JBoss Community

TESTING JSF APPLICATIONS

With Arquillian And Selenium

Brian Leathem and Lincoln Baxter, III
Senior Software Engineers, Red Hat

JavaOne 2012 - San Francisco
2012-09-04

THE PLAN

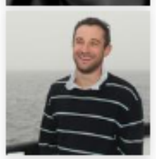
Brief Review of *back-end* testing
How to test the *front-end* with Selenium
Beyond Selenium with Arquillian Extensions

WHO ARE WE?

Senior Software Engineers at Red Hat
RichFaces and Forge Project Leads
Represent Red Hat on the JSF 2.2 EG (JSR-344)

WHO'S BEHIND ALL THIS?

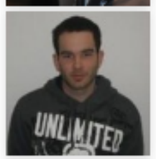
Aslak Knutsen (@aslakknutsen)
Arquillian Lead



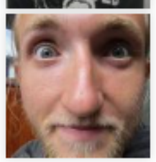
Andrew Lee Rubinger (@ALRubinger)
Shrinkwrap Lead

Lukáš Fryč (@lfryc)
Arquillian Graphene Lead, Arquillian Warp Lead

Karel Piwko (@karelpiwko)
Arquillian Drone Lead

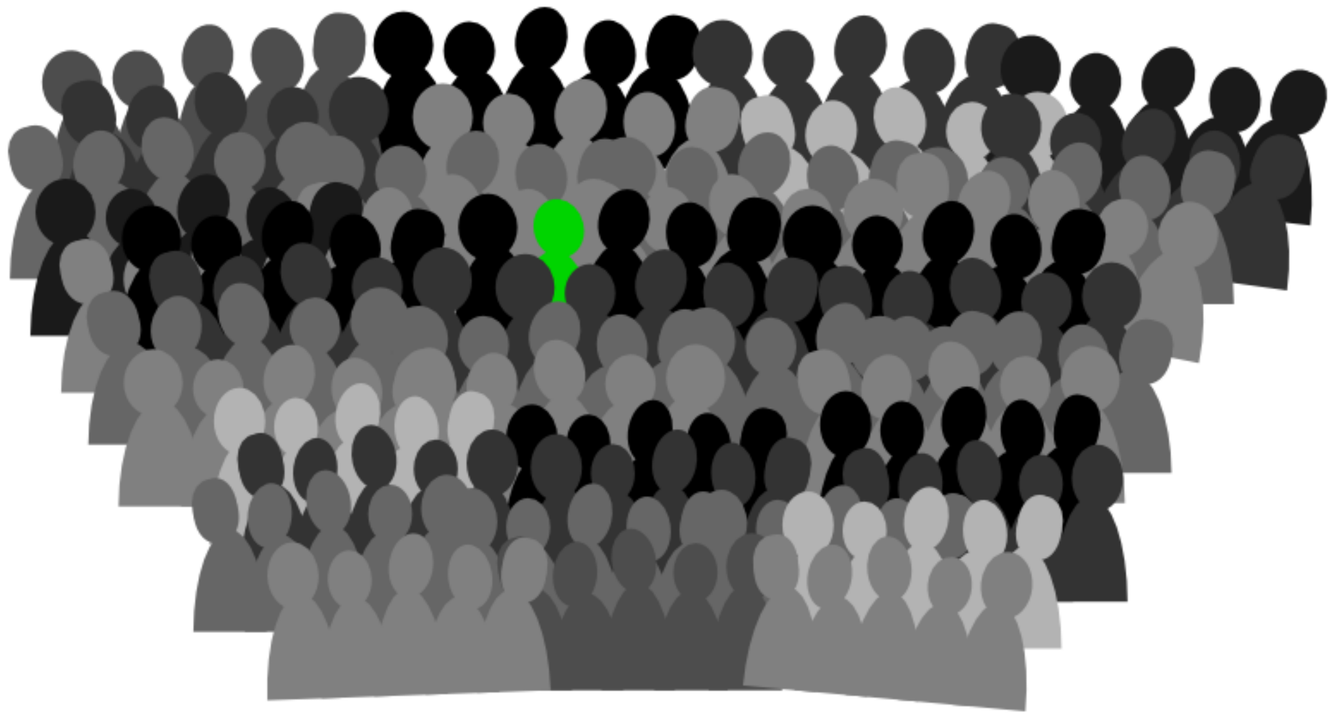


Juraj Huska (@j_huska)
Arquillian Graphene, Arquillian Drone



Jan Papoušek (@jan_papousek)
Arquillian Graphene, Arquillian Drone

WHO ARE YOU?



BACK-END TESTING

A well understood problem

BACK-END TESTING | UNIT TESTS

JUNIT / TESTNG

- Easy to test self-contained logic
- Mocks allow us to extend unit tests further

BACK-END TESTING | 'REAL' TESTS

"REAL TESTS"

- Test code in-container
- No "surprises" in production

ENTER ARQUILLIAN

- Bring the container to the test
- Micro-deployments

DON'T MOCK ME!

EXAMPLE | GREETERTEST

HELLO WORLD!

- Basic Arquillian test
- Demonstrate `@Inject`
- Introduce Shrinkwrap

GREETERTEST | THE BEAN

```
public class Greeter {

    private PhraseBuilder phraseBuilder;

    @Inject
    public Greeter(PhraseBuilder phraseBuilder) {
        this.phraseBuilder = phraseBuilder;
    }

    public void greet(PrintStream to, String name) {
        to.println(createGreeting(name));
    }

    public String createGreeting(String name) {
        return phraseBuilder.buildPhrase("hello", name);
    }
}
```

GREETERTEST | INJECTION

```java
public class PhraseBuilder {
    private Map<String, String> templates;

    public String buildPhrase(String id, Object... args) {
        return MessageFormat.format(templates.get(id), args);
    }

    @PostConstruct
    public void initialize() {
        templates = new HashMap<String, String>();
        templates.put("hello", "Hello, {0}!");
    }
}
```

GREETERTEST | TEST

```java
@RunWith(Arquillian.class)
public class GreeterTest {
    @Deployment
    public static JavaArchive createDeployment() {
        JavaArchive jar = ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, PhraseBuilder.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
        return jar;
    }

    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals("Hello, Earthling!",
            greeter.createGreeting("Earthling"));
    }
}
```

GREETERTEST | TEST

```
@RunWith(Arquillian.class)
public class GreeterTest {
    @Deployment
    public static JavaArchive createDeployment() {
        JavaArchive jar = ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, PhraseBuilder.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
        return jar;
    }

    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals("Hello, Earthling!",
            greeter.createGreeting("Earthling"));
    }
}
```

GREETERTEST | TEST

```
@RunWith(Arquillian.class)
public class GreeterTest {
    @Deployment
    public static JavaArchive createDeployment() {
        JavaArchive jar = ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, PhraseBuilder.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
        return jar;
    }
    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals("Hello, Earthling!",
            greeter.createGreeting("Earthling"));
    }
}
```

GREETERTEST | TEST

```java
@RunWith(Arquillian.class)
public class GreeterTest {
    @Deployment
    public static JavaArchive createDeployment() {
        JavaArchive jar = ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, PhraseBuilder.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
        return jar;
    }

    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals("Hello, Earthling!",
            greeter.createGreeting("Earthling"));
    }
}
```

GREETERTEST | TEST

```java
@RunWith(Arquillian.class)
public class GreeterTest {
    @Deployment
    public static JavaArchive createDeployment() {
        JavaArchive jar = ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, PhraseBuilder.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
        return jar;
    }

    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals("Hello, Earthling!",
            greeter.createGreeting("Earthling"));
    }
}
```

