Testing Smalltalk AJAX/SJAX Web Applications with Selenium 4

29th ESUG Conference
Novi Sad, Serbia
25th August 2022

Carsten Härle, straightec GmbH

Who am I?

▸ Carsten Härle: Founded straightec GmbH 2000

▸ Experience in Smalltalk since 1986

▸ IT-Consulting, Software Development, Training and Coaching for OO, eXtreme Programming, Automated Testing, TDD

▸ Shrink wrapped product „beno" in Dolphin Smalltalk for teaching and network management in Schools, Microsoft Certified Product
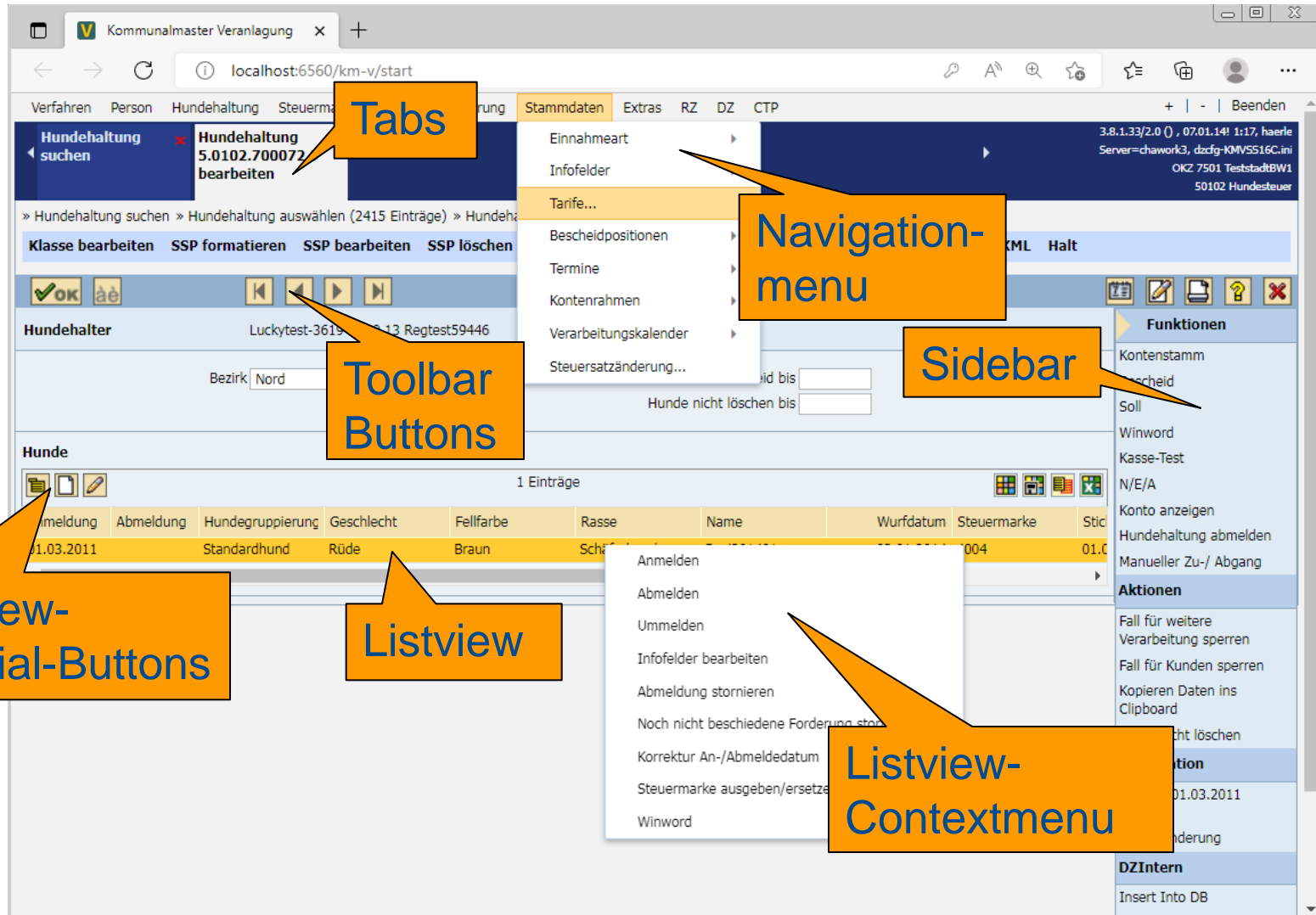
Outline

