Best Practices

Einige Tipps, die Playwright erleichtern

Einführung

Hier sollen einige Tipps und Tricks gezeigt werden die den Umgang mit Playwright erleichtern sollen.

awesome Playwright

Auch Playwright besitzt ein grosses Netzwerk und viele zusätzliche Tools.
Die meisten die einen Namen haben, sind hier gelistet:

https://github.com/mxschmitt/awesome-playwright

Unter anderem findet man auch Ports auf Sprachen, die offiziell oder nicht offiziell supportet werden zB. Rust.

9 Gute Tipps

1. Im Voraus sicher sein, was und wie viel man testen möchte. Darüber haben wir uns schon unterhalten.

Darauf "lostesten" führt in seltenen Fällen zum Ziel und ist meist teurer.

9 Gute Tipps

2. Verwendung stabiler Selektoren zum Auffinden von Elementen.

Bevor wir das Verhalten unserer Webanwendung testen können, müssen wir Elemente auf der **`Seite finden und Aktionen mit diesen durchführen`**.

Playwright empfiehlt die Verwendung der eingebauten **`Locators`**.

9 Gute Tipps

3. Tests fokussieren und isolieren

Playwright-Tests sind so konzipiert, dass sie in isolierten Umgebungen ausgeführt werden, so dass jeder Test seinen eigenen lokalen Speicher, Sitzungsspeicher, Cookies usw. hat. Diese Isolierung garantiert, dass die Tests nicht durch die Ergebnisse oder Nebenwirkungen anderer Tests beeinträchtigt werden, was unabhängige und zuverlässige Testergebnisse fördert.

9 Gute Tipps

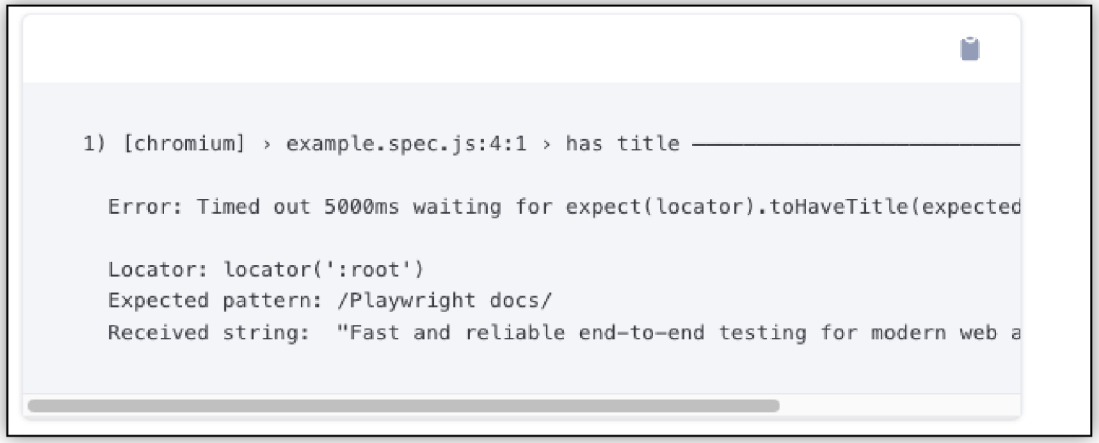
4. End-User Perspektive

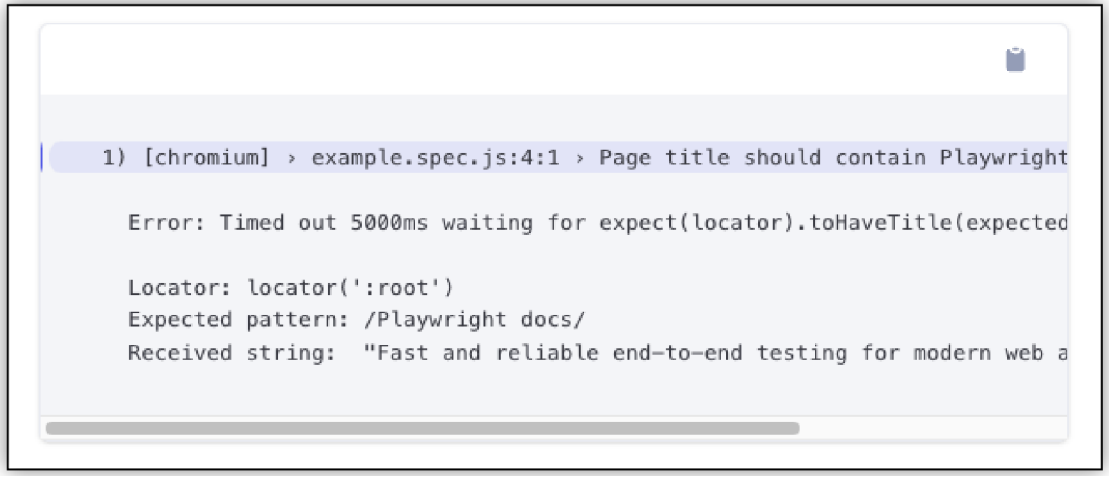
Aussagekräftige Assertions sind solche, die die Benutzerinteraktionen und -erwartungen innerhalb Ihrer Anwendung genau nachbilden. Dabei geht es nicht nur darum zu prüfen, ob ein Element vorhanden ist, sondern auch darum, ob das Verhalten der Anwendung dem entspricht, was ein Benutzer erwarten oder tun würde. Dies beinhaltet:

9 Gute Tipps

5. Beschreibende Test- und Schritttitel, um die Absicht zu verdeutlichen.

Stell dir vor, es ist schon spät am Tag und Sie sind gerade dabei, mehrere Stunden Refactoring-Arbeiten abzuschliessen. Alles scheint in Ordnung zu sein, aber dann schlägt ein Test fehl:





9 Gute Tipps

6. Alle relevanten Browser

Playwright vereinfacht das browserübergreifende Testen auf jeder Plattform, um sicherzustellen, dass die Anwendung für alle Benutzer korrekt funktioniert.

