GEB UND GRAPHENE IM VERGLEICH

Stefan Hildebrandt / @hildebrandttk

FOLIEN ALS HTML-PRÄSENTATION

TESTEN VON WEBANWENDUNGEN

- Akzeptanztests
- Funktionale Tests
- Unit-Tests von Komponenten
- Last- / Kapazitätstests

BEISPIELE

1. Google Suche
2. Java EE 7 Petclinic
   - Von Thomas Wöhlke auf github
   - Fachlichkeit:
     - Tierärzte mit Spezialisierungen
     - Haustiere mit Arten
     - Besitzer haben Haustieren
     - Besitzer kommen mit Haustieren zu einem Besuch
   - Fork mit Testerweiterungen auf github

GEMEINSAME BASIS

SELENIUM

SELENIUM HISTORIE

- Selenium RC: 2004
- WebDriver: 2006
- Merge zu Selenium 2: 2008

SELENIUM 2 BINDINGS

- java
- C#
- phyton
- ruby
- php
- perl
- javascript

SELENIUM 2 BROWSERUNTERSTÜTZUNG

- Firefox
- Internet Explorer
- Chrome
- Safari
- HTMLUnit
- Phantom JS
- iOS
- Android

GEB (PRONOUNCED “JEB”)

Hint: "gebish" für Suchen

- WebDriver
- jQuery Selection-API
- Groovy
- JUnit, TestNG oder Spock
- Release 0.4 vor 4,5 Jahren, aktuell: 0.10.0
- gradleware-Entwickler

ARQUILLIAN

- Von JBoss für Tests ihres AS und Frameworks entwickelt
- Deployment des Testobjekts in einen EE Container (CDI, Servlet, Appserver)
- Tests im Container oder als Client
- Injection von EE-Komponenten in die Tests
- JUnit und TestNG

ARQUILLIAN DRONE & GRAPHENE

- Graphene & Drone sind Arquillian Extensions
- Aus dem JBoss Umfeld
- Drone ca. 3,5 Jahren
- Graphene ca. 3 Jahre

BEISPIEL VON DER SELENIUM HOMEPAGE:

```
public class Selenium2Example  {
   public static void main(String[] args) {
      WebDriver driver = new FirefoxDriver();
      driver.get("http://www.google.com");
      WebElement element = driver.findElement(By.name("q"));
      element.sendKeys("Cheese!");
      element.submit();
      System.out.println("Page title is: " + driver.getTitle());
      (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {
         public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("cheese!");
         }
      });
      System.out.println("Page title is: " + driver.getTitle());
      driver.quit();
   }
}
```

LESBARKEIT

WIEDERVERWENDBARKEIT

SELENIUM PAGE OBJECTS

- Seitenstruktur
- Bedienlogik
- Fluent API
  ⇒ Führung bei der Testerstellung per Code Completion
- Synchron

PAGE

```
public class FindOwnersPage extends AbstractPage<FindOwnersPage> {

   @FindBy(id = "findOwnersForm:search")
   private WebElement search;

   @FindBy(css = "input[type='text']")
   private WebElement nameInput;

   ...

   public FindOwnersResultPage searchForOwner(String name) {
      nameInput.clear();
      nameInput.sendKeys(name);
      search.click();
      return new FindOwnersResultPage().waitForIsLoaded();
   }
}
```

TEST

```
@Test
public void testOpenNewOwnerPageFromOwnersList() {
   final FindOwnersPage findOwnersPage = new FindOwnersPage();
   findOwnersPage.get();
   findOwnersPage.assertIsLoaded()
      .searchForOwner("")
      .assertIsLoaded()
      .clickNewOwner()
      .assertPageIsLoaded();
}
```

TECHNISCHE OBERKLASSE

```
public abstract class AbstractPage<T extends AbstractPage<T>> extends Loadable
   private static WebDriver driver;

   protected AbstractPage() {
      driver = WebDriverHolder.getDriver();
      PageFactory.initElements(driver, this);
   }

   @Override
   protected final void load() {
      getDriver().get(BASE_URL + pageUrl);
   }

   @Override
   protected void isLoaded() throws Error {
      assertTrue(getDriver().getCurrentUrl().endsWith(pageUrl));
   }
```

HILFSKLASSE

```
public class WebDriverHolder {
   private static WebDriver driver;

   public static WebDriver getDriver() {
      if (driver == null) {
            ...
            driver = new FirefoxDriver(profile);
      }
      return driver;
   }

   public static void closeDriver() {
      if (driver != null) {
         driver.quit();
         driver = null;
      }
   }
```