▸ Web Application

▸ Requirements for a web test framework

▸ Basis Parasol, Improvements, port to VisualWorks

▸ Simplified layer

▸ Problems and solutions

▸ Demos

▸ State and Web and Server for a test case

▸ Continuous, cooperative Build and Test, VM workers

Carsten Härle, www.straightec.de

Web application

- ▸ Project : Tax assesment of regional taxes and fees (dog tax, kindergarden, real estate tax, business taxes, water / wastewater)

- ▸ Started as ObjectStudio Fat-Client

- ▸ Now AJAX rich client web application with about 90% GUI and model code compatiblity to Fat-client system

- ▸ Libraries DHTMLX, Custom Controls, Javascript

UI-Elements

Carsten Härle, www.straightec.de

Droplist

Searchable Droplist

TreeView

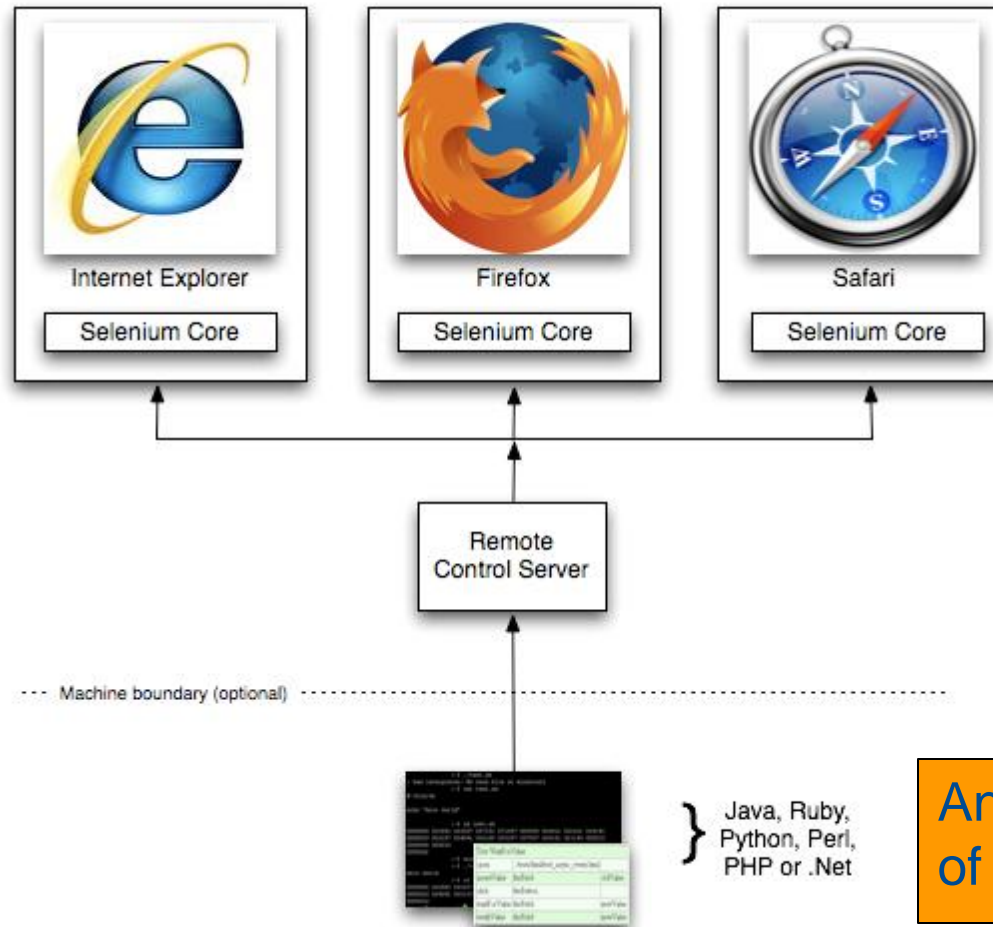Carsten Härle, www.straightec.de

Requirements for a Web Testing framework

▸ Multiple web browsers: IE (deprecated) MS-Edge, Firefox (started with IE6, Firefox 3)

