

11 mars 2008



Paris JUG

www.parisjug.org





11 mars 2008

Selenium

Un outil de tests fonctionnels

Zouheir CADI

Developpeur Senior

Novedia (Smart-up)



www.parisjug.org



Selenium : un outil de tests fonctionnels

application en JAVA/J2EE

Intervenant

- **Zouheir CADI**

- Ingénieur informaticien depuis 2000
- Développeur Java depuis 2002
- Nombreuses missions avec utilisation de la méthodologie Test Driven Development
- Actuellement chez Novedia (Smart-up)

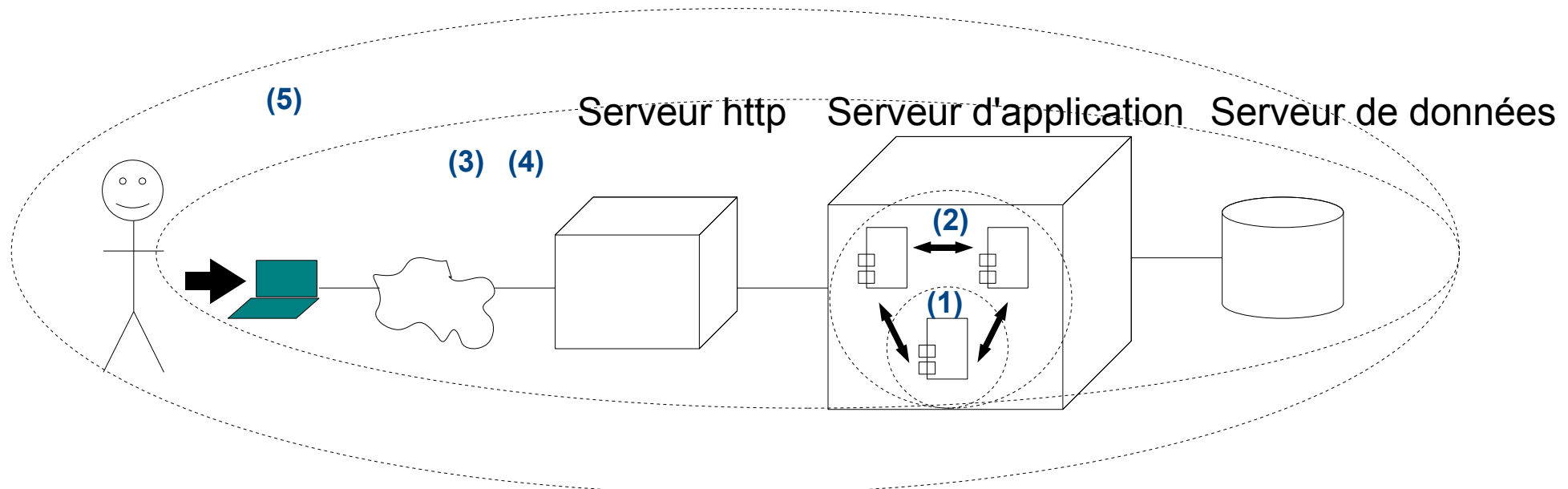
Sommaire

- **Introduction**
- **Les composants de Selenium**
 - **Selenium core**
 - **Selenium IDE**
 - **Selenium Remote Control**
 - **Selenium & JAVA**
- **Démonstration**
- **Conclusion**

Introduction

• Les différents types de test

- Tests unitaires : est ce qu'un composant pris individuellement fonctionne (1) ?
- Tests d'intégration : est ce que les composants fonctionnent (2) ?
- Tests fonctionnels : est ce que l'application fonctionne (3) ?
- Tests de performance : est ce que l'application fonctionne bien (en terme de charge, temps de réponse, ... (4))?
- Tests d'acceptance : est ce que l'utilisateur est satisfait de l'application (5) ?



Introduction

- **Une revue non exhaustive des outils**
 - Outils d'éditeurs
 - Suite d'HP (Mercury) : Quick test pro, test director, WinRunner
 - Rational Functional Tester
 - Outils open source (web testing tools)
 - HtmlUnit, CanooWebTests, JwebUnit
 - HttpUnit
 - SAHI
 - WATIJ
 - Fitness
 - StoryTestIQ (STIQ)

