

Vertrauen ist gut, Kontrolle ist besser! APEX Applikationen automatisiert testen



Vorstellung

- Henner Hucke
- Dipl. Ing. agr. an der Martin Luther Universität Halle
- seit 1994 Tätigkeitsschwerpunkt Datenbanken
- seit 1999 im Oracle Umfeld tätig
- Oracle RDBMS von 7.3 – 12c
- Java, C, SAP, ...
- Aktuell angestellt bei DATAGROUP Enterprise Services GmbH

DATAGROUP im Überblick



- DATAGROUP ist einer der führenden IT-Service-Anbieter Deutschlands*
- Unsere Kernkompetenz: We manage IT
- Wir betreuen über 600.000 IT-Arbeitsplätze weltweit
- Standorte in allen wichtigen Wirtschaftsregionen Deutschlands
- Rund 1.900 Mitarbeiter, davon rund 60 Azubis
- Wir denken anders.
Wir arbeiten anders.
Wir finden bessere Lösungen.

Vertrauen ist gut, Kontrolle ist besser!

- Zitat kommt angeblich von Lenin
- Erweiterung bestehender APEX Applikation beim Kunden
- neu zusammengestelltes, räumlich getrenntes Projektteam
- „What one programmer can do in one month, two programmers can do in two months.” (Frederick P. Brooks)

APEX Applikationen automatisiert testen

- Akzeptanztest der APEX-Applikation
- Unit-Test der PL/SQL Funktionalität mit z.B. utPLSQL

Anforderungen

Automatische Aufzeichnung der Testfälle (Recorder)

Bedingungen für Erfolg/Misserfolg müssen definiert werden können

Screenshots müssen sich erstellen lassen

definierte Testumgebung muss sich automatisiert erstellen lassen

Tests sollten über Jobsteuerung getriggert werden können

Testwerkzeuge

- Ghost Inspector (<http://exemacro.com/oracle-apex-testing/>)
- Selenium IDE (Firefox Plugin)
- APEX test automation framework (ATAF)
- Katalon Studio
- Katalon Recorder (Firefox Plugin)
- Selenium WebDriver mit Python

Testfälle

- Anmeldung an Applikation
- Anzeige eines Reports und Inhaltsprüfung
- Erstellen eines neuen Datensatzes

Selenium IDE

- DEMO



Selenium IDE

- einfach zu installieren
- intuitiv zu benutzen
- sehr gut geeignet um interaktiv eine Anzahl von Testfällen zu prüfen
- sehr gut geeignet um Fehler in der Anwendung mit dem Entwickler zu kommunizieren
- relative wenig Nacharbeit beim Aufzeichnen der Testcases

Selenium IDE

- Resultat

Anforderungen

Automatische Aufzeichnung der Testfälle (Recorder)



Bedingungen für Erfolg/Misserfolg müssen definiert werden können



Screenshots müssen sich erstellen lassen

definierte Testumgebung muss sich automatisiert erstellen lassen

Tests sollten über Jobsteuerung getriggert werden können

Katalon Recorder

- DEMO



Katalon Recorder

- einfach zu installieren
- intuitiv zu benutzen
- sehr gut geeignet um interaktiv eine Anzahl von Testfällen zu prüfen
- sehr gut geeignet um Fehler in der Anwendung mit dem Entwickler zu kommunizieren
- Befehlssatz ist viel größer als in Selenium IDE
- manchmal ist Nacharbeit beim Aufzeichnen der Testcases erforderlich

Katalon Recorder

- Resultat

Anforderungen

Automatische Aufzeichnung der Testfälle (Recorder)



Bedingungen für Erfolg/Misserfolg müssen definiert werden können



Screenshots müssen sich erstellen lassen



definierte Testumgebung muss sich automatisiert erstellen lassen

Tests sollten über Jobsteuerung getriggert werden können

Selenium WebDriver mit Python

- Voraussetzungen
 - Python (Version ≥ 3.6)
 - Python bindings for Selenium (pip install selenium)
 - GeckoDriver für die Fernsteuerung des Firefox-Browsers
 - cx_oracle Paket für die Interaktion mit der Oracle Datenbank (pip install cx_oracle)