THE MAGIC

POM Dependencies

- Arquillian
- JUnit | TestNG
- Deployment Container

Arquillian.xml

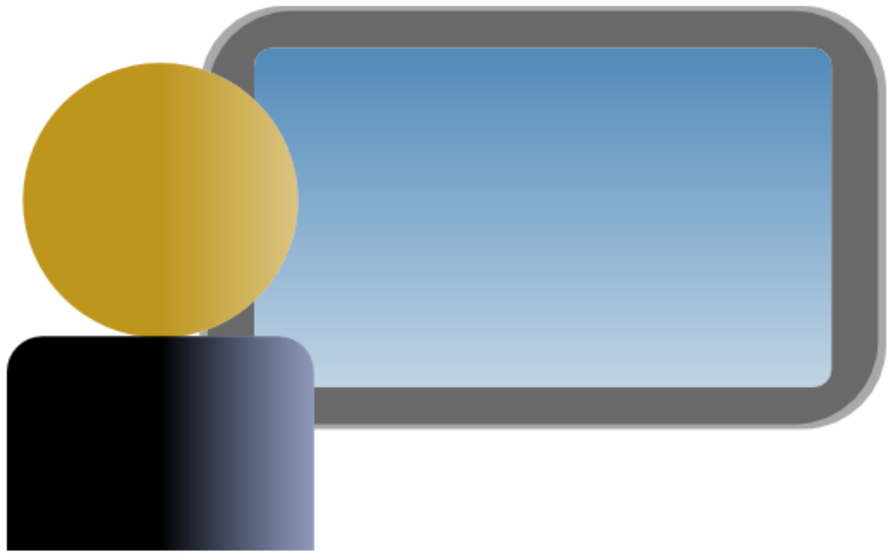
- Configuration of the test container

GREETERTEST | POM

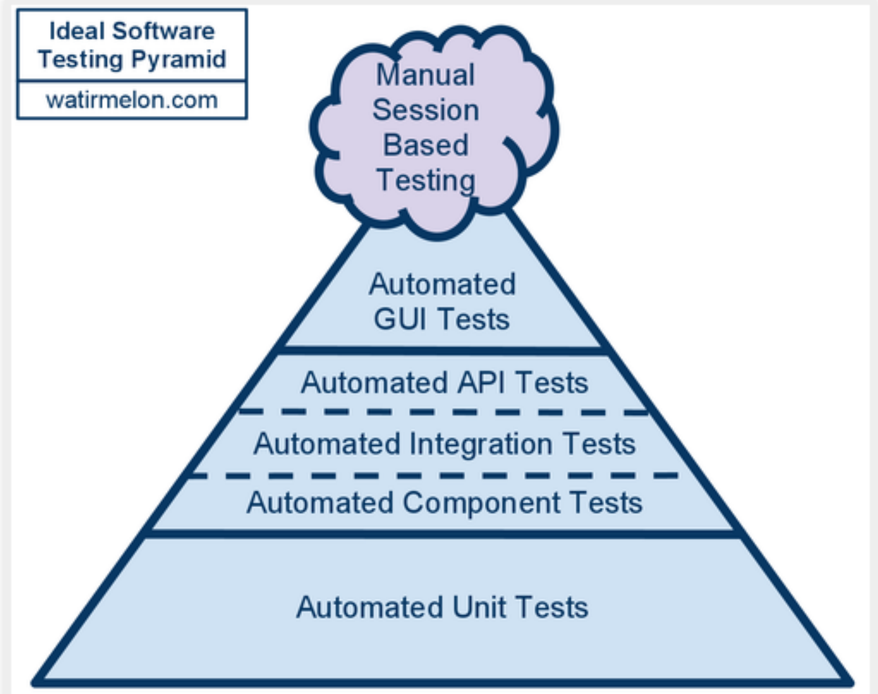
```
<profile>
    <id>arquillian-jbossas-managed</id>
    <dependencies>
        ...
        <dependency>
            <groupId>org.jboss.as</groupId>
            <artifactId>jboss-as-arquillian-container-managed</artifactId>
            <version>7.1.1.Final</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.jboss.arquillian.protocol</groupId>
            <artifactId>arquillian-protocol-servlet</artifactId>
            <scope>test</scope>
        </dependency>

    </dependencies>
</profile>
```

DEMO TIME

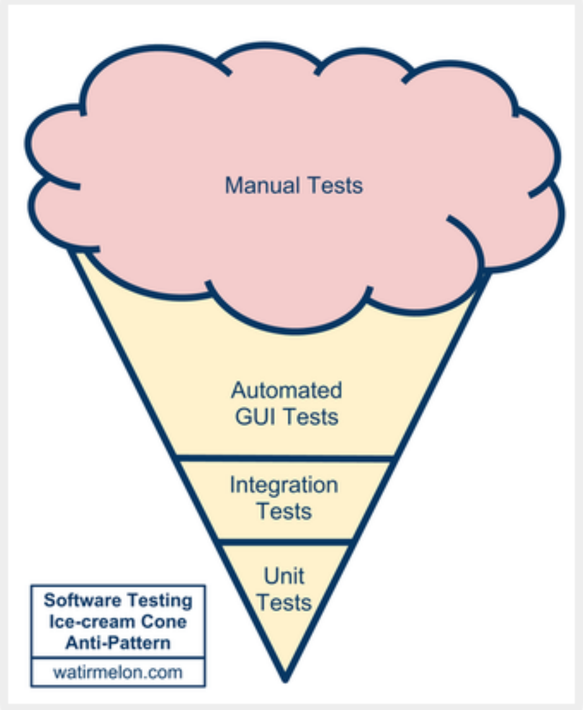
FRONT-END TESTING



IDEAL: PYRAMID OF TEST COVERAGE



REALITY: ICE-CREAM CONE OF TEST COVERAGE



HOWTO INCREASE AUTOMATED TEST COVERAGE?

Unit tests of GUI code **-- lots of mocks!**

Let's focus on real-tests

REAL TESTS → REAL BROWSERS









SELENIUM

Selenium automates browsers. *That's it.*

WebDriver API:

- Unified API for all browsers
- Headless tests via HtmlUnit

SELENIUM | API

WebDriver

Represents the web browser

Methods:

- .get()
- .findElement(By)

SELENIUM | API

WebElement

Represents an HTML element.

Methods:

- .click()
- .sendKeys()
- .getText()
- .findElement(By)

WEBELEMENT LOOKUP

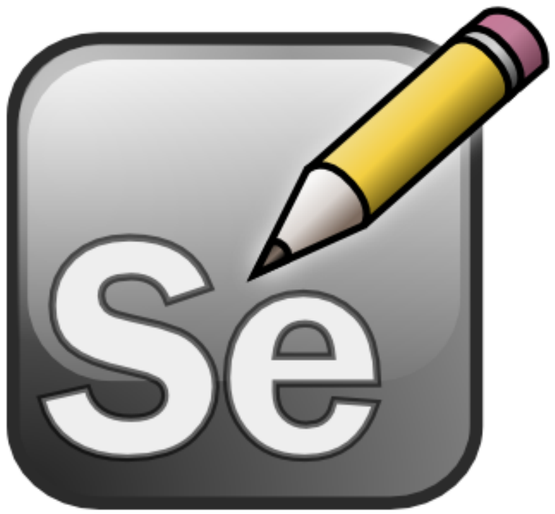
Selenium/WebDriver programmatic API:

```
WebElement username
    = driver.findElement(By.id("username"))
```

Alternative @FindBy annotation:

