GUI unit tests for Apache OFBiz

Olivier Heintz,
OFBiz functional consultant

OFBiz extra
- components
- addons
- tools
- ...

APACHECON EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
NOVEMBER 17-21, 2014

Presentation will cover

- Required tools
  - ✔ Selenium, (ide, builder, Server, language bindings)
  - ✔ Jenkins and test recording tools (saucelab)
- How to create Gui automated tests
  - ✔ For standard OFBiz GUI : screen, menu, form, xml
  - ✔ For OFBiz Portal and portlet (Ajaxified or not)
- Implementation feedback and Best Practice

Selenium :
Browser Automation

- Start from OpenQA.org (Quality Assurance) project
- licence Apache 2.0
- 3 sub-projects
  - ✔ IDE : is a Firefox add-on that makes it easy to record and playback tests in Firefox 2+
  - ✔ API to drive interner Browser
    - Webdriver : can drive a browser natively either locally or on remote machines using browser native API.
    - Selenium RC (Remote Control) a leagy technique that uses injected javascript to drive the browser. (Deprecated)
  - ✔ Selenium server
    - manage multiple environments (texts excution units) from a central point.
    - scale by distributing tests on several machines (parallel execution).
- Usable for
  - ✔ Non regression testing in a continuous integration system
  - ✔ System load test

Presentation environment

- OFBiz 13.07.01 with load-demo
  - ✔ With webdriver integrated
  - ✔ With webhelp – portlet-widget – portlet-party (from ofbizextra)
- Firefox browser
- Selenium
  - ✔ Ide on http://www.seleniumhq.org/download/
    http://release.seleniumhq.org/selenium-ide/2.8.0/selenium-ide-2.8.0.xpi
  - ✔ Builder https://saucelabs.com/builder
    http://www.saucelabs.com/addons/selenium-builder-latest-sauce.xpi