FAZIT

↗ Wiederverwendbarkeit
→ Fluent-API mit Problemen
↘ Eigene Framework-Klassen notwendig
↘ Manuelles Setup des Browsers

ARQUILLIAN GRAPHENE INKL. DRONE

ARQUILLIAN DRONE

- WebDriver Lifecycle inkl. Konfiguration
- WebDriver Injection in den Test

WEBDRIVER INJECTION

```
@RunWith(Arquillian.class)
public class TestDroneOnly {

   @Drone
   private WebDriver driver;

   @ArquillianResource
   private URL deploymentUrl;

   @Test
   public void testOpeningHomePage() {
      driver.get(deploymentUrl + "/hello.jsf");
      assertEquals("Java EE 7 Petclinic", driver.getTitle());
   }
}
```

KONFIGURATION IN ARQUILLIAN.XML

```
<arquillian xmlns="http://jboss.org/schema/arquillian"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xsi:schemaLocation="http://jboss.org/schema/arquillian
                                     http://jboss.org/schema/arquillian/arqui

   <extension qualifier="webdriver">
      <property name="browser">firefox</property>
      <!--<property name="browser">phantomjs</property>-->
      <!--<property name="browser">chrome</property>-->
   </extension>

   <extension qualifier="drone">
      <property name="instantiationTimeoutInSeconds">120</property>
   </extension>
</arquillian>
```

ARQUILLIAN GRAPHENE

PAGE OBJECT

```
@Location("findOwners.jsf")
public class FindOwnersPage<T extends FindOwnersPage<T>> extends AbstractFindC

   @Page
   private FindOwnersResultPage findOwnersResultPage;

   public FindOwnersResultPage searchForOwner(String name) {
      searchForOwnerInternal(name);
      return findOwnersResultPage;
   }
}
```

TECHNISCH NOTWENDIGE FACHLICHE OBERKLASSE

```
public abstract class AbstractFindOwnersPage<T extends AbstractFindOwnersPage<
   @FindBy(css = "input[type='text']")
   private WebElement nameInput;
   @Page
   private NewOwnerPage newOwnerPage;

   ...

   public NewOwnerPage openNewOwnersPage() {
      addNewOwnerLink.click();
      return newOwnerPage;
   }

   protected void searchForOwnerInternal(String name) {
      nameInput.clear();
      nameInput.sendKeys(name);
      search.click();
   }
```

TEST

```
@Test
public void testOpenNewOwnerPageFromOwnersList() {
   goTo(FindOwnersPage.class)
      .assertIsLoaded()
      .searchForOwner("")
      .assertIsLoaded()
      .openNewOwnersPage()
      .assertIsLoaded();
}
```

FAZIT

- ↗ Browser Lifecycle
- ↗ Page und WebElement Injection
- ↗ echte jQuery Selector-API
- ↘ Direkte Verwendung der Selenium-API
- ↘ Injection von Unterklasse in Oberklasse nicht möglich

GEB

GEB PAGE

```
class FindOwnersPage extends AbstractPetClinicPage {

   static url ='findOwners.jsf'

   static at = { pageHeader.present }

   static content = {
      pageHeader { $('h2', id: 'findOwners') }
      nameInput { $('input', type:'text') }
      searchButton { $('input', type: 'submit') }
      addNewOwnerType { $('a', text: 'Add New Owner') }
   }

   FindOwnersResultPage searchForOwner(String name){
      nameInput.value(name)
      searchButton.click()
      return waitForAtPage(FindOwnersResultPage)
   }
```

TECHNISCHE UND FACHLICHE OBERKLASSE

```
abstract class AbstractPetClinicPage extends Page {

   static content = {
      ...
      findOwnersLink { $("a", text: "Find Owners") }
   }

   ...

   FindOwnersPage toFindOwners() {
      findOwnersLink.click()
      return waitForAtPage(FindOwnersPage);
   }

   def <T extends Page> T waitForAtPage(Class<T> targetPageClass){
               waitFor { browser.isAt(targetPageClass) }
               return browser.page as T;
   }
```

GEB TEST

```
@RunWith(Arquillian)
class Test04Owner extends GebTest {

   ...

   @Test
   public void testOpenNewOwnerPageFromOwnersList() {
      to(HelloPage)
         .toFindOwners()
         .searchForOwner('')
         .openNewOwnersPage()
}}
```