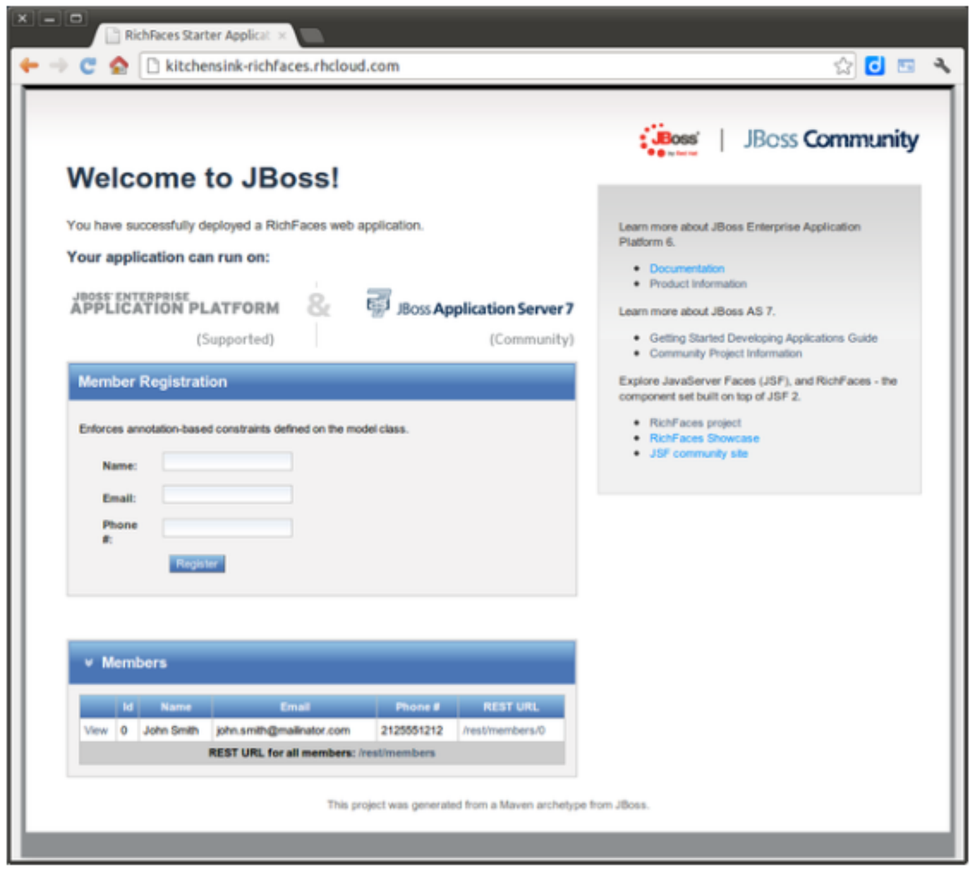
```
@FindBy(id="username")
WebElement username;
...
PageFactory.initElements(driver, this);
```

SELENIUM IDE - EASY!

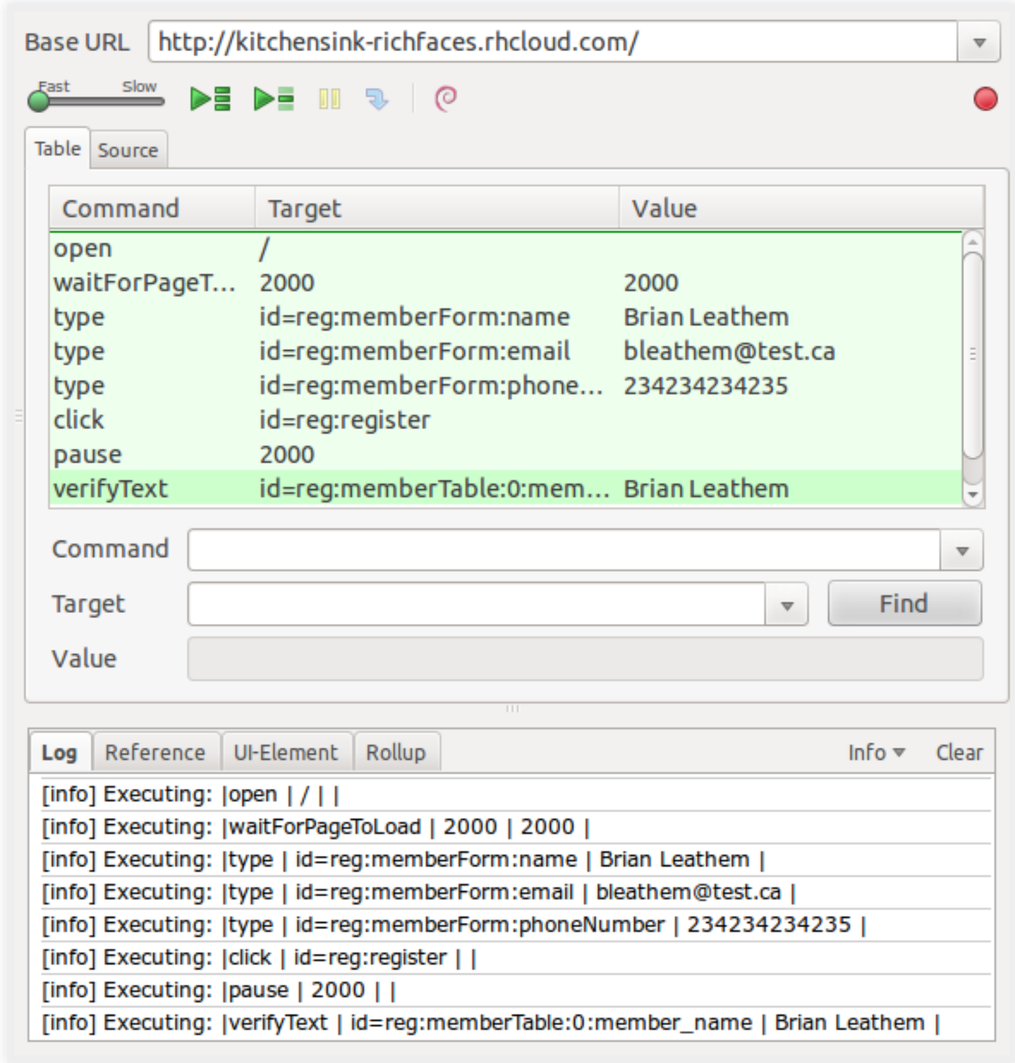


- Easy to write/record tests
- Replay tests against all browsers

EXAMPLE | SELENIUM



SELENIUM IDE



SELENIUM EX. | THE TEST

```
public class SeleniumTest {
    private WebDriver driver;
    private String baseUrl;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
      driver = new FirefoxDriver();
      baseUrl = "http://kitchensink-richfaces.rhcloud.com/";
      driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void test() throws Exception {
      driver.get(baseUrl + "/");
      driver.findElement(By.id("reg:memberForm:name")).clear();
      driver.findElement(By.id("reg:memberForm:name")).sendKeys("Brian Leathem");
      driver.findElement(By.id("reg:memberForm:email")).clear();
      driver.findElement(By.id("reg:memberForm:email")).sendKeys("bleathem@test.ca");
      driver.findElement(By.id("reg:memberForm:phoneNumber")).clear();
      driver.findElement(By.id("reg:memberForm:phoneNumber")).sendKeys("234234234235");
      driver.findElement(By.id("reg:register")).click();
      Thread.sleep(2000);
      try {
        assertEquals("Brian Leathem", driver
          .findElement(By.id("reg:memberTable:0:member_name")).getText());
      } catch (Error e) {
        verificationErrors.append(e.toString());
      }
    }

    @After
    public void tearDown() throws Exception {
      driver.quit();
      String verificationErrorString = verificationErrors.toString();
      if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
      }
    }
}
```

SELENIUM EX. | THE TEST

```
public class SeleniumTest {
    private WebDriver driver;
    private String baseUrl;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
      driver = new FirefoxDriver();
      baseUrl = "http://kitchensink-richfaces.rhcloud.com/";
      driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }
    @Test
    public void test() throws Exception {
      driver.get(baseUrl + "/");
      driver.findElement(By.id("reg:memberForm:name")).clear();
      driver.findElement(By.id("reg:memberForm:name")).sendKeys("Brian Leathem");
      driver.findElement(By.id("reg:memberForm:email")).clear();
      driver.findElement(By.id("reg:memberForm:email")).sendKeys("bleathem@test.ca");
      driver.findElement(By.id("reg:memberForm:phoneNumber")).clear();
      driver.findElement(By.id("reg:memberForm:phoneNumber")).sendKeys("234234234235");
      driver.findElement(By.id("reg:register")).click();
      Thread.sleep(2000);
      try {
        assertEquals("Brian Leathem", driver
          .findElement(By.id("reg:memberTable:0:member_name")).getText());
      } catch (Error e) {
        verificationErrors.append(e.toString());
      }
    }

    @After
    public void tearDown() throws Exception {
      driver.quit();
      String verificationErrorString = verificationErrors.toString();
      if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
      }
    }
}
```