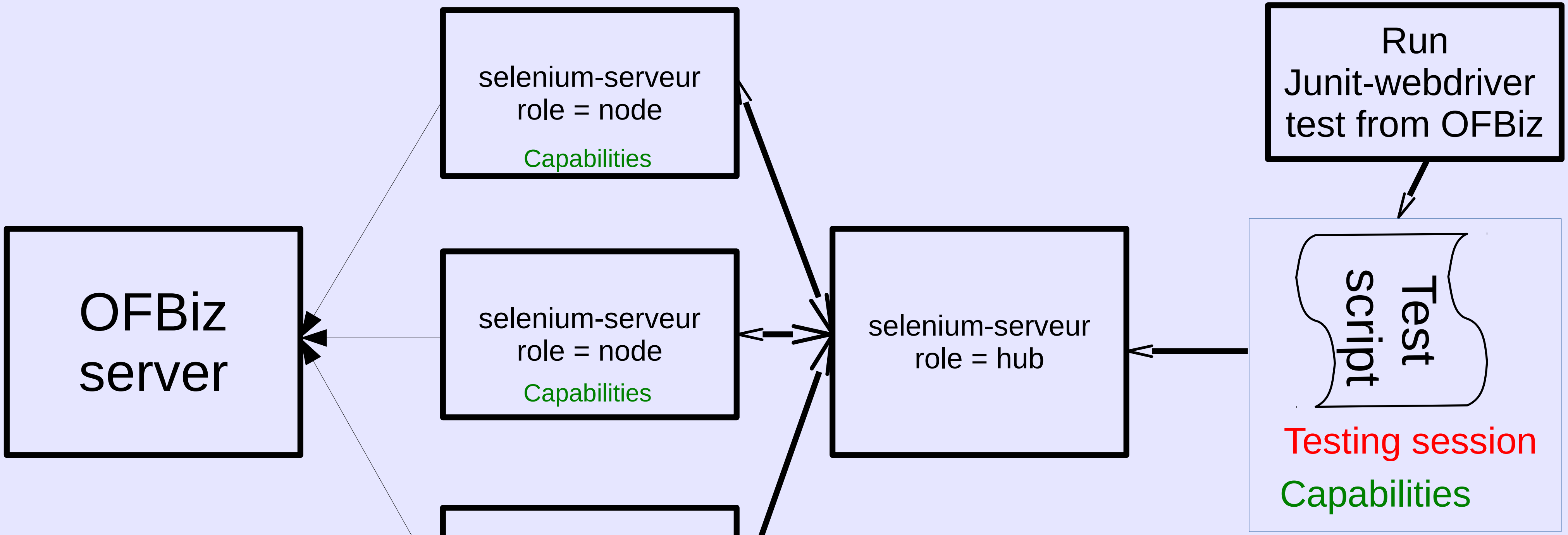
First Test, creation

- Party creation
  - ✔ OFBiz connexion
  - ✔ Theme / langage selection
  - ✔ Party creation
  - ✔ Validation
- Using
  - ✔ Selenium IDE / Builder

First test, using it

- Directly with record tools
- Record it as a java file
  - ✔ Modification to be usable
- Run it with ant task ant run-one-webdriver-test -DtestName=Test_step1
  - ✔ Local mode (no, too slow)
  - ✔ Grid mode
    - Run hub
    - Run node
    - Run test

Grid mode component



Result : xxx/apache-ofbiz-13.07.01/framework/testtools/selenium/build/reports/html/index.html

OFBiz webdriver integration

- Add in testtools component
  - ✔ selenium directory
    - lib directory with all jar need to build a “selenium class” copied from maven repository http://mvnrepository.com/artifact/org.seleniumhq.selenium
    - resources directory with properties file and browser driver
      - selenium.properties
      - simplelog.properties and commons-logging.properties
      - Chromedriver-32 and chromedriver-64
    - tests/org/ofbiz/testools/webdriverTests
      - utils directory
      - 2 test exemples
  - ✔ some target in build.xml
- java interface creation : each new test extend WebdriverTestCase
- Add two ant entries in build.xml
  - ✔ Running test, (1) build 2) run test)

```
  <!--#Bam# webdriver-tools -->
  <property name="wd.base.dir" value="framework/testtools/selenium"/>
  <loadproperties srcFile="${wd.base.dir}/ressources/selenium.properties"/>
  <target name="clean-webdriver-test-data" description="cleans all files
generated by webdriver test processig">
      <ant antfile="framework/testtools/build.xml" target="clean-wd"/>
  </target>
  <target name="run-one-webdriver-test" depends="start-ofbiz-for-tests"
description="run one webdrver test. example of commande : ant run-one-wd
-DtestName=ExampleWebDriverTest (ExampleWebDriverTest is the java file
name without extension)">
      <ant antfile="framework/testtools/build.xml" target="run-one-wd"/>
      <ant target="stop-ofbiz-after-test"/>
  </target>
  <target name="run-all-webdriver-tests" depends="start-ofbiz-for-tests"
description="run all available webdriver tests">
      <ant antfile="framework/testtools/build.xml" target="run-all-wd"/>
      <ant target="stop-ofbiz-after-test"/>
  </target>
  <target name="stop-ofbiz-after-test" unless="ALREADY_STARTED">
      <if>
          <http url="${ofbiz.started.test.url}"/>
          <then>
              <echo message="Stopping Ofbiz"/>
              <antcall target="stop"/>
              <echo message="Ofbiz Stopped"/>
         </then>
          <else>
              <echo message="Ofbiz was not already Started"></echo>
          </else>
      </if>
  </target>
```

```
  <target name="start-ofbiz-for-tests">
      <if>
          <not>
              <http url="${ofbiz.started.test.url}"/>
          </not>
          <then>
              <echo message="Starting Ofbiz"/>
              <antcall target="start-batch"/>
              <waitfor maxwait="3" maxwaitunit="minute"
checkevery="20" checkeveryunit="second">
                  <http url="${ofbiz.started.test.url}"/>
              </waitfor>
              <echo message="Ofbiz Started"/>
         </then>
         <else>
             <property name="ALREADY_STARTED" value="true"/>
             <echo message="Ofbiz already Started"></echo>
         </else>
      </if>
  </target>
  <target name="wd-javadocs" description="run all available
webdriver tests">
      <ant antfile="framework/testtools/build.xml" target="wd-
javadocs"/>
  </target>
  <!--#Eam# webdriver-tools -->
```