KONFIGURATION: GEBCONFIG.GROOVY

```
baseUrl='http://localhost:8080/petclinic-all/'
driver = {
   def FirefoxProfile profile = new FirefoxProfile();
   ...
   def ffDriver = new FirefoxDriver(profile)
   ffDriver.manage().window().maximize()
   return ffDriver
}
waiting {
   timeout = 10
   retryInterval = 0.5
   presets {
      test {
         timeout = 3
         retryInterval = 0.5
      }
   }
}
```

FAZIT

↗ Page und Browser Lifecyle
↗ an jQuery angelehnte Selector-API
→ Auch Selenium-API verwendbar
↘ Warten auf Page in Page fehlt

KOMPONENTEN

- Natürliche Komponenten
  - Tabellen inkl. Zugriff auf einzelne Zeilen und Spalten
  - Menüs
  - Gleichartige Validierung, Fehlermeldungen, ...
  - Wizzards
- 3. Party Komponenten
  - Komplexe Inputs (Kalender, Vorschlagsboxen, ...)
  - jquery ui, PrimeFaces, RichFaces, ...

GRAPHENE PAGE FRAGMENTS

- Verwendung der selben Annotationen wie in der Page
- Keine Oberklasse
- @Root für die Basis
- Verwendung in der Page wie WebElement

GRAPHENE PAGE FRAGMENT

```
public class OwnersTableFragment {

   @Root
   private WebElement root;

   @FindBy(css = "tbody.rf-dt-b > tr")
   private List<OwnersTableRowFragment> rows;

   public List<OwnersTableRowFragment> findRowsByParameters(String firstName,
                                                            String city, St
      List<OwnersTableRowFragment> matchingRows = new ArrayList<>();
      for (OwnersTableRowFragment row : rows) {
         if (row.getLastName().equals(lastName) && row.getFirstName().equals(f
             && row.getAddress().equals(address) && row.getCity().equals(city)
            matchingRows.add(row);
         }
      }
      return matchingRows;
```

GRAPHENE PAGE FRAGMENTS FÜR 3. PARTY-FRAMEWORKS

Framework	Verfügbarkeit von Page Fragments
Richfaces 4.5	Final
Richfaces 5.0	Alpha3
Primefaces	✘
jQuery UI	✘

GEB MODULE

- geb.Module analog zu geb.Page
- Verwendung im static-Bereich mit dem Schlüsselwort: module
- Unterstützung von Listen mit moduleList

```
petBirthDateInput { module RichFacesCalendar, $('#editPetForm\\:petBirthDate')
```

```
rowsInTable { moduleList OwnersTableRowModule, $('table.table tbody tr') }
```

GEB MODULE

```
class OwnersTableRowModule extends Module {
   static content = {
      cell(required: false ) { $("td", it) }
      editOwnerLink(required: false ) { cell(0).find('a') }
      name(required: false ) { cell(0).text() }
      address(required: false ) { cell(1).text() }
      city(required: false ) { cell(2).text() }
      telephone(required: false ) { cell(3).text() }
   }

   ShowOwnerPage openDetails() {
      editOwnerLink.click()
      waitFor { browser.isAt(ShowOwnerPage) }
      return browser.page as ShowOwnerPage
   }
}
```

GEB KOMPONENTENBIBLIOTHEKEN FÜR 3. PARTY

Einfach möglich

Existieren nicht (öffentlich)

AJAX

GRAPHENE REQUEST GUARDS

- Blockiert Test für eine konfigurierte Wartezeit
- Wirft eine Exception falls kein Request mit Response verzeichnet wurde

```
guardHttp(buttonWhichMakesFullPageRefresh).click();
guardAjax(buttonWhichMakesAjaxRequest).click();
guardNoRequest(buttonWhichMakesNoRequest).click();
```

GRAPHENE 2 WAITINGS

- Fluent API
- Referenzierung von WebElements

```
button.click();
waitGui()
    .withMessage("Popup should be opened after clicking on that button!")
    .until().element(popupPanel).is().visible();
```

GEB WAITINGSUPPORT

- durch Closure sehr flexibel
- Konfiguration von Defaults in GebConfig.groovy

```
waitFor {}
waitFor(10) {}
waitFor(10, 0.5) {}
waitFor("quick") {}
```

```
waitFor { theResultDiv.present }
```

GEB LAZY CONTENT

```
class DynamicPage extends Page {
    static content = {
        dynamicallyAdded(wait: true) { $("p.dynamic") }
    }
}

Browser.drive {
    to DynamicPage
    assert dynamicallyAdded.text() == "I'm here now"
}
```