SELENIUM EX. | THE TEST

```
public class SeleniumTest {
    private WebDriver driver;
    private String baseUrl;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
      driver = new FirefoxDriver();
      baseUrl = "http://kitchensink-richfaces.rhcloud.com/";
```

```
@Test
public void test() throws Exception {
  driver.get(baseUrl + "/");
  driver.findElement(By.id("reg:memberForm:name")).clear();
  driver.findElement(By.id("reg:memberForm:name")).sendKeys("Brian Leathem");
  driver.findElement(By.id("reg:memberForm:email")).clear();
  driver.findElement(By.id("reg:memberForm:email")).sendKeys("bleathem@test.ca");
  driver.findElement(By.id("reg:memberForm:phoneNumber")).clear();
  driver.findElement(By.id("reg:memberForm:phoneNumber")).sendKeys("234234234235");
  driver.findElement(By.id("reg:register")).click();
  Thread.sleep(2000);
  try {
    assertEquals("Brian Leathem", driver
      .findElement(By.id("reg:memberTable:0:member_name")).getText());
  } catch (Error e) {
    verificationErrors.append(e.toString());
  }
}
```

```
    @After
    public void tearDown() throws Exception {
      driver.quit();
      String verificationErrorString = verificationErrors.toString();
      if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
      }
    }
}
```

SELENIUM EX. | THE TEST

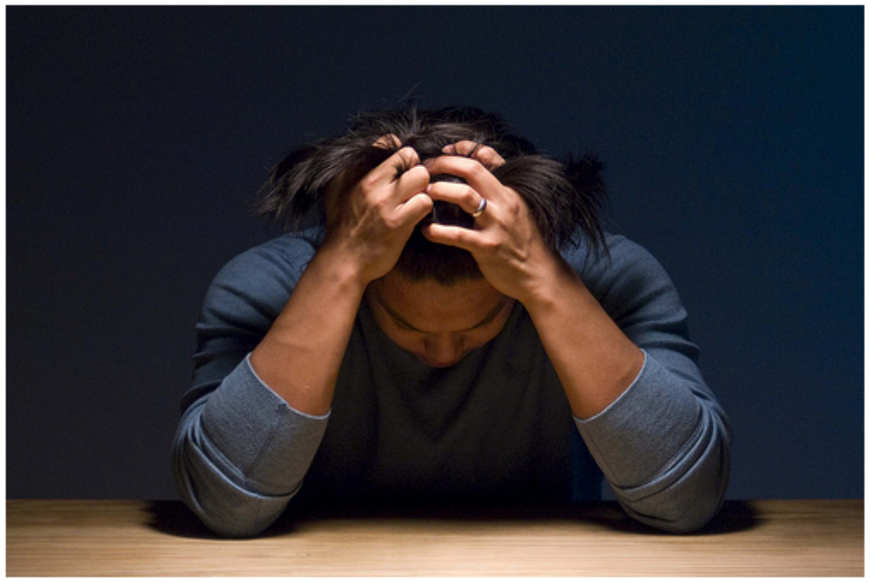
```
public class SeleniumTest {
    private WebDriver driver;
    private String baseUrl;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
      driver = new FirefoxDriver();
      baseUrl = "http://kitchensink-richfaces.rhcloud.com/";
      driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void test() throws Exception {
      driver.get(baseUrl + "/");
      driver.findElement(By.id("reg:memberForm:name")).clear();
      driver.findElement(By.id("reg:memberForm:name")).sendKeys("Brian Leathem");
      driver.findElement(By.id("reg:memberForm:email")).clear();
      driver.findElement(By.id("reg:memberForm:email")).sendKeys("bleathem@test.ca");
      driver.findElement(By.id("reg:memberForm:phoneNumber")).clear();
      driver.findElement(By.id("reg:memberForm:phoneNumber")).sendKeys("234234234235");
      driver.findElement(By.id("reg:register")).click();
      Thread.sleep(2000);
      try {
        assertEquals("Brian Leathem", driver
          .findElement(By.id("reg:memberTable:0:member_name")).getText());
      } catch (Error e) {
        verificationErrors.append(e.toString());
      }
    @After
    public void tearDown() throws Exception {
      driver.quit();
      String verificationErrorString = verificationErrors.toString();
      if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
      }
    }

}
```

DEMO TIME

SELENIUM - PROBLEMS!

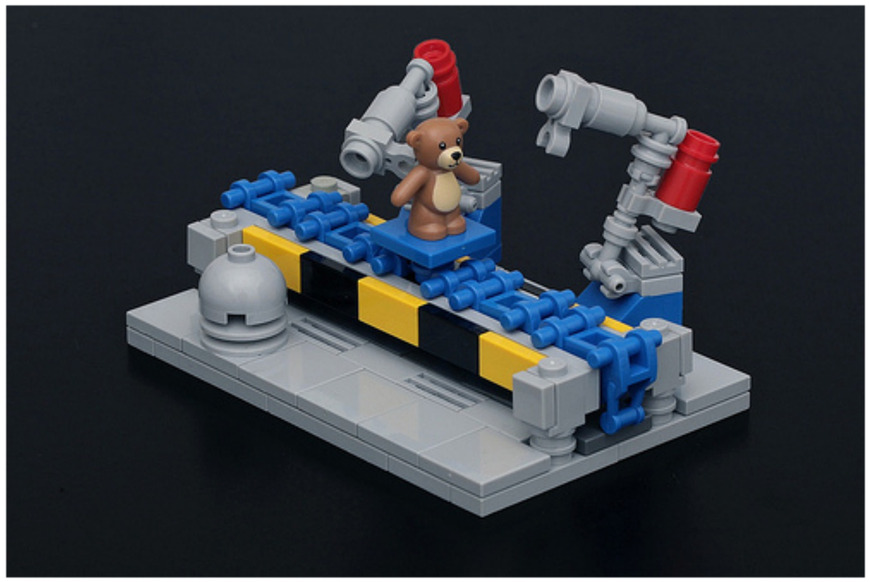
Not enough abstraction, Highly repetitive

→ Maintenance problem



AUTOMATING CLIENT-SIDE TESTING

Deploy the app, run the tests!

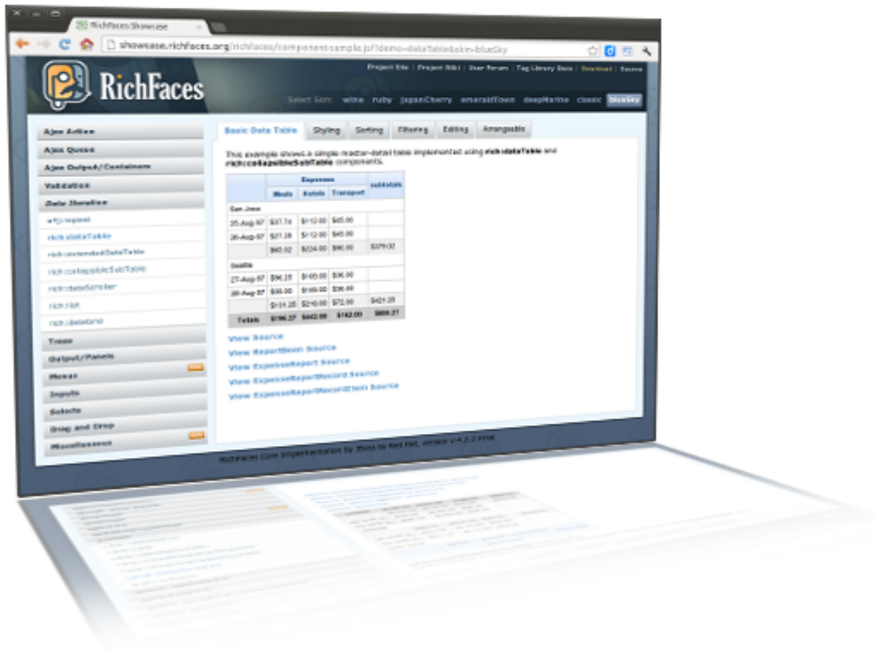


CONTAINER LIFECYCLE

IKE HAS US COVERED!

