playwright.config.ts).

* **Global Configuration**: Enable video recording for all tests, capturing video only on failures.

**Config Example**:

typescript

Copy code

import { defineConfig } from '@playwright/test';

export default defineConfig({

use: {

video: 'retain-on-failure',

},

});

* **Video Options**:
  + off: Disables video recording.
  + on: Records videos for every test.
  + retain-on-failure: Records videos but retains them only if the test fails.
  + retain-on-all: Keeps the video even for passing tests (useful for auditing or full test documentation).

**Test-Level Configuration**

You can also configure video recording for individual tests using the use option inside the test definition. This is useful when you want to record videos only for specific tests.

* **Per-Test Configuration**: Specify video recording at the test level.

**Test Example**:

typescript

Copy code

import { test } from '@playwright/test';

test.use({ video: 'retain-on-failure' });

test('example failing test with video', async ({ page }) => {

await page.goto('https://playwright.dev');

// Intentionally cause a failure

await expect(page).toHaveTitle('Wrong Title');

});

* **Use Case**: Record videos for critical or complex tests where a failure might require deeper investigation.

**Where Videos Are Saved**

When a test fails and video recording is enabled, Playwright saves the recorded video in the test-results folder by default. The directory structure will reflect the test name and browser used, just like with screenshots.

* **Directory Structure**:
  + Each test run creates a unique folder containing video recordings and logs.
  + Videos are named based on the test file and browser.

**Example Folder Structure**:

markdown

Copy code

test-results/

└── example.spec.ts/

└── example-test-1-chromium-failure.webm

* **Custom Video Path**: You can customize where the videos are saved by setting the videosDir option in the Playwright configuration.

**Config Example**:

typescript

Copy code

export default defineConfig({

use: {

video: 'retain-on-failure',

videosDir: 'videos', // Custom directory for videos

},

});

**Example: Recording Videos on Test Failure**

Here’s an example test that demonstrates how Playwright records a video when the test fails.

**Test Example**:

typescript

Copy code

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

test('failing test with video', async ({ page }) => {

await page.goto('https://playwright.dev');

// This assertion will fail

await expect(page).toHaveTitle('Nonexistent Title');

});

* **What Happens**:
  + Playwright will record a video of the entire test execution.
  + If the test fails, the video will be saved in the test-results folder (or custom path, if specified).
  + You can review the video to trace back the exact steps leading up to the failure.

**Video Quality and Size Configuration**

Playwright allows you to customize the quality and size of the recorded videos for efficient storage and better clarity.

* **Quality Configuration**: Set the video size, codec, and quality in the Playwright config file.

**Config Example**:

typescript

Copy code

export default defineConfig({

use: {

video: 'retain-on-failure',

viewport: { width: 1280, height: 720 }, // Set resolution for video recording

},

});

* **Use Case**: Control the file size and clarity of the videos, especially when running tests in parallel or storing the video logs for longer periods.

**Analyzing Videos**

Once the video is captured, you can watch the entire test flow and spot UI interactions, rendering issues, or any unexpected behaviors.

* **How to Analyze**:
  + Review the test video to understand the exact point where the failure occurred.
  + Compare UI behaviors across different browsers or environments using the recorded videos.
  + Videos are saved in .webm format, compatible with most media players and browsers.

**Conclusion**

Recording videos on test failures in Playwright is a highly valuable debugging tool, especially for complex UI interactions and dynamic content. By configuring video recording globally or per test, you can ensure that critical information is captured, aiding in faster issue resolution. Whether you're analyzing full-page flows or specific test cases, Playwright's video recording capabilities provide in-depth insight into test failures.