- Es wird beim Zugriff automatisch gewartet
- Timing kann konfiguriert werden

VERGLEICH PROGRAMMIERSTIEL & LESBARKEIT

GRAPHENE & DRONE

- ↗ Annotation für Java EE Entwickler gewohnt
- ↗ Verbessertes Selektor-Api
- ↗ Größtenteils transparentes Wireing
- ↗ Waiting API ermöglicht Referenzierung von Feldern
- ↘ WebDriver-API ist in die Jahre gekommen

GEB

- ↗ Transparentes Wireing
- ↗ Verbesserte Selektor-Api
- ↗ Konfiguration z.B. für wait()
- ↗ Groovy-Power-Assertions
- ↗ groovy
- ↘ groovy
- ↘ static-Bereich
- ↘ teilweise untypisiert

KOMBINATIONSMÖGLICHKE

- Arquillian
- Arquillian Suite Deployments
- Arquillian Warp
- Cucumber / fit
- ArquillianCucumber
- jMeter

TESTAUSFÜHRUNG MIT ARQUILLIAN

ARQUILLIAN GRAPHENE

- Arquillian-Extension ✔
- Keine Interferenzen mit Arquillian-Suite ✔

GEB

- Kein eigener Testrunner ✔
- Keine Interferenzen mit Arquillian-Suite ✔

ARQUILLIAN GRAPHENE MIT CUCUMBER

- Ohne Arquillian-Testrunner nicht lauffähig ✘
- Page Injection mit ArquillianCucumber fehlerhaft ✘

GEB MIT CUCUMBER

- Etwas Glue-Code ✔
- ArquillianCucumber ✔

```
class GebStepDefinitions {
   String gebConfEnv = null
   String gebConfScript = null

   private Browser _browser

   Configuration createConf() {
      new ConfigurationLoader(gebConfEnv, System.properties, new GroovyClassLo
   }

   Browser createBrowser() {
      new Browser(createConf())
   }

   Browser getBrowser() {
      if (_browser == null) {
         _browser = createBrowser()
      }
```

PERFORMANCE / CAPACITY TESTS MIT SELENIUM

- ↗ Korrekte Bedienung
- ↗ "Echte" Last
- ↗ Revisionssichere Pflege durch fachliche Tests
- ↗ HtmlUnit oder Phantom JS
- ↘ Im Client Latenz (bis 200ms) durch WebDriver waitLoop
- ↘ Hoher Ressourcenbedarf
- ↘ Eingeschränkte Last

GEB MIT JMETER

- Mit etwas Glue-Code ✔

```
abstract class AbstractGebSamplerClient extends AbstractJavaSamplerClient impl

   @Override
   Arguments getDefaultParameters() {
      return new Arguments()
   }

   @Override
   void setupTest(final JavaSamplerContext context) {
      to(HelloPage)
   }

   @Override
   void teardownTest(final JavaSamplerContext context) {
      resetBrowser()
   }

   String gebConfEnv = null
```

	Arquillian Graphene	geb
Arquillian Deployment	✔	✔
Arquillian Suite Deployment	✔	✔
Arquillian Warp	✔	✔
Cucumber	✘	✔
Cucumber mit Arquillian Deployment	✘	✔
Cucumber mit Arquillian Suite Deployment	✘	✔
jMeter	✘	✔

GESCHWINDIGKEIT DER TESTAUSFÜHRUNG

→ Selenium: Referenz
→ Graphene: geringe Nachteile bei Komponenten
↘ geb: : Redundante-Webdriver-Aufrufe
↗ geb: WebDriver-Cache

STABILITÄT

→ Keine grundsätzlichen Unterschiede

GOODIES

	Arquillian Graphene	geb
Vereinfachte Screenshoots	✔	✔
Javascript einfacher im Browser ausführen	✔	✔
Download	✘	✔
Konfiguration von Browser	✔	✔
Konfiguration von Timings	✘	✔
Angepasste Selektoren	✔	✘
jQuery Selektoren	✔	✔
AngularJS	✔	✘

AUSWAHL KRITERIEN

- graphene
  - ↗ Java EE - Server
  - ↗ Große Basis existierender Selenium Page Objects
  - ↘ Außerhalb von Unit-Tests nicht einsetzbar
- geb
  - ↗ Kombinationsmöglichkeiten
  - ↘ "Andere Sprache"

LINKS

- Folien: stefanh.de/vortraege.htm

- Beispiel auf github
- Luke Daley: Geb -- Very Groovy Browser Automation

STEFAN HILDEBRANDT - CONSULTING.HILDEBRANDT.TK

- Beratung, Coaching und Projektunterstützung
- Java EE
- Buildsysteme gradle und maven/ant-Migration
- Testautomatisierung
- Coach in agilen Projekten
- DevOps