ToiletPaper #97

Honey, I've automated the frontend tests!

Author: Attila von Thermann / Software Architect / Business Division New Business

✖ Problem

You probably know that: You build a web frontend with many input fields and small functionality details. It is not only difficult to develop and to check whether a feature is implemented correctly, but also bugs can occur quickly. Eventually, you do not check every existing feature after every deployment.

✔ Solution

The solution is automated testing – just as in the backend. Depending on the technology, the provided tests usually work at the component level. In the test pyramid this would be the bottom layer. For business features, the top layer is an important addition: UI or End-2-End Tests can be easily implemented with a combination of Selenium and Cucumber.

Selenium accepts action commands and conveys them to a browser. The Selenium driver can be used to query various features, e.g. whether an element is visible or not. Selenium-based tests can be written in different languages and can also be linked to BrowserStack, where they can run with a selection of OS / browser combinations.
Cucumber has test scenarios formulated in "Gherkin", a "business-readable language" that does not require any programming skills.

E2E Tests can be integrated into Jenkins build when using a headless browser or setting up a virtual display with "Xvfb" on Jenkins. For our tests, we used selenium-cucumber-js, which combines Selenium and Cucumber. Carrying out the E2E tests, a test report with pie charts and screenshots is generated in HTML format.

➔ Example

We are testing on the jambit website – if you click the cookie-OK button, you will see the header image:

Feature File in Gherkin:

```
# Feature: Freetext
Feature: jambit landing page
  # Scenario: Freetext
  Scenario: Visiting jambit.com
    # Test steps: will be matched with
    # regex terms in Step Definition
    Given I browse to "www.jambit.com"
    When I click the cookie-OK button
    Then I should see the header image
```

Step Definition File 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);
      });
  });
};
```

✚ Further Aspects

- https://martinfowler.com/bliki/TestPyramid.html (Test pyramid by Martin Fowler)
- https://www.quora.com/Why-would-you-write-end-to-end-tests-with-Selenium-if-you-can-use-Puppeteer (Comparison 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 – used by "selenium-cucumber-js")