In der Konfigurationsdatei kann man Projekte einrichten, indem man deren Namen zusammen mit dem gewünschten Browser oder Gerät angibt.

9 Gute Tipps

7. Automatisieren und überwachen

Für einen stabilen Entwicklungszyklus reicht es nicht aus, Tests nur in der lokalen Umgebung auszuführen.

Man sollte diese auch in die CI/CD-Prozesse integrieren, damit diese während des Build-Steps überwacht werden können.

9 Gute Tipps

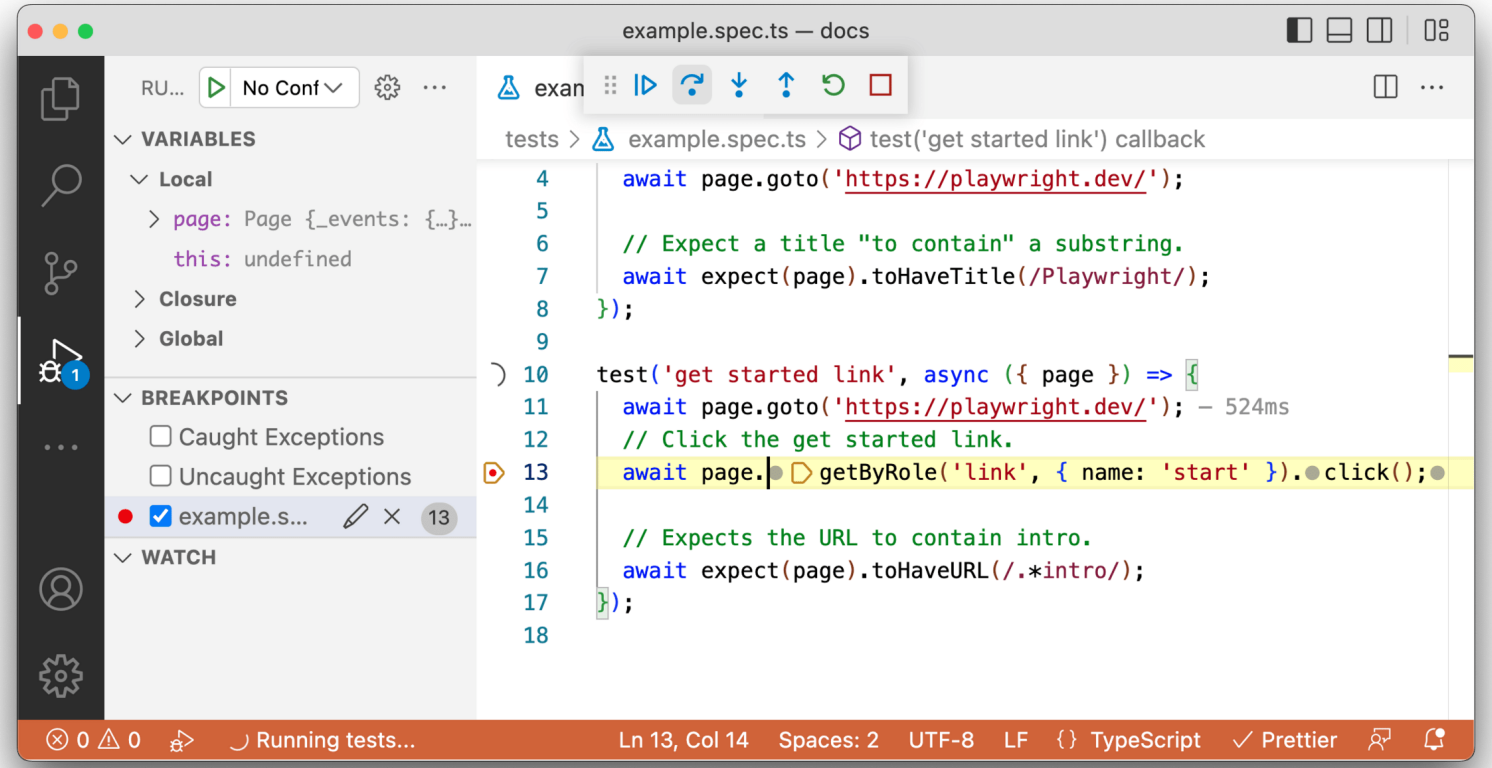
8. Typescript tooling

Inspector

UI-Mode

Trace viewer

Code generator



Sind da, um verwendet zu werden.

9 Gute Tipps

9. Keine Integrationen von Drittanbietern testen

Es ist üblich, dass Webanwendungen von APIs von Drittanbietern abhängig sind.

Deren Integration in Ihre End-to-End-Tests (E2E) kann jedoch Probleme mit sich bringen, z. B. unvorhersehbare Antwortzeiten, Ratenbeschränkungen und zusätzliche Kosten.

Diese Faktoren können Ihre Tests verlangsamen und zu zeitweiligen Fehlern aufgrund von Netzwerkinkonsistenzen führen, wodurch eine unzuverlässige Testumgebung entsteht.

Ende

Das war alles für dieses Kapitel