Selenium core

■ Généralités

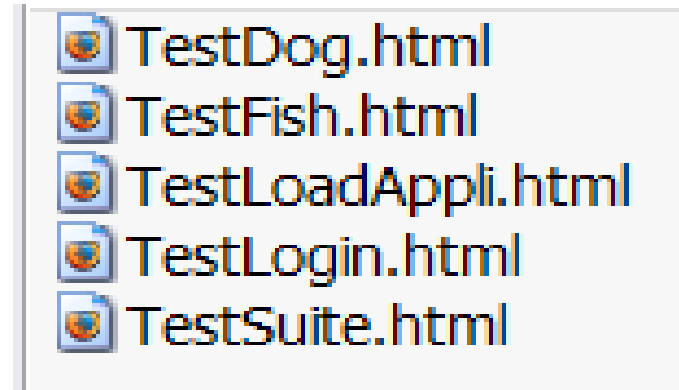
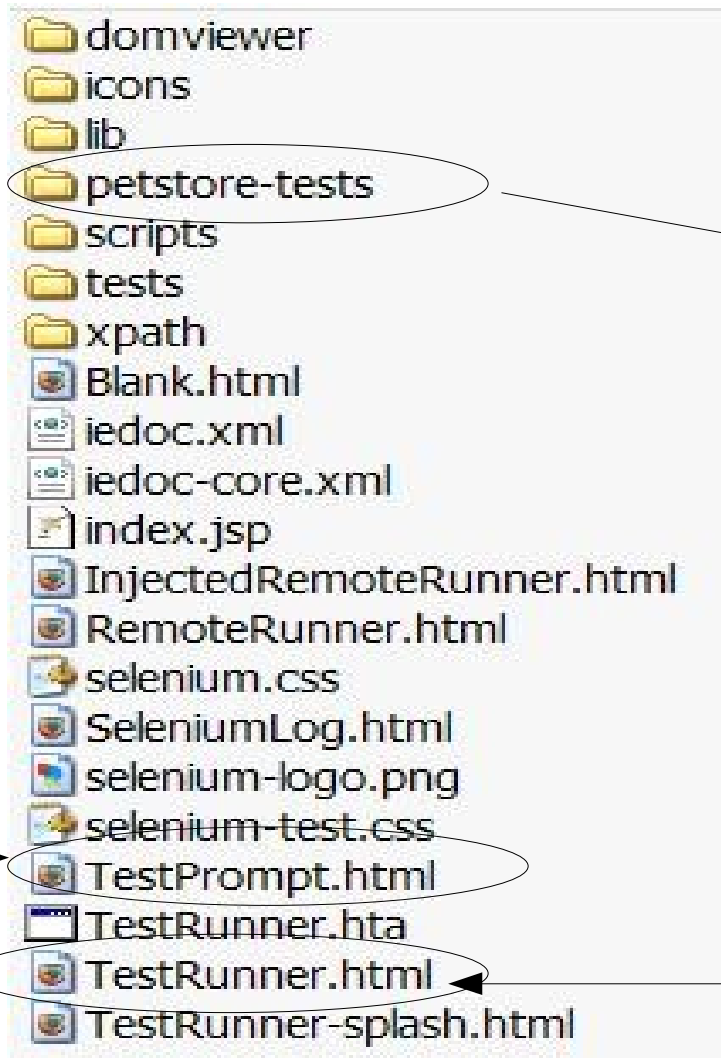
- Utilise javascript et des iFrames pour incorporer un moteur de tests dans le navigateur
- Principe de fonctionnement compatible avec tout navigateur où l'exécution de javascript est autorisée
- Supporte plusieurs types de navigateurs
 - IE 6 et 7, Firefox 0.8 à 2.0, Opera 8 et 9, Seamonkey 1.0, Safari (Mac OS X)
- Langage permettant de réaliser les tests : Selenese
- Langage de script sans conditionnel ni boucle

Selenium core

- **Installation**

- Installation coté serveur !
 - au niveau du serveur http
 - au niveau du serveur d'application
- fichier compressé à dézipper
- version actuelle : 0.8.3

Selenium core



Point d'entrée de l'application

Résultat de la décompression de selenium-core

Selenium core

Selenium Functional Test Runner v0.8.3 [1879] - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

http://localhost/TestRunner.html

Débuter avec Firefox À la une

Test Suite:
./petstore-tests/TestSuite.html
Go

Options

- AUT in separate window
- Run automatically

Selenium TestRunner

Execute Tests

Fast Slow

Highlight elements

Elapsed: 00.00

Tests	Commands
● run	● passed
● failed	● failed
	● incomplete

Tools

View DOM Show Log

↑
Test Suite

↑
Current Test

↑
Control Panel

Terminé

Appel de la page TestRunner.html hébergée dans un serveur apache

Selenium core

Selenium Functional Test Runner v0.8.3 [1879] - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

http://localhost/TestRunner.html?test=.%2Fpetstore-tests%2FTestSuite.html&resultsUrl=..%2FpostF