First test, conclusions

- Theme and language dependent, like all gui test
  - ✔ Need documentation to easily correct it after a gui modification
- Data dependent, like all tests
- Difficult to use it on project context if screen or menu is modified

Unit test for GUI => unit screen

- Portlet
  - ✔ Using in a portal page => same test mutiple context
- Ajaxified Portlet
  - ✔ No dependency from screen context
- Portlet with parameters
  - ✔ Test should be done for each parameters

Selenium with Ajax

- Only one rule, but mandatory
  - ✔ waitFor after each click to manually manage when it's possible to continue
  - ✔ driverWait.until(ExpectedConditions.presenceOfElementLocated(By.id("xxxxx")));
  - ✔ driverWait.until(ExpectedConditions.invisibilityOfElementLocated(By.id("xxxxxx")));
  - ✔ driverWait.until(ExpectedConditions.textToBePresentInElement(By.cssSelector("li.nav-displaying"), "Displaying 21 - 25 of 25"));
- html, xpath, css knowlegde

Portlet (with portlet-type) test

- One test file by Portal Page
  - ✔ Portal page type : Mgmt, Recap
- Example
  - ✔ ExampleMgmt
    - Find, List
    - portlet by portlet
  - ✔ ExampleRecap
    - Re-use portlet method from ExampleMgmt

Continuous integration

- With jenkins
  - and saucelabs (there is a saucelabs plugin)
    - Change the gridHub url

      http://${username}:${acces-key}@ondemand.saucelabs.com:80/wd/hub
  - With owns tools
    - Video recorder
      - Debug
      - Demonstration
      - Tutorial
    - To easily copy video and log in jenkins job

Advices

- Javadoc of course
- Process but unit test
  - ✔ One test per method
  - ✔ Test should be run two times without problem
    - Currently, not use delegator test
      - Sometime it's necessary to remove manually some data with webtools

```
Utils.removeMultipleWithWebtools(ofbizBaseUrl, driverWait, driver, "ExampleStatus", "exampleId", "EX35");
```

For a quick test (1/2)

- Download ofbiz : trunk or 13.07
- Download adm

http://sourceforge.net/projects/ofbizextra/files/addons_manager/

- Install addon webdriver-tools
  - ✔ Trunk :

```
java -jar /usr/local/app/addonmanager.jar install org.ofbizextra.framework.webdriver-tools
```

  - ✔ 13.07

```
java -jar /usr/local/app/addonmanager.jar init -b 13.07
java -jar /usr/local/app/addonmanager.jar install org.ofbizextra.framework.webdriver-tools
```

- Download selenium-server on http://www.seleniumhq.org/download/

http://selenium-release.storage.googleapis.com/2.44/selenium-server-standalone-2.44.0.jar

- Load demo data in OFBiz
- Change in framework/testtools/selenium/ressources/selenium.properties
  - ✔ From testType=local to testType=grid

For a quick test (2/2)

- Start ofbiz
- Start a selenium-server hub

```
java -jar selenium-server-standalone-2.44.0.jar -role hub
```

- Start a selenium-server node connected to the hub

```
java -jar selenium-server-standalone-2.44.0.jar -role  node  -hub http://localhost:4444/grid/register
```

- Run the 3 base tests

```
ant run-all-webdriver-tests
```

- Look the result

file:///your Local path to ofbiz/framework/testtools/selenium/build/reports/html/index.html

- For example test with ajaxified portlet

```
java -jar /usr/local/app/addonmanager.jar install org.ofbizextra.framework.portlet-widget
```

or

```
java -jar /usr/local/app/addonmanager.jar install org.ofbizextra.party.portlet-party
```

and load-demo and run-all-webdriver-tests

For looking result

https://jenkins.ofbizextra.org/view/13.07-Sel./

✔ Looking selenium job

- log
- Video