<https://playwright.dev/>

Installation

Node.js

npm

npm init playwright@latest

View -- > Extensions 🡪 Playwright Test for VS code

<https://playwright.dev/docs/test-cli>

npx playwright test

npx playwright test tests/todo-page.spec.ts

npx playwright test my-spec my-spec-2

npx playwright test --grep "has title"

npx playwright test –-headed

npx playwright test –-debug

npx playwright test --ui

npx playwright show-report

npx playwright codegen test2

<https://playwright.dev/docs/test-annotations>

test.only *// Run only focused tests in the entire project.*

test.skip*/ This test is not run*

test('Test login page @fast') //Tags

test('Test login page @smoke) //Tags

npx playwright test --grep "@smoke"

npx playwright test --grep-invert "@smoke"

npx playwright test --grep "@fast||@slow"

// Test Suite and set of Tests

test.describe('two tests', () => {  
 test('one', async ({ page }) => {  
 *// ...*  
 });  
  
 test('two', async ({ page }) => {  
 *// ...*  
 });  
});

Test configuration

<https://playwright.dev/docs/test-configuration>

*// Look for test files in the "tests" directory, relative to this configuration file.*  
 testDir: 'tests',

use: {  
 *// Capture screenshot after each test failure.*  
 screenshot: 'only-on-failure',  
  
 *// Record trace only when retrying a test for the first time.*  
 trace: 'on-first-retry',  
  
 *// Record video only when retrying a test for the first time.*  
 video: 'on-first-retry'  
 },

use: {  
 *// Maximum time each action such as `click()` can take. Defaults to 0 (no limit).*  
 actionTimeout: 0,  
  
 *// Name of the browser that runs tests. For example `chromium`, `firefox`, `webkit`.*  
 browserName: 'chromium',  
  
 *// Toggles bypassing Content-Security-Policy.*  
 bypassCSP: true,  
  
 *// Channel to use, for example "chrome", "chrome-beta", "msedge", "msedge-beta".*  
 channel: 'chrome',  
  
 *// Run browser in headless mode.*  
 headless: false,  
  
 *// Change the default data-testid attribute.*  
 testIdAttribute: 'pw-test-id',  
 },

use: {  
 launchOptions: {  
 slowMo: 50,  
 },

**Emulation**

With Playwright you can test your app on any browser as well as emulate a real device such as a mobile phone or tablet

export default defineConfig({  
 projects: [  
 {  
 name: 'chromium',  
 use: {  
 ...devices['Desktop Chrome'],  
 },  
 },  
 {  
 name: 'Mobile Safari',  
 use: {  
 ...devices['iPhone 13'],  
 },  
 },  
 ],  
});

**Fixtures**

Here is a list of the pre-defined fixtures that you are likely to use most of the time:

| **Fixture** | **Type** | **Description** |
| --- | --- | --- |
| page | [Page](https://playwright.dev/docs/api/class-page) | Isolated page for this test run. |
| context | [BrowserContext](https://playwright.dev/docs/api/class-browsercontext) | Isolated context for this test run. The page fixture belongs to this context as well. Learn how to [configure context](https://playwright.dev/docs/test-configuration). |
| browser | [Browser](https://playwright.dev/docs/api/class-browser) | Browsers are shared across tests to optimize resources. Learn how to [configure browser](https://playwright.dev/docs/test-configuration). |
| browserName | [string](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#String_type) | The name of the browser currently running the test. Either chromium, firefox or webkit. |
| request | [APIRequestContext](https://playwright.dev/docs/api/class-apirequestcontext) | Isolated [APIRequestContext](https://playwright.dev/docs/api/class-apirequestcontext) instance for this test run. |

**Locators**

[Locator](https://playwright.dev/docs/api/class-locator)s are the central piece of Playwright's auto-waiting and retry-ability. In a nutshell, locators represent a way to find element(s) on the page at any moment.