- Startup/shutdown the container
- Create the testable archive
- Deploy the application

CLIENT LIFECYCLE?



Start / Stop the Browser

ARQUILLIAN EXTENSIONS

ARQUILLIAN EXTENSIONS

Arquillian Drone

Brings the browser to the test

ARQUILLIAN EXTENSIONS

Arquillian Drone

Brings the browser to the test

Arquillian Graphene

Selenium integration and extension

ARQUILLIAN EXTENSIONS

Arquillian Drone

Brings the browser to the test

Arquillian Graphene

Selenium integration and extension

Arquillian Warp

Testing on both sides of the request

ARQUILLIAN DRONE

Manages the browser life-cycle

ARQUILLIAN DRONE

Integrates with:

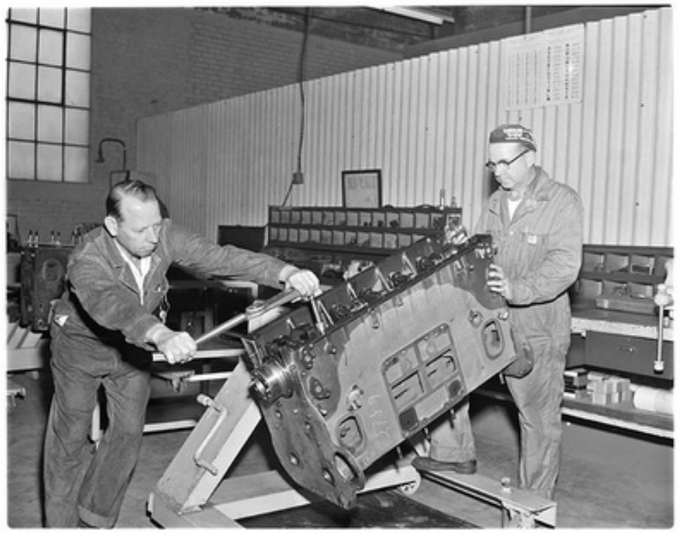
- Selenium/Webdriver
- Arquillian Graphene

```
@Drone
WebDriver browser
```

```
<extension qualifier="webdriver">
    <property name="browserCapabilities">chrome</property>
</extension>
```

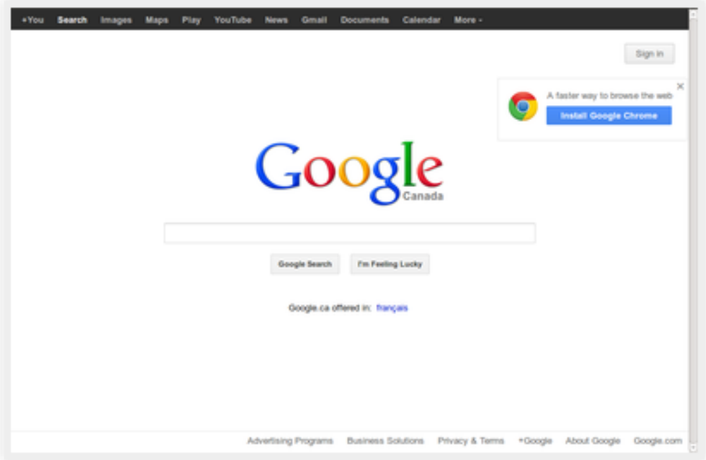
SEPARATION OF CONCERNS

Devs focus: Author tests



QA focus: Automate tests

EXAMPLE | GOOGLE DRONE TEST



- Simple Drone test
- Demonstrate `@Drone`

GOOGLE DRONE TEST | TEST

```java
@RunWith(Arquillian.class)
public class GoogleDroneTest {

    @Drone
    WebDriver browser;

    @Test
    public void testOpeningHomePage() {
        browser.get("http://www.google.com/");
        List<WebElement> elements = browser.findElements(
            By.xpath("//span[contains(text(), 'Google Search')]"));

        Assert.assertTrue("Page not loaded", elements.size() > 0);
    }
}
```

GOOGLE DRONE TEST | TEST

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

    @Drone
    WebDriver browser;

    @Test
    public void testOpeningHomePage() {
        browser.get("http://www.google.com/");
        List<WebElement> elements = browser.findElements(
            By.xpath("//span[contains(text(), 'Google Search')]"));

        Assert.assertTrue("Page not loaded", elements.size() > 0);
    }
}
```

GOOGLE DRONE TEST | TEST

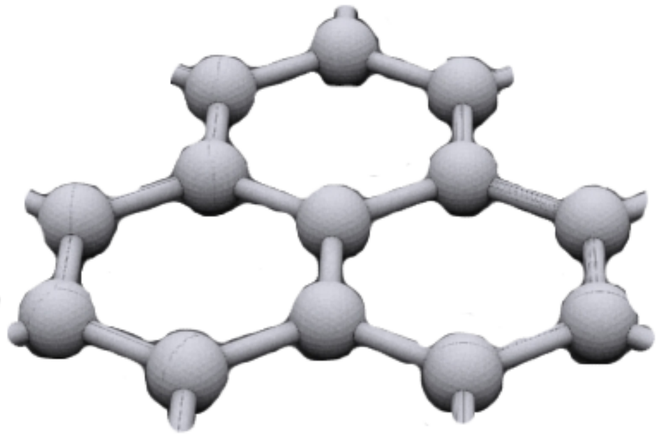
```java
@RunWith(Arquillian.class)
public class GoogleDroneTest {

    @Drone
    WebDriver browser;

    @Test
    public void testOpeningHomePage() {
        browser.get("http://www.google.com/");
        List<WebElement> elements = browser.findElements(
            By.xpath("//span[contains(text(), 'Google Search')]"));

        Assert.assertTrue("Page not loaded", elements.size() > 0);
    }
}
```

GOOGLE DRONE TEST | TEST

```java
@RunWith(Arquillian.class)
public class GoogleDroneTest {

    @Drone
    WebDriver browser;

    @Test
    public void testOpeningHomePage() {
        browser.get("http://www.google.com/");
        List<WebElement> elements = browser.findElements(
            By.xpath("//span[contains(text(), 'Google Search')]"));

        Assert.assertTrue("Page not loaded", elements.size() > 0);
    }
}
```

DEMO TIME

ARQUILLIAN GRAPHENE



Selenium *ARQUILLIAN-STYLE*

GRAPHENE 1.0

GrapheneSelenium class

type-safe Selenium class for Selenium 1.x

```
@Drone
GrapheneSelenium browser;
```

GRAPHENE 1.0

GrapheneSelenium class

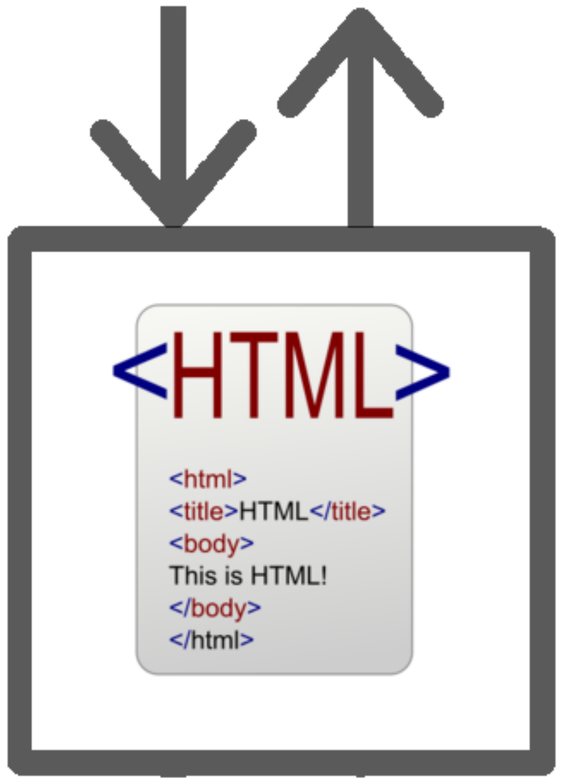
type-safe Selenium class for Selenium 1.x

