Chandrawathanan Anujan

CEMBRA MONEY BANK | Bändliweg 20, 8048 Zürich

Testautomation Webservices

IPA 2020

Table of Contents

[1. Umfeld und Ablauf 2](#_Toc39833041)

[1.1 Ausgangslage 2](#_Toc39833042)

[1.2 Detaillierte Aufgabenstellung 2](#_Toc39833043)

[1.3 Mittel und Methoden 3](#_Toc39833044)

[1.4 Vorkenntnisse 3](#_Toc39833045)

[1.5 Vorarbeiten 3](#_Toc39833046)

[1.6 Neue Lerninhalte 3](#_Toc39833047)

[1.7 Arbeiten in den letzten 6 Monaten 4](#_Toc39833048)

[2. Projekt 4](#_Toc39833049)

[2.1 Informieren 4](#_Toc39833050)

[2.2 Planen 4](#_Toc39833051)

[2.3 Entscheiden 4](#_Toc39833052)

[2.4 Realisieren 4](#_Toc39833053)

[2.5 Kontrollieren 4](#_Toc39833054)

[2.6 Auswerten 4](#_Toc39833055)

# Umfeld und Ablauf

## Ausgangslage

Cembra testet neue Webapplikationen aktuell manuell. Dazu werden Benutzereingaben vorgegeben, die händisch in die Applikation (üblicherweise ein Formular) eingetippt werden. Die vorgegebenen Daten unterscheiden sich dabei oft nur marginal, z. B. wechselt der Wohnkanton. Ausserdem werden bewusst falsche oder fehlende Benutzereingaben simuliert, um die Validierungsfunktionen der Applikation zu testen. Diese Test sind sehr zeitaufwendig und extrem repetitiv, ausserdem müssen diese Tests bei jeder Änderung der Webapplikation zumindest teilweise wiederholt werden. Die Testergebnisse werden aktuell manuell in ein Excelsheet und ein Ticketingsystem übertragen.

Ein Teil dieser Tests und der Ablage der Ergebnisse soll mit Hilfe von Selenium automatisiert werden.

## Detaillierte Aufgabenstellung

Implementierung von Selenium WebDriver for Java zur Automation der Tests neuer Webapplikationen im Cembra Internet, Programmieren der nötigen Java Classes und Konfigurieren mindestens eines "real life" Testcases für eine vorhandene Cembra Webapplikation. "Testcases" beinhalten einen oder mehrere Satz von Eingabedaten für die Webappplikation, beispielsweise "Familie mit 2 Kindern, deutsch, Zusatzeinkommen ja / nein, usw. Testcases sind vordefiniert und müssen 1:1 abgearbeitet werden. Für jeden Schritt innerhalb eines Testcases existiert ein erwartetes Ergebnis (z. B. das Anzeigen und die Validierung des Eingabefelds für Zusatzeinkommen im "ja" Fall). Die Lösung muss in mindestens einem der bankintern genutzten Browser funktionieren (MS Edge, MS IE 11, Chrome, Firefox).

Selenium soll genutzt werden, um eine bereits bestehende, aber nicht veröffentlichte Applikation zu testen. Die Applikation ist im Prrinzip ein sehr langes Formular, dessen Eingabefelder aber nicht vollständig normalen HTML Input-Feldern entsprechen. Die Applikation ist in Javascript (React) als Single Page Application geschrieben und die Formularfelder basieren auf selbst entwickelten React Komponenten. Das Verhalten dieser Komponenten ist abhängig von Benutzereingaben und Regeln (z. B. bestimmt die Sprachauswahl bestimmte Formatierungen). Zur Navigation innerhalb der Applikation (in einer normalen Webseite entspricht das dem Aufruf einer neuen Seite) wird React Router verwendet, es wird beim "Seitenwechsel" daher kein GET Request an den Webserver geschickt, sondern einfach die URL in der Adresszeile des Browsers umgeschrieben. Statt eine neue Seite zu laden, wird die aktuelle Seite basierend auf den Regeln, die React Router für diese "Seite" liefert, neu gerendert. Validierungen (z. B. auf fehlende oder ungültige Angaben) und Formatierungen (z. B. Regeln für die korrekte Darstellung bestimmter Angaben wie Währungen) werden zudem nicht in der "Seite", sondern einem sog. "Store" ausgeführt (in etwa vergleichbar mit dem Aufruf einer Klasse in OOP).

