RichFaces – Testing on Mobile Devices

Pavol Pitoňák
Quality Assurance Engineer, Red Hat
17 Feb 2012

Why Are We Doing This?

- canalys.com

487.7 mil

414.6 mil

Maven

RichFaces

Arquillian

Jenkins

TestNG

Android

Selenium

JBoss AS

Eclipse

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Testing a Java Web Application

1. Build archive

2. Start a container

3. Deploy archive

4. Open web browser

5. Run tests (continuously)

6. Review test results

RichRates

- github.com/qa/richrates

- richrates-ppitonak.rhcloud.com

richrates-ppitonak.rhcloud.com

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

richrates-ppitonak.rhcloud.com

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

richrates-ppitonak.rhcloud.com

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

richrates-ppitonak.rhcloud.com

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

richrates-ppitonak.rhcloud.com

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Project Structure

- Standard Maven structure

- testng.xml

- testng-ftest.xml

- testng-all.xml

Running Unit Tests

mvn clean package

testng.xml

Running Functional Tests

mvn clean package -P jbossas-managed-71,ftest

testng-ftest.xml

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Running All Tests

mvn clean package -P jbossas-managed-71,all-tests

testng-all.xml

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Sample Unit Test

```
public class UAgentInfoTest {

    @Test

    public void testIPad() {

        Assert.assertTrue(...);

    }

}
```

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Sample Arquillian Test

```
public class SkinBeanTest extends Arquillian {

    @Inject private SkinBean skinBean;

    @Deployment

    public static JavaArchive createTestArchive() {

        return ...;

    }

    @Test

    public void testDefaultSkin() {

        Assert.assertEquals(skinBean.getSkin(), "blueSky");

    }

}
```

Sample WebDriver Test

```
public class SampleTest extends Arquillian {

    @Drone public WebDriver driver;

    @ArquillianResource private URL deployedRoot;

    @Deployment

    public static WebArchive createTestArchive() {

        return ...;

    }

    @Test

    public void testDefaultSkin() {

        driver.get("http://devconf.cz");

        ...

    }

}
```

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Deployment

```
@Deployment(testable = false)

public static WebArchive createTestArchive() {

        WebArchive war =
ShrinkWrap.createFromZipFile(WebArchive.class, new
File("target/richrates.war"));

 }
```

... or construct the WAR yourself :)

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Deployment

```
@Deployment(testable = false)
public static WebArchive createTestArchive() {
        WebArchive war = ShrinkWrap.createFromZipFile(WebArchive.class, new File("target/richrates.war"));
 }
```

... or construct the WAR yourself :)

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Deployment 2

```java
@Deployment(testable = false)

    public static WebArchive createTestArchive() {

        MavenDependencyResolver resolver = DependencyResolvers.use(MavenDependencyResolver.class).loadMetadataFromPom( "pom.xml");

        WebArchive war = ShrinkWrap.create(WebArchive.class, "richrates.war");

        war.addPackage("org.richfaces.examples.richrates");

        war.addPackage("org.richfaces.examples.richrates.annotation");

        war.addPackages(true, "org.slf4j");


        war.merge(ShrinkWrap.create(ExplodedImporter.class, "tmp1.war").importDirectory("src/main/webapp").as(WebArchive.class));

        war.merge(ShrinkWrap.create(ExplodedImporter.class, "tmp2.war").importDirectory("src/main/resources").as(WebArchive.class), "WEB-INF/classes");


        war.addAsLibraries(resolver.artifact("org.richfaces.ui:richfaces-components-ui").resolveAsFiles());

        war.addAsLibraries(resolver.artifact("org.richfaces.core:richfaces-core-impl").resolveAsFiles());

        war.addAsLibraries(resolver.artifact("org.jdom:jdom").resolveAsFiles());

        war.addAsLibraries(resolver.artifact("joda-time:joda-time").resolveAsFiles());

        return war;

    }
```

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Arquillian.xml

```xml
<extension qualifier="webdriver">

    <property
name="implementationClass">org.openqa.selenium.fir
efox.FirefoxDriver</property>

</extension>
```

-Darquillian.browser.implementation.class=...