```
@Drone
GrapheneSelenium browser;
```

GRAPHENE 2.0

More Injection
More Abstractions
More Extensions

PAGE ABSTRACTIONS



Encapsulate DOM elements
behind a custom API

SELENIUM | PAGE OBJECTS

A class that encapsulates the behaviour of a page

SELENIUM | LOGINPAGE

```
public class LoginPage {
    private final WebDriver driver;

    public LoginPage(WebDriver driver) {
        this.driver = driver;
    }

    public HomePage loginAs(String username, String password) {
        driver.findElement(By.id("username")).sendKeys(username);
        driver.findElement(By.id("passwd")).sendKeys(password);
        driver.findElement(By.id("login")).submit();

        return new HomePage(driver);
    }
}
```

SELENIUM | PAGE INSTANTIATION

Selenium PageFactory class instantiates Page Objects:

```
LoginPage loginPage =
    PageFactory.initElements(driver, LoginPage.class);
```

GRAPHENE | PAGE INJECTION

Graphene test enrichment manages the PageFactory

```
@Page
LoginPage loginPage
```

```
@FindBy(id="username")
WebElement username;
```

GRAPHENE | PAGE FRAGMENTS

- Page Objects ++
- Encapsulate the behaviour of a page fragment

```
@FindBy(css=".calendar")
CalendarFragment calendar;
```

ABSTRACTION FTW!

Single place to update WebDriver code
when the underlying DOM changes

WAITING...

Simplest case:

```
Thread.sleep(2000);
```

Selenium provides the WebDriverWait class

```
WebElement myDynamicElement = (new WebDriverWait(driver, 10))
  .until(new ExpectedCondition<WebElement>(){
  @Override
public WebElement apply(WebDriver d) {
  return d.findElement(By.id("myDynamicElement"));
}});
```

More boiler plate!

GRAPHENE WAIT HELPERS

Helper	Description	Timeout
waitGui()	waits for a short time eg. wait for client-side operations	1
waitAjax()	waits for longer time eg. wait for simple ajax request	2
waitModel()	waits for a long time eg. wait for database requests	5

WAIT HELPER USAGE

```
By id = By.id("button");
WebElement element = driver.findElement(id);
...
waitModel(element(id).isVisible));
...
waitModel(element(element).not().textContains("blahblah"));
...
waitModel(attribute(id, "value").valueContains("blahblah"));
...
waitModel(attribute(element, "value").not().valueEquals("blahblah"));
```

REQUEST GUARDS

Blocks the Selenium test execution until a network communication caused by a given action ends

guardHttp(button).click();
blocks on HTTP

guardXhr(button).click();
blocks on XHR (Ajax)

REQUEST GUARD | USAGE

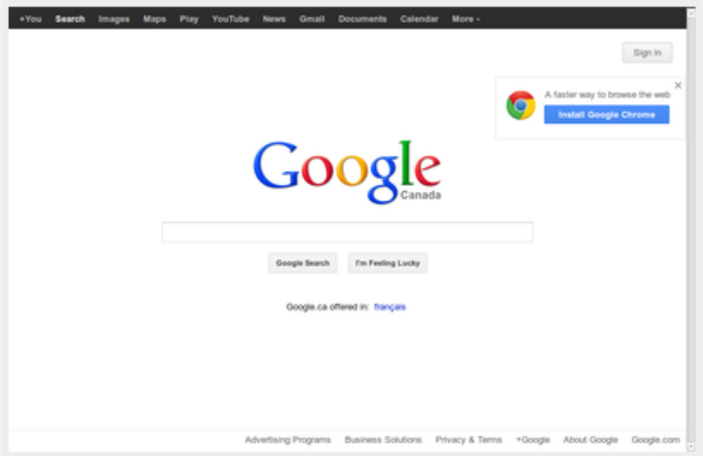
```
@RunWih(Arquillian.class)
public class TestClass {

    @FindBy(id="http")
    private WebElement httpButton;

    @FindBy(id="xhr")
    private WebElement xhrButton;

    @Test
    public void testSimple() {
        guardHttp(httpButton).click();
        guardXhr(xhrButton).click();
    }
}
```

JAVASCRIPT INTERFACES

Execute JavaScript from your Java test

```
@JavaScript("javascriptObject")
@Dependency(sources = {"file.js"})
public interface Background {
    void voidMethod(String color);
    String someFunction();
}
```

EXAMPLE | JAVASCRIPT INTERFACE



- Manipulate the DOM
- Bi-directional communication

EXAMPLE | JAVASCRIPT INTERFACE

BackGround.java:

```
@JavaScript("myBackground")
@Dependency(sources = {"background.js"})
public interface Background {
    void setBackground(String color);
    String getBackground();
}
```

background.js:

```
myBackground = {
    setBackground : function (color) {
        document.body.style.background = color;
    },

    getBackground : function () {
        return document.body.style.background;
    }
}
```

EXAMPLE | JAVASCRIPT INTERFACE

```java
@RunWith(Arquillian.class)
public class JavaScriptTest {

    @Drone
    WebDriver browser;

    @Test
    public void testOpeningHomePage() throws  Exception {
        browser.get("http://www.google.com/");
        Background background = JSInterfaceFactory.create(Background.class);

        System.out.println(String.format(
            "Background color is: %s", background.getBackground()));

        background.setBackground("red");
        System.out.println(String.format(
            "Background color is: %s", background.getBackground()));
        Thread.sleep((2000));

        background.setBackground("");
        System.out.println(String.format(
            "Background color is: %s", background.getBackground()));
        Thread.sleep((2000));
    }
}
```

DEMO TIME

PAGE EXTENSIONS

Automatically include Javascript Interfaces in your tests

Simply:

- Implement an interface
- Register the extension

See the docs...

https://docs.jboss.org/author/display/ARQGRA2/Page+Extensions

REQUEST GUARDS | IMPL

```
@JavaScript(value = "Graphene.Page.RequestGuard")
@Dependency(sources = "Graphene.Page.RequestGuard.js",
            interfaces=XhrInterception.class)
public interface RequestGuard extends InstallableJavaScript {

    /**
     * @return the last request type
     */
    RequestType getRequestDone();

    /**
     * Clears the request type cache and returns the last
     * request type
     * @return the last request type
     */
    RequestType clearRequestDone();

}
```

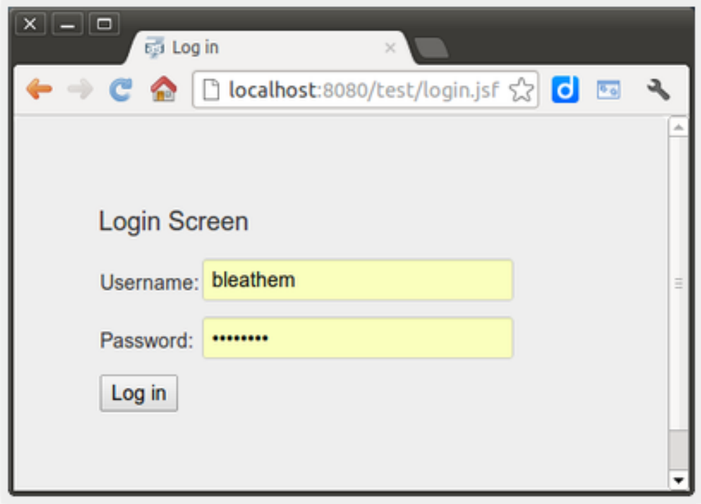
INTERCEPTORS WITH GRAPHENE

Work is in progress to port the Graphene 1 interceptor API to Graphene 2

- Code executed around each WebDriver invocation
- Cross-cutting concerns

https://issues.jboss.org/browse/ARQGRA-79

EXAMPLE | LOGINSCREEN



- Graphene Test
- Demonstrate Page Fragments

LOGINSCREEN | DEPLOYMENT