▸ Completely scriptable for full automation

▸ Automatic waiting for AJAX request termination

▸ Automatic Message Box and wait cursor detection

▸ Scriptable from Smalltalk, compatible with SUnit

▸ Optimal execution times (no unneccessary waits)

▸ High level test code (no „technology noise")

▸ Easy to write, read and maintain, usable even by non-programmers

- Selenium 1.0-4.0 as the execution engine
- SUnit-Integration in Smalltalk
- Interfaces for custom detection of AJAX request termination
- Interface for custom Messagebox detection
- Extended High-Level-Command-Library for short and easy to write and maintainable code
- Integration with parallel test execution servers
- based Parasol

Selenium Remote-Control Architecture

Windows, Linux, or Mac (as appropriate)...

Internet Explorer — Selenium Core

Firefox — Selenium Core

Safari — Selenium Core

Remote Control Server

Machine boundary (optional)

Java, Ruby, Python, Perl, PHP or .Net

And Smalltalk of course

Parasol implementation: Minimal platform class

- Have a minmal possible Platform-compatibility class for adapt to different dialects (just 14 methods):
  - #byteArrayFromBase64String:
  - #dateAndTimeFromUnixTime:, #unixTimeFromDateAndTime
  - #ensureSeasideServerRunning
  - #handleRequest:
  - #fileDelete:, #fileWrite:contents:
  - #httpDelete:, #httpGet:, #httpPost:data:timeout:
  - #jsonRead: , #jsonWrite:
  - #pngImageFromByteArray:
  - #signalException:

Selenium 1 concepts: Commands / Locators

▸ Selenium commands are actions on a web element identified by a locator and with an optional parameter

▸ Syntax

  ▸ *selenium action: <Locator>*

  ▸ *selenium action: <Locator> value: <Parameter>*

▸ Examples

  ▸ *selenium click: #MakeOrder*

  ▸ *selenium type: #NumberOfItems value: 5*

  ▸ *selenium select: #Country value: 'Belgium'*

  ▸ *selenium isElementPresent: #OK*

Carsten Härle, www.straightec.de

▸ Simple / Built-In

  ▸ HTML-Id of the Elements

  ▸ link (E.g. link= 'link=Neu...')

  ▸ X-Path, CSS-Selectors

▸ Problem: Often unreadable and/or indetermistic

  ▸ selenium click: 'css=body'.

  ▸ selenium click: 'css=a[title="Grundstück Neu"] > img'.

  ▸ selenium click: 'id=toolBar.btAbbrechen'.

  ▸ selenium click: 'css=a[title="Bearbeiten"] > img'.

  ▸ selenium click: 'id=WebModalMessageBoxButton~OK'.

  ▸ selenium click: 'id=image_w5R1BYpeaf9Fx'.

  ▸ selenium click: '//div[@id="tlbObjekte"]/div[2]/table/tbody/tr[2]/td[4]'.

▶ Have easy methods to generate Locators

  ▶ *selenium listView: listViewName row: anInteger*

  ▶ *selenium listView: listViewName cellText: 'Moby Dick'*

▶ Simplified Syntax

  ▶ Standard: *selenium click: (selenium listView: #Overview row: 1)*

  ▶ Simplified: *selenium clickListView: #Overview row: 1*

▶ Have easy methods for complex interactions / UI items

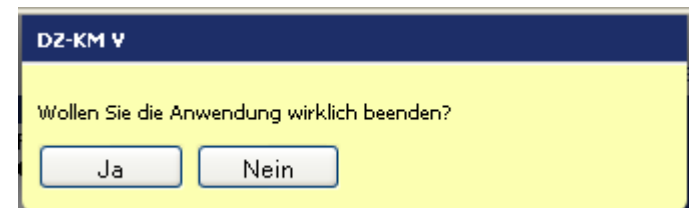 ▶ *selenium clickTreeView: #Overview itemLabelPlus: #('Water' 'Wastewater')*