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Arquillian.xml

- Define your containers

- Maven profiles

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Sample WebDriver Test

```
public class SampleTest extends Arquillian {

    @Drone public WebDriver driver;

    @ArquillianResource private URL deployedRoot;

    @Deployment

    public static WebArchive createTestArchive() {

        return ...;

    }

    @Test

    public void testDefaultSkin() {

        driver.get("http://devconf.cz");

        ...

    }

}
```

Page Factory

```
@Test

public void googleSearchTest {

    WebDriver driver = ...

    driver.get("http://www.google.com/");

    GoogleSearchPage page =
PageFactory.initElements(driver,GoogleSearchPage.class);

    page.searchFor("Cheese");

}
```

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Page Factory

Page Object

```
public class CalculatorPage extends AbstractPage {

    @FindBy(id = "calculator:amount")
    private WebElement amountInput;

    @FindBy(xpath = "//div[contains(@class, 'result')]")
    private WebElement resultOutput;

    @FindBy(id = "calculator:calculateButton")
    private WebElement submitButton;

    ...

}
```

Mobile Web Testing

- Almost no changes in Arquillian config

- Android 1.x not supported
- Android 2.x – phones
- Android 3.x – tablets
- Android 4.x – phones and tablets

- Physical device vs. emulator

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Physical Devices

- Windows – install USB drivers

- Linux – set up udev

- Mac OS X – no setup needed

Android Emulator

- Part of Android Software Development Kit

- Android 1.5 – 4.0.3

- <sdk_path>/tools/android

- Eclipse plugins

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Emulators

- 2.2
  - Relatively fast
  - Works with selenium-server 2.6
- 2.3
  - Unreliable
- 3.x
  - Slow
- 4.x
  - Relatively fast
  - Works well with newest Selenium

Demo

Creating and launching a new emulator

Android-x86 Project

- www.android-x86.org

- Project porting Android to x86 platform

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Demo

Launching Android in VirtualBox

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

AndroidDriver Config

```
<extension qualifier="webdriver">

<property name="implementationClass">
org.openqa.selenium.android.AndroidDriver
</property>

<property name="remoteAddress">

http://localhost:4444/wd/hub

</property>

</extension>
```