```
public class Deployments {
    public static final String WEBAPP_SRC = "src/main/webapp";

    public static WebArchive getLoginScreenDeployment() {
        return ShrinkWrap.create(WebArchive.class, "login.war")
                .addClasses(Credentials.class, User.class, LoginController.class)
                .addAsWebResource(new File(WEBAPP_SRC, "resources/bootstrap/css/bootstrap.css"),
                                  "resources/bootstrap/css/bootstrap.css")
                .addAsWebResource(new File(WEBAPP_SRC, "login.xhtml"))
                .addAsWebResource(new File(WEBAPP_SRC, "home.xhtml"))
                .addAsWebInfResource(EmptyAsset.INSTANCE, "beans.xml")
                .addAsWebInfResource(
                        new StringAsset("<faces-config version=\"2.0\"/>"),
                        "faces-config.xml");
    }
}
```

LOGINSCREEN | GRAPHENETEST

```java
@RunWith(Arquillian.class)
public class LoginScreenGrapheneTest {

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    @Test
    @RunAsClient
    public void should_login_successfully() throws Exception {
        String page = contextPath + "login.jsf";
        browser.get(page);

        usernameInput.sendKeys("demo");
        passwordInput.sendKeys("demo");
        loginButton.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

LOGINSCREEN | GRAPHENETEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    @Test
    @RunAsClient
    public void should_login_successfully() throws Exception {
        String page = contextPath + "login.jsf";
        browser.get(page);

        usernameInput.sendKeys("demo");
        passwordInput.sendKeys("demo");
        loginButton.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

LOGINSCREEN | GRAPHENETEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    @Test
    @RunAsClient
    public void should_login_successfully() throws Exception {
        String page = contextPath + "login.jsf";
        browser.get(page);

        usernameInput.sendKeys("demo");
        passwordInput.sendKeys("demo");
        loginButton.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

LOGINSCREEN | GRAPHENETEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    @Test
    @RunAsClient
    public void should_login_successfully() throws Exception {
        String page = contextPath + "login.jsf";
        browser.get(page);

        usernameInput.sendKeys("demo");
        passwordInput.sendKeys("demo");
        loginButton.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

LOGINSCREEN | GRAPHENETEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
```

```
@Test
@RunAsClient
public void should_login_successfully() throws Exception {
    String page = contextPath + "login.jsf";
    browser.get(page);

    usernameInput.sendKeys("demo");
    passwordInput.sendKeys("demo");
    loginButton.click();

    Assert.assertTrue("User should be logged in!",
            browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
}
```

```
}
```

LOGINSCREEN | LOGINFRAGMENT

```
public class LoginFragment {

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    public void setUsername(Object username) {
        usernameInput.sendKeys(username.toString());
    }

    public void setPassword(Object password) {
        passwordInput.sendKeys(password.toString());
    }

    public void click() {
        loginButton.click();
    }
}
```

LOGINSCREEN | LOGINFRAGMENT

```
public class LoginFragment {

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    public void setUsername(Object username) {
        usernameInput.sendKeys(username.toString());
    }

    public void setPassword(Object password) {
        passwordInput.sendKeys(password.toString());
    }

    public void click() {
        loginButton.click();
    }
}
```

LOGINSCREEN | LOGINFRAGMENT

```
public class LoginFragment {

    @FindBy(id="loginForm:username")
    private WebElement usernameInput;

    @FindBy(id="loginForm:password")
    private WebElement passwordInput;

    @FindBy(id="loginForm:login")
    private WebElement loginButton;

    public void setUsername(Object username) {
        usernameInput.sendKeys(username.toString());
    }

    public void setPassword(Object password) {
        passwordInput.sendKeys(password.toString());
    }

    public void click() {
        loginButton.click();
    }
}
```

LOGINSCREEN | FRAGMENTTEST

```java
@RunWith(Arquillian.class)
public class LoginScreenFragmentTest {

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        loginForm.setUsername("demo");
        loginForm.setPassword("demo");
        loginForm.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

LOGINSCREEN | FRAGMENTTEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        loginForm.setUsername("demo");
        loginForm.setPassword("demo");
        loginForm.click();

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
    }

}
```

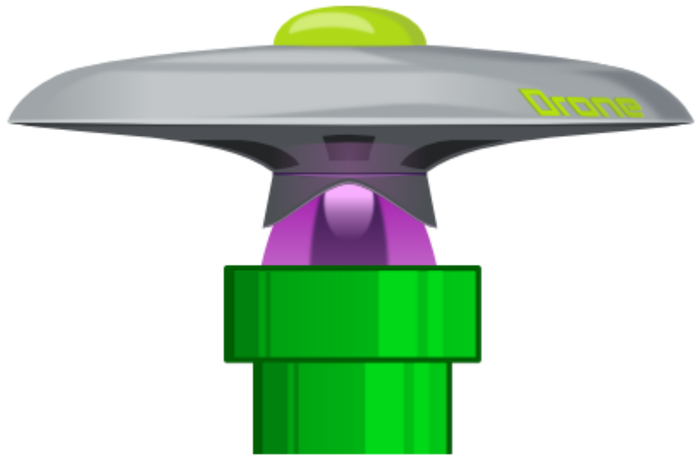
LOGINSCREEN | FRAGMENTTEST

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() {
        return Deployments.getLoginScreenDeployment();
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;
```

```
@Test
@RunAsClient
public void should_login_successfully() {
    String page = contextPath + "login.jsf";
    browser.get(page);

    loginForm.setUsername("demo");
    loginForm.setPassword("demo");
    loginForm.click();

    Assert.assertTrue("User should be logged in!",
            browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]")).size() > 0);
}
```

```
}
```

ARQUILLIAN WARP



- Testing on both sides of the request
- En-rich the client request with a test to run on the server
- En-rich the server response with results of the test

JSFUNIT | THE OLD WAY

- HtmlUnit - no real browsers
- Assert state only at the end of the JSF lifecycle
- JSF Only (No CDI, EJB etc.)

WARP | THE NEW WAY

Advantages:

- Selenium / any HTTP client
- Test the entire JSF lifecycle
- Test any injectable resource
  - CDI
  - EJB
  - Any framework!