Selenium WebDriver mit Python

- Beispiel

```
# -*- coding: utf-8 -*-
import time
from selenium import webdriver

driver = webdriver.Firefox()
driver.get("https://www.google.de/")
assert "Google" in driver.title
time.sleep(10)
driver.close()
```

Selenium WebDriver mit Python

- DEMO



Selenium WebDriver mit Python

- Python mit cx_oracle

```
import cx_Oracle

connection = cx_Oracle.connect("apxdemoadm", "geheim", "pdbapx5.vrn")
cursor = connection.cursor()
statement = "select 'Connected to Oracle' x from dual"
cursor.execute(statement)
for x in cursor:
    print("Values:", x)
cursor.close()
connection.close()
```

Selenium WebDriver mit Python

- Beispielcode Testsuite 01

imports

class Test01(unittest.TestCase):

wird einmal beim Erstellen der Instanz der Klasse aufgerufen

@classmethod

def setUpClass(cls):

cls.driver = webdriver.Firefox()

...

wird für jeden Testcase ausgeführt

def setUp(self):

...

Selenium WebDriver mit Python

- Beispielcode Testsuite 02

Testcase-01

```
def test_01(self):
    driver = self.driver
    driver.get("https://cent7.vm:8453/ords/f?p=117")
    driver.find_element_by_id("P101_USERNAME").click()
    ...
```

Testcase-02

```
def test_02(self):
    driver = self.driver
    driver.get("https://cent7.vm:8453/ords/f?p=117")
    driver.find_element_by_id("P101_USERNAME").click()
    ...
```

Selenium WebDriver mit Python

- Beispielcode Testsuite 03

...

```
def tearDown(self):
    self.assertEqual([], self.verificationErrors)
```

wird beim Abräumen der Klasse aufgerufen

```
@classmethod
def tearDownClass(cls):
    cls.driver.quit()
```

Selenium WebDriver mit Python






- DEMO



Selenium mit Python

- Resultat

Anforderungen

- | | |
|--|---|
| Automatische Aufzeichnung der Testfälle (Recorder) |  |
| Bedingungen für Erfolg/Misserfolg müssen definiert werden können |  |
| Screenshots müssen sich erstellen lassen |  |
| definierte Testumgebung muss sich automatisiert erstellen lassen |  |
| Tests sollten über Jobsteuerung getriggert werden können |  |

Stolperfallen

- IDs der Applikation unterschiedlich (DEV/UAT/PROD)
 - Alias anstelle der ID der Applikation benutzen
- systemgenerierte IDs von z.B. Buttons/Reports sind unterschiedlich
- XPath-Ausdruck anstelle des Verweise auf die systemgenerierte ID verwenden

Fazit

- automatisierte Tests lassen sich relativ einfach erstellen
- Komplexität steigt mit den Anforderungen
 1. Selenium IDE
 2. Katalon Recorder
 3. Selenium WebDriver mit Python
- Aufwand für die Erstellung der Testcases sollte nicht unterschätzt werden

Quellen

- Vergleich Katalon-Studio / Katalon Recorder
https://medium.com/@katalon_studio/katalon-recorder-3-0-9279f02d16da
- Selenium mit Python <https://selenium-python.readthedocs.io/>
- Selenium IDE (Firefox Plugin)
<https://addons.mozilla.org/de/firefox/addon/selenium-ide/>
- Katalon Recorder <https://addons.mozilla.org/en-US/firefox/addon/katalon-automation-record/?src=search>

Fragen / Anregungen



Vielen Dank für Ihre Aufmerksamkeit!

DATAGROUP Enterprise Services GmbH

Auf den Tongruben 3 ▪ 53721 Siegburg

Tel. +49 1520 165 5394 ▪ henner.hucke@datagroup.de ▪ www.datagroup.de

Sie finden uns auch auf 