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Emulator Configuration

- Install android-server.apk

- Launch WebDriver application in emulator

- platform-tools/adb forward tcp:4444 tcp:8080

- Use 10.0.2.2 for host's localhost

VirtualBox Configuration

- Set up network

- platform-tools/adb connect <android_ip>

- Install android-server.apk

- Launch WebDriver application in emulator

- platform-tools/adb forward tcp:4444 tcp:8080

- Tests need server IP address

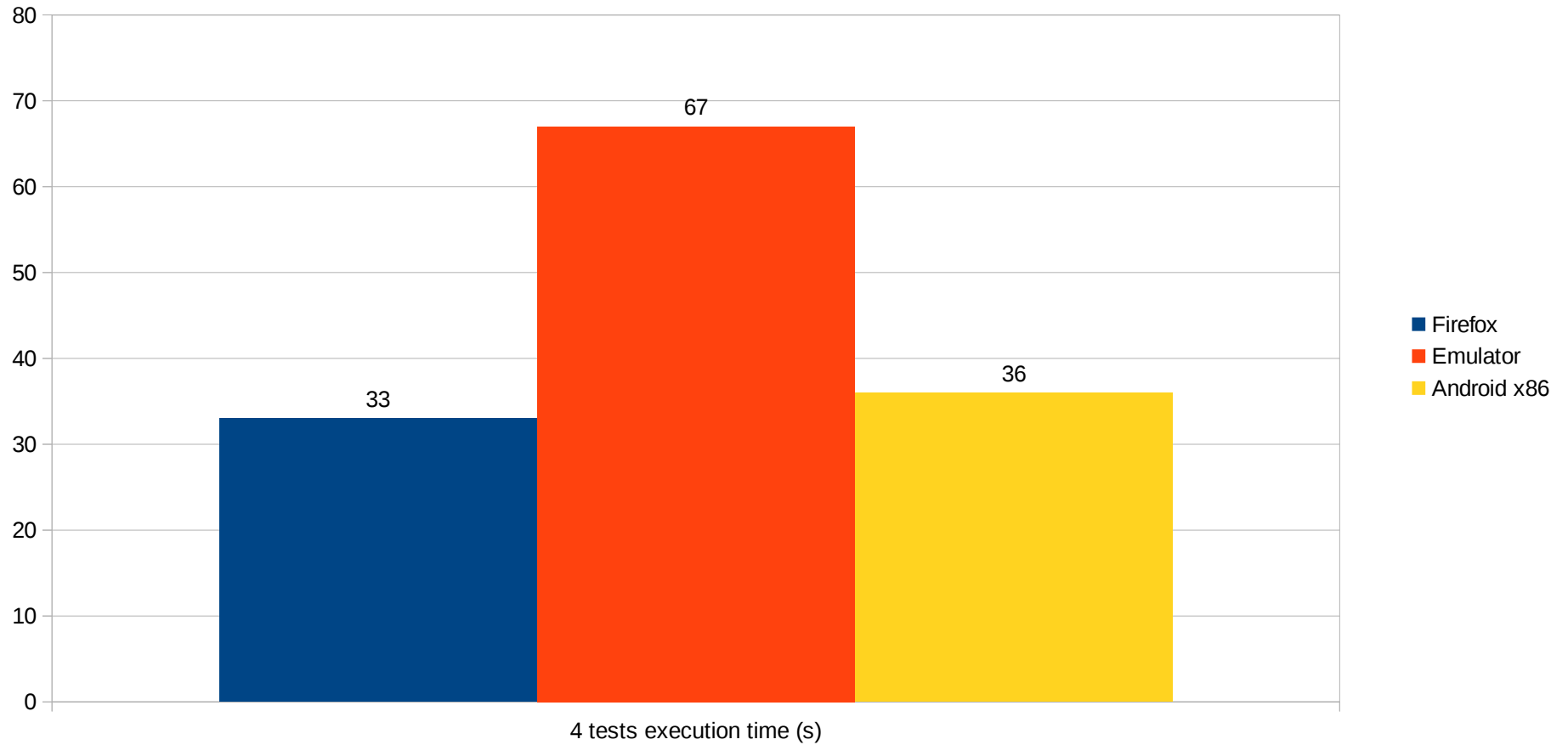**TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK**

Demo

Tests running with Firefox, Android emulator and Android-x86

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Fast, Faster, Fastest

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Eclipse

- Plugins
  - Android
  - JBoss Tools
  - m2e
  - TestNG

- Supports both emulator and Android-x86

Demo

Running tests in Eclipse

Debugging

- You already know how :)

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Demo

Debugging

Arquillian Android Extension

- github.com/kpiwko/arquillian-extenstion-android

- Hot news!

- Create emulator

- Start emulator

- Install Android server

- Start Android server

- Can connect to real device

Console Logging Listener

- Usual console log:

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Console Logging Listener

- Our console log:

[10:47:54] STARTED: ftest.ExchangeRatesTest.testInit()

[10:47:54] SUCCESS: ftest.ExchangeRatesTest.testInit()

[10:47:55] STARTED: ftest.ExchangeRatesTest.testNextAndPrevious()

[10:47:57] SUCCESS: ftest.ExchangeRatesTest.testNextAndPrevious()

[10:48:09] STARTED: ftest.CalculatorTest.testCalculate()

[10:48:12] SUCCESS: ftest.CalculatorTest.testCalculate()

Console Logging Listener

- Register in pom.xml or in test class

    @Listeners(ConsoleStatusTestListener.class)

    public abstract class AbstractWebDriverTest<P extends Page> extends Arquillian {

    ...

    }

Screenshot on Failure

```
@AfterMethod(alwaysRun = true, groups = "arquillian")

public void handleTestError(ITestResult result) {

    if (result.getStatus() == ITestResult.SUCCESS) {

        return;

    }

    ...

    File scrFile = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);

    FileUtils.copyFile(scrFile, imageOutputFile);

}
```