WARP | EXAMPLE

```java
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
        webArchive.addAsWebInfResource(new File("src/test/resources/beans.xml"));
        return webArchive;
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        Warp.filter(new JsfRequestFilter()).execute(new ClientAction() {
            @Override
            public void action() {
                loginForm.setUsername("demo");
                loginForm.setPassword("demo");
                loginForm.click();
            }
        }).verify(new CheckUsername());

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]"))
                    .size() > 0);
    }

    public static class CheckUsername extends ServerAssertion {
        @Inject
        Credentials credentials;

        @BeforePhase(Phase.UPDATE_MODEL_VALUES)
        public void beforeUpdateModelValues() {
            Assert.assertNull(credentials.getUsername());
        }

        @AfterPhase(Phase.UPDATE_MODEL_VALUES)
        public void afterUpdateModelValues() {
            Assert.assertEquals("demo", credentials.getUsername());
        }

    }
}
```

WARP | EXAMPLE

```
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
        webArchive.addAsWebInfResource(new File("src/test/resources/beans.xml"));
        return webArchive;
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        Warp.filter(new JsfRequestFilter()).execute(new ClientAction() {
            @Override
            public void action() {
                loginForm.setUsername("demo");
                loginForm.setPassword("demo");
                loginForm.click();
            }
        }).verify(new CheckUsername());

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]"))
                    .size() > 0);
    }

    public static class CheckUsername extends ServerAssertion {
        @Inject
        Credentials credentials;

        @BeforePhase(Phase.UPDATE_MODEL_VALUES)
        public void beforeUpdateModelValues() {
            Assert.assertNull(credentials.getUsername());
        }

        @AfterPhase(Phase.UPDATE_MODEL_VALUES)
        public void afterUpdateModelValues() {
            Assert.assertEquals("demo", credentials.getUsername());
        }

    }
}
```

WARP | EXAMPLE

```
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
        webArchive.addAsWebInfResource(new File("src/test/resources/beans.xml"));
        return webArchive;
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        Warp.filter(new JsfRequestFilter()).execute(new ClientAction() {
            @Override
            public void action() {
                loginForm.setUsername("demo");
                loginForm.setPassword("demo");
                loginForm.click();
            }
        }).verify(new CheckUsername());

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]"))
                    .size() > 0);
    }

    public static class CheckUsername extends ServerAssertion {
        @Inject
        Credentials credentials;

        @BeforePhase(Phase.UPDATE_MODEL_VALUES)
        public void beforeUpdateModelValues() {
            Assert.assertNull(credentials.getUsername());
        }

        @AfterPhase(Phase.UPDATE_MODEL_VALUES)
        public void afterUpdateModelValues() {
            Assert.assertEquals("demo", credentials.getUsername());
        }

    }
}
```

WARP | EXAMPLE

```
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
        webArchive.addAsWebInfResource(new File("src/test/resources/beans.xml"));
        return webArchive;
    }
```

```
@Drone
WebDriver browser;

@ArquillianResource
URL contextPath;

@FindBy(id="loginForm")
LoginFragment loginForm;
```

```
    @Test
    @RunAsClient
    public void should_login_successfully() {
        String page = contextPath + "login.jsf";
        browser.get(page);

        Warp.filter(new JsfRequestFilter()).execute(new ClientAction() {
            @Override
            public void action() {
                loginForm.setUsername("demo");
                loginForm.setPassword("demo");
                loginForm.click();
            }
        }).verify(new CheckUsername());

        Assert.assertTrue("User should be logged in!",
                browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]"))
                    .size() > 0);
    }

    public static class CheckUsername extends ServerAssertion {
        @Inject
        Credentials credentials;

        @BeforePhase(Phase.UPDATE_MODEL_VALUES)
        public void beforeUpdateModelValues() {
            Assert.assertNull(credentials.getUsername());
        }

        @AfterPhase(Phase.UPDATE_MODEL_VALUES)
        public void afterUpdateModelValues() {
            Assert.assertEquals("demo", credentials.getUsername());
        }

    }
}
```

WARP | EXAMPLE

```
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
```

```
@Test
@RunAsClient
public void should_login_successfully() {
    String page = contextPath + "login.jsf";
    browser.get(page);

    Warp.filter(new JsfRequestFilter()).execute(new ClientAction() {
        @Override
        public void action() {
            loginForm.setUsername("demo");
            loginForm.setPassword("demo");
            loginForm.click();
        }
    }).verify(new CheckUsername());

    Assert.assertTrue("User should be logged in!",
            browser.findElements(By.xpath("//li[contains(text(), 'Welcome')]"))
                .size() > 0);
}
```

```
public static class CheckUsername extends ServerAssertion {
    @Inject
    Credentials credentials;

    @BeforePhase(Phase.UPDATE_MODEL_VALUES)
    public void beforeUpdateModelValues() {
        Assert.assertNull(credentials.getUsername());
```

WARP | EXAMPLE

```java
@WarpTest
@RunWith(Arquillian.class)
public class LoginScreenWarpTest {

    @Deployment
    public static WebArchive createDeployment() {
        WebArchive webArchive = Deployments.getLoginScreenDeployment();
        webArchive.delete("WEB-INF/beans.xml");
        webArchive.addAsWebInfResource(new File("src/test/resources/beans.xml"));
        return webArchive;
    }

    @Drone
    WebDriver browser;

    @ArquillianResource
    URL contextPath;

    @FindBy(id="loginForm")
    LoginFragment loginForm;

    public static class CheckUsername extends ServerAssertion {
        @Inject
        Credentials credentials;

        @BeforePhase(Phase.UPDATE_MODEL_VALUES)
        public void beforeUpdateModelValues() {
            Assert.assertNull(credentials.getUsername());
        }

        @AfterPhase(Phase.UPDATE_MODEL_VALUES)
        public void afterUpdateModelValues() {
            Assert.assertEquals("demo", credentials.getUsername());
        }
    }
}
```

DEMO TIME

REDUCING TEST DEVELOPMENT TURNAROUND

Best practices

REDUCING TEST DEVELOPMENT TURNAROUND

Best practices

Use a Remote Container

REDUCING TEST DEVELOPMENT TURNAROUND

Best practices

Use a Remote Container
Re-usable Browser Session

REDUCING TEST DEVELOPMENT TURNAROUND

Best practices

- Use a Remote Container
- Re-usable Browser Session
- Use Shrinkwrap micro-deployments
- Share deployments across tests

REDUCING TEST DEVELOPMENT TURNAROUND

Best practices

- Use a Remote Container
- Re-usable Browser Session
- Use Shrinkwrap micro-deployments
- Share deployments across tests

Arquillian jRebel extension

THE END?

THE END?

THE BEGINNING - OF JSF TESTING ADVENTURE WITH ARQUILLIAN AND SELENIUM!

CREDITS

http://www.flickr.com/photos/jdhancock/5845280258/
http://leslycorazon.wikispaces.com/file/detail/head-silhouette-with-question-mark.png/319199232
http://watirmelon.files.wordpress.com/2012/01/idealautomatedtestingpyramid.png
http://watirmelon.files.wordpress.com/2012/01/softwaretestingicecreamconeantipattern.png
http://www.flickr.com/photos/cellardoorfilms/7620375702/
http://www.flickr.com/photos/pasukaru76/6893926948/
http://www.flickr.com/photos/metrolibraryarchive/3267555668/
http://www.flickr.com/photos/metrolibraryarchive/3267249336/
http://www.flickr.com/photos/core-materials/5057399792/
http://www.flickr.com/photos/abennett96/2717629123/

ADDITIONAL RESOURCES

- Arquillian Getting Started guides
  http://arquillian.org/guides/getting_started/

- Arquillian Extensions:
  - Drone docs
    https://docs.jboss.org/author/display/ARQ/Drone
  - Graphene 2 docs
    https://docs.jboss.org/author/display/ARQGRA2/Getting+Started
  - Warp docs
    https://docs.jboss.org/author/display/ARQ/Warp

- Sample Code
  https://github.com/bleathem/TestingJSF

ARQUILLIAN @ JAVAONE

TUT5039 – Cover Your Web and Mobile Applications with Integration Tests from Scratch

CON7469 – Apache TomEE, a Java EE 6 Web Profile on Tomcat

CON2818 – A Bird's-Eye View of the CDI Ecosystem

CON5312 – Continuous Enterprise Development: Case Studies in Java EE Integration Testing

CON6918 – The Arquillian Universe: A Tour Around the Astrophysics Lab

BOF7997 – A New Approach to Testing Your Java EE Applications Deployed on a PaaS

CON7622 – Testing JSF Applications with Arquillian and Selenium

CON5458 – Today's Rapid Java EE Development: Live Coding from Scratch to Deployment

STAY IN THE LOOP

Project	arquillian.org	richfaces.org
Twitter:	@arquillian	@richfaces
Google+:	+Arquillian	+RichFaces
Forums:	Arquillian User forum	RichFaces User forum
IRC	#jbosstesting	#richfaces
Blog feed:	arquillian.org/blog/	planet.jboss.org /feed/richfacesall