Sub Applications

Applikation 1 with 2 windows

Applikation 2 with 1 windows

Carsten Härle, www.straightec.de
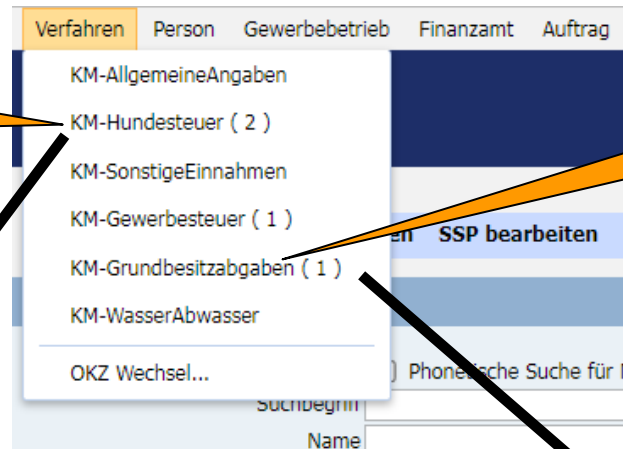
Complex, high level commands

▸ Open context menu of a list and execute menu item

  ▸ *selenium clickListView: #BookList menuItem: 'Order'*

▸ Click in a hierarchical navigation menu

  ▸ *selenium clickNavigationsmenu: #('Tax Type' 'Dog tax' 'Dog' 'New...')*

▸ Confirm message box

  ▸ *selenium clickMessageBoxYes: 'Order book?'*

  ▸ Exception if wrong text (e.g. 'Format hard disk?')

DZ-KM V

Wollen Sie die Anwendung wirklich beenden?

[Ja]   [Nein]

▶ Click button in toolbar

  ▶ *selenium clickToolbarButton: #ok*

  ▶ *selenium clickToolbarButton: #cancel*

  ▶ *selenium clickToolbarButton: #print*

▶ Open subtree in a tree widget

  ▶ *selenium clickTreeView: #myTree itemLabelPlus: 'Tax'*

  ▶ *selenium clickTreeView: #myTree itemLabelMinus: 'Tax'*

  ▶ *selenium clickTreeView: #myTree itemLabel: #('Tax' 'A')*

  ▶ *selenium clickTreeView: #myTree itemLabelCheck: 'Tax'*
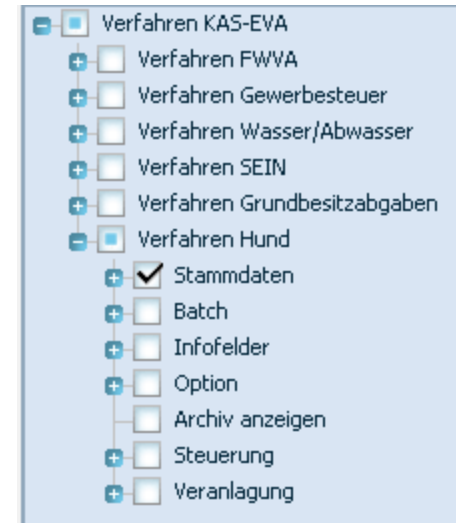
testAddress

    self login.

    selenium clickNavigationsmenu: #('Tax Type' 'Dog Tax' 'Dogs' 'Search...').

    selenium type: 'AccountNumber' value: 7700020.

    selenium clickToolbarButton: #ok.

    selenium clickSidebar: 'Account' group: 'Functions'.

    selenium clickListView: #table row: 1.

    selenium clickListView: #table specialButton: 'Edit'.

    selenium clickTab: 'Address'.

    selenium clickListView: #Addresses menuItem: 'New'.

    selenium typeComboBox: 'Street' value: 'Main street'.

    selenium type: 'HouseNumber' value: 1.

    selenium clickListView: #Addresses specialButton: 'Apply'.

    selenium clickMessageBoxYes: 'Do you want to change the address? '

    selenium clickToolbarButton: #ok.

