ToiletPaper #97

Liebling, ich habe die Frontend-Tests automatisiert!

Von Attila von Thermann / Software Architect / Business Division New Business

✖ Problem

Ihr kennt das sicher: Ihr baut ein Webfrontend mit vielen Eingabefeldern und kleinen Funktionalitäts-Details. Da ist es nicht nur beim Entwickeln mühsam, zu überprüfen, ob ein Feature richtig umgesetzt ist - auch können sich leicht Bugs einschleichen. Man überprüft ja nicht nach jedem Deployment jedes vorhandene Feature.

✔ Lösung

Die Lösung sind, wie im Backend, automatisierte Tests. Je nach Technologie arbeiten die mitgelieferten Tests meist auf Komponenten-Ebene. In der Test-Pyramide wäre das die unterste Schicht. Für Business-Features ist die oberste Schicht eine wichtige Ergänzung: UI- bzw. End-2-End-Tests kann man gut mit einer Kombination aus Selenium und Cucumber umsetzen. Selenium nimmt Aktionen entgegen und vermittelt sie an einen Browser. Mit dem Selenium-Driver lassen sich verschiedene Eigenschaften abfragen, z.B. ob ein Element sichtbar ist. Selenium-basierte Tests können in verschiedenen Sprachen geschrieben werden und lassen sich auch an Browserstack anbinden, wo sie mit einer Auswahl an OS-/Browser-Kombinationen ausgeführt werden können.
Cucumber lässt Test-Szenarios in "Gherkin" formulieren, einer "business-readable language", für die keine Programmierkenntnisse erforderlich sind.
E2E-Tests lassen sich im Jenkins-Build integrieren, wenn man einen Headless-Browser benutzt oder auf Jenkins ein virtuelles Display mit "Xvfb" einrichtet. Wir haben für unsere Tests "selenium-cucumber-js" benutzt, das Selenium und Cucumber bündelt. Beim Durchlaufen der E2E-Tests wird auch ein Testreport im HTML-Format, mit Tortendiagrammen und Screenshots, generiert.

➔ Beispiel

Hier testen wir, dass man auf der jambit-Webseite, wenn man den Button der Cookie-Warnung klickt, danach das Hauptbild sieht:

Feature-Datei in Gherkin:

```
# Feature: Freitext
Feature: jambit landing page
  # Scenario: Freitext
  Scenario: Visiting jambit.com
    # Testschritte: werden mit regex-
    # Ausdrücken in Step Definition
    # gematcht
    Given I browse to "www.jambit.com"
    When I click the cookie-OK button
    Then I should see the header image
```

Step-Definition-Datei in Javascript:

```
module.exports = function() {
  this.Given(/^I browse to "([^"]*)"$/, url => {
    return helpers.loadPage(url);
  });
  this.When(/^I click the cookie-OK button$/, () => {
    return driver.findElements(by.id("cookie-ok-button")).then(button => {
      button.click();
    });
  });
  this.Then(/^I should see the header image$/, () => {
    return driver
      .findElements(by.id("header-image"))
      .then(headerImage => headerImage.isDisplayed())
      .then(result => {
        expect(result).to.equal(true);
      });
  });
};
```

+ Weiterführende Aspekte

- https://martinfowler.com/bliki/TestPyramid.html (Test-Pyramide von Martin Fowler)
- https://www.quora.com/Why-would-you-write-end-to-end-tests-with-Selenium-if-you-can-use-Puppeteer (Vergleich Puppeteer / Selenium)
- https://docs.seleniumhq.org/ (Selenium)
- https://docs.cucumber.io/gherkin/reference/ (Gherkin)
- https://www.browserstack.com/ (Browserstack)
- https://github.com/john-doherty/selenium-cucumber-js (selenium-cucumber-js)
- https://www.npmjs.com/package/cucumber-html-reporter (Cucumber HTML Reporter, wird von "selenium-cucumber-js" genutzt)