Jenkins Android Plugin

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Jenkins Android Plugin

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK
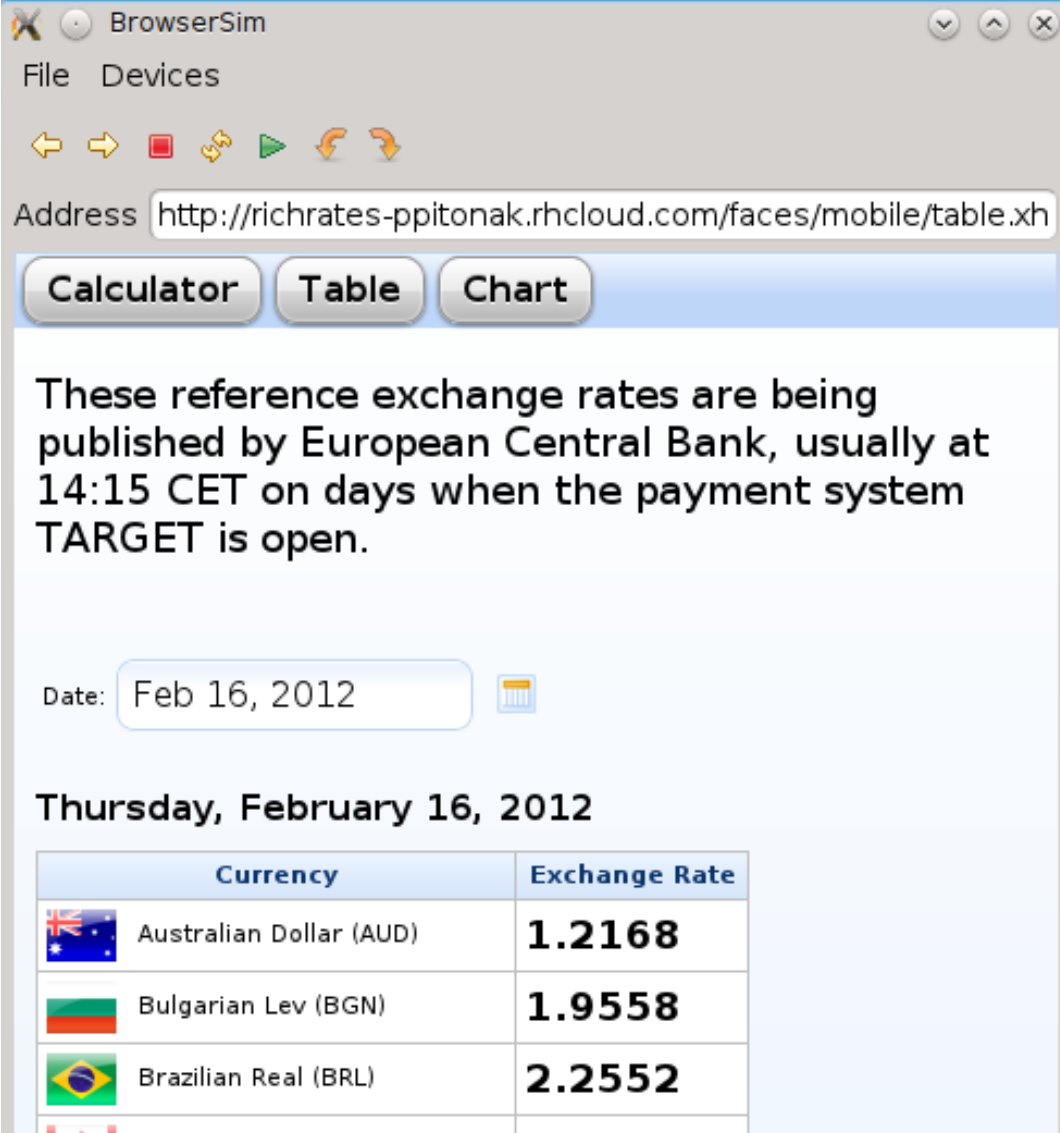
Mobile Browser Simulator

- JBoss Tools 3.3 M5

- Simulates mobile phones and tablets

- Perfect for layout testing

- Not for automation

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Mobile Browser Simulator

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

Conclusion

- Arquillian + Selenium 2

- Run from console and Eclipse

- How to create Android emulator

- Android-x86

- Emulator snapshots

- ID locators, Xpath locators, CSS locators

TESTING ON MOBILE DEVICES | PAVOL PITOŇÁK

ppitonak@redhat.com
@ppitonak

jboss.org/richfaces

Q/A