Die von Selenium als Referenz zur Verfügung gestellten Beispiele beziehen sich auf klassische Webseiten mit "normalen" HTML Inputs und einer normalen Navigation. Die Aufgabe ist, diese Beispiele für SPA und die Cembra-spezifischen Eingabefelder anzupassen. Es sollen keine neuen Klassen entwickelt werden, sondern die von Selenium bereitgestellten Beispiele sinnvoll erweitert bzw. geändert werden. Ziel ist eine Referenzimplementierung, die von Cembra beliebig erweitert bzw. für andere Testfälle neu konfiguriert werden kann. Geschätzter Aufwand: 3 Arbeitstage

Die Lösung soll die ersten 5-6 (von insgesamt 15) "Seiten" der Applikation anhand vorgegebener Kriterien testen können, auf den zu testenden Seiten tauchen alle, für die Applikation relevanten, Eingabefelder mindestens einmal auf. Die Kriterien für den Test entsprechen dabei denen, die zur Zeit manuell durchgeführt werdenuellen Test (z. B. Tests auf falsche oder unvollständige Eingaben und die Validierung der Fehlermeldung, Verhalten beim Sprachwechsel usw.). Selenium muss so konfiguriert werden und es müssen alle nötigen Javaklassen verfügbar sein, damit ein Testcase für diesen Teil der Applikation vollständig durchlaufen werden kann. Jedes von der Vorgabe des Testcases abweichende Verhalten der Applikation muss mit einem aussagekräftigen Fehler und einem Screenshot der Applikation dokumentiert werden. Dieser Test muss in Selenium demonstriert werden. Geschätzter Aufwand: 2 Arbeitstage

Die Anpassungen in den Javaklassen müssen ausreichend dokumentiert werden (Kommentare in den Klassen sind ausreichend, sofern sie entsprechend aussagekräftig sind). Die Dokumentation muss so umfangreich sein, dass ein nicht mit dem Projekt befasster Programmierer die Änderungen nachvollziehen kann. Die Konfiguration der Tests in Selenium muss reproduzierbar sein, für jeden Schritt des Testcases muss eine entsprechende Konfigurationsanweisung dokumentiert werden. Dieser Teil der Dokumentation ist nötig, um die Tests formal abschliessen zu können, d. h., es muss eine vollständige und reproduzierbare Beschreibung der verwendeten Parameter vorliegen. Die Dokumentaton der Parameter benötigt keine Erläuterungen, es sei denn, sie weicht aus irgendwelchen Gründen von Vrogaben der Selenium-Dokumentation ab. Dieser Teil der Dokumentation richtet sich ausdrücklich an Personen, die Erfahrung mit der Konfiguration von Selenium haben. Geschätzter Aufwand: ca. 1 Arbeitstag

## Mittel und Methoden

Selenium WebDriver  
Java 8   
IntelliJ   
Browser   
Git

## Vorkenntnisse

Java  
Git  
IntelliJ

## Vorarbeiten

Einarbeiten in Selenium WebDriver

## Neue Lerninhalte

Selenium WebDriver  
Selenium Dokumentation  
Google

## Arbeiten in den letzten 6 Monaten

Software Pakete erstellt  
Skripts für Active Directory und Windows 10 Migration

# Projekt

## Informieren

## Planen

## Entscheiden

## Realisieren

## Kontrollieren

## Auswerten