Problem AJAX

- Problems
  - No page loads occur (page loading event doesn't fire)
  - AJAX requests are asynchronously and may even occur in parallel or with some delay (window.setTimeout())
  - Result: Test script tries to click on elements not present or clicks before the page or element is fully initialized
- Solution
  - Have JavaScript code executed in the browser to detect rendering status and poll before command execution
  - Sometimes framework modifications is necessary

Carsten Härle, www.straightec.de

executeWebDriverCommand: aBlock message: aMessage
  self checkUnexpectedMessageBoxForMessage: aMessage.
  aBlock value.
  self waitForWaitCondition

Selenium class>> waitCondition
  ^'return waitCondition == null || waitCondition()

But only if it is not a read command:
isReadCommand: aMessage
  ^'waitForCondition.|get.*|is.*' asRegex matches: aMessage selector*

Problem modal dialog and custom wait modes

- ▸ Problem
  - ▸ Rich AJAX applications often have modal dialog boxes (e.g. Yes/No/OK) or custom wait modes (e.g. „Wait for processing", animated wait cursors or gray wait screens)
  - ▸ Result: Test script doesn't wait and may even interact with elements behind a modal dialog box

- ▸ Solution
  - ▸ Write special JavaScript code to detect wait conditions.
    - ▸ Wait for wait screens with timeout
    - ▸ If modal dialog box jumps in the way, fail with „Unexpected messagebox: <messagebox text>"

Problem modal dialog and custom wait modes

▸ Check for a present Messagebox before each action (not being a pure read command)

▸ If yes, ensure that either:

  ▸ Action refers to typical message id elements, like MessageBoxButtonYes, MessageBoxButtonNo, MessageBoxButtonOK

  ▸ Test code explictily declares messagebox clicks
    selenium clickMessageBox: #Proceed
    selenium clickMessageBoxYes: 'Are you sure?'

  ▸ Test code explicitly declares message interaction by:
    selenium messageBoxDuring: [selenium …….]

Carsten Härle, www.straightec.de

▸ Problem

  ▸ Input fields are disabled or a custom pseudo disablement is used (e.g. another color and attached click handlers)

  ▸ Result: Test script which doesn't expect disablement , inserts data into these fields

▸ Solution

  ▸ Write special JavaScript code to detect normal or custom disablementments and check before value insertion

▸ Dynamic list loading

  ▸ Wait for list to fill

▸ Dynamic Droplist loading

  ▸ Scroll down and wait for list to fill

▸ Item to be selected not visible and not rendered

  ▸ Scroll to the element, wait for rendering, then click

▸ EdgeDriver, ChromeDriver needs executeable depending from version

  ▸ Automatically determine version and start it

Demos

▸ Application

  ▸ Demo ComboBox

  ▸ Demo scrolling

  ▸ Demo dynamic resize

▸ Testing

  ▸ Demo Testcase running

Carsten Härle, www.straightec.de

▸ Applications codes runs in the same image as test code

▸ Possibility to interfere with application and access, change and create Smalltalk objects

▸ Extremely powerful compared to just UI access

▸ *testSmalltalkObjects*
        self login.
        selenium click: #OpenLastBookOrder.
        *self assert: (self **activeController** object bookName = 'Moby Dick').*

Continuous Build and Test

▸ A central database keeps track of

   ▸ All test cases

   ▸ All the test case results

   ▸ All code versions and environment parameters like database type (Oracle, MSSQL) and browser (Edge, FF)

▸ All test case executions of versioned code stored in DB

▸ 50 VMs constantly loading code and execute test cases or different versions, databases and browsers

▸ Determination of version in which test case failed first

Carsten Härle, www.straightec.de

Coordinated Test Results from other VMs

Carsten Härle, www.straightec.de

Log all test case executions in a database

Carsten Härle, www.straightec.de

Thanks for your attention.
Questions?

Contact: Carsten Härle
Email: carsten.haerle@straightec.de

straightec GmbH
Frankfurter Straße 48, 63150 Heusenstamm, Germany
Freephone: 0800 - straightec (0800 - 7872444)
Tel +49 (6104) 60 08-0, Fax +49 (6104) 60 08-29
E-Mail info@straightec.de , Internet https://www.straightec.de