Débuter avec Firefox À la une

Test Suite

- Test load application
- Test dog
- Test fish

Not supported in this browser

Skipped Tests

TestLoadAppliPetstore

open /petstore

Selenium TestRunner

Execute Tests

Fast Slow

Highlight elements

Elapsed: 00.00

Tests **Commands**

- run passed
- failed failed
- incomplete

Tools

View DOM Show Log

↑

Test Suite **Current Test** **Control Panel**

Terminé

Chargement de la page contenant les tests

Selenium core

- **Présentation de code**
 - **TestSuite.html**

```
<tbody>
  <tr><td><b>Test Suite</b></td></tr>
  <tr>
    <td><a href=" ./TestLoadAppli.html">Test load</a></td>
  </tr>
  <tr>
    <td><a href=" ./TestDog.html">Test dog</a></td>
  </tr>
  <tr>
    <td><a href=" ./TestFish.html">Test fish</a></td>
  </tr>
</tbody>
```

Selenium core

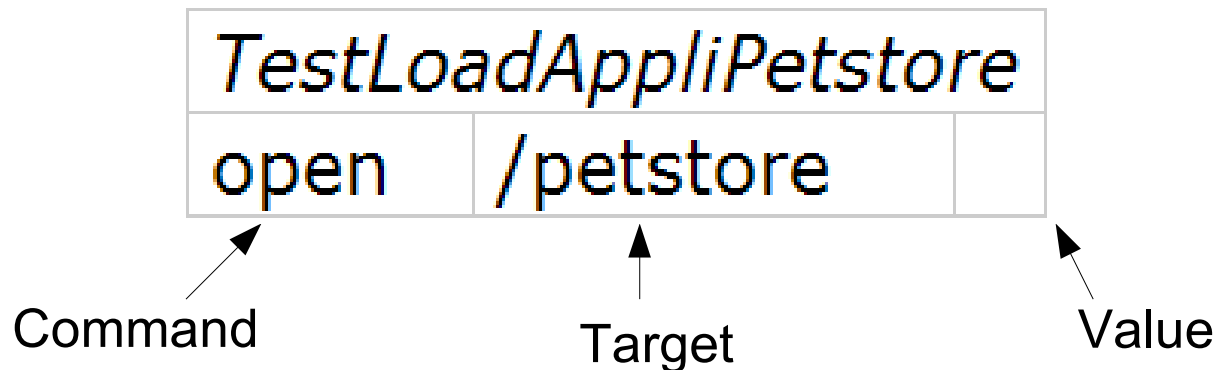
- **Présentation de code**
 - TestLoadAppli.html

```
<table cellpadding="1" cellspacing="1" border="1">  
  <tbody>  
    <tr>  
      <td rowspan="1" colspan="3">TestLoadAppliPetstore<br>  
    </td>  
  </tr>  
  <tr>  
    <td>open</td>  
    <td>/petstore</td>  
    <td>&nbsp;</td>  
  </tr>  
</tbody>  
</table>
```

Selenium core

- **Selenese**

- Instruction élémentaire : **command**
 - Actions
 - Accessors
 - Assertions
- Locators



Selenium core

▪ Actions

- Simulent une interaction de l'utilisateur avec l'application web (le navigateur)
- `<action>AndWait` : action correspondante avec attente
- Ont une cible
- Exemple
 - `open` : ouvre la page à partir d'une url
 - `type` : remplit un `textField/textArea`
 - `select` : sélectionne un élément dans une liste déroulante
 - `check` : valide une case à cocher
 - `click` : clique sur un lien, un bouton, ...
 - `submit` : soumet un formulaire

Selenium core

▪ Accessors - Assertions

- Inspectent l'état d'une application pour
 - enregistrer le resultat dans une valeur : accessors
 - `store<Xxx>(locator, variable)`
 - Exemple
 - `alert`, `bodyText`, `eval`, `text`, `title`
 - vérifier que la valeur correspond à un état pré-défini : assertions
 - `verify<Xxx>` et `verifyNot<Xxx>(locator, pattern)`
 - `assert<Xxx>` et `assertNot<Xxx>(locator, pattern)`
 - `waitFor<Xxx>` et `waitForNot<Xxx>(locator, pattern)`

Selenium core

▪ Locators (1)

- Indiquent à Selenium quelle est la cible de la commande
- format *{locator-prefix}={locator-string}*
- *identifier=theIdentifier*
- *id=theId*
- *name=oneName*
- *dom=javascriptExpression*
- *xpath=xpathExpression*
- *link=textPattern*
- *css=cssSelectorSyntax*

Selenium core

▪ Locators (2)

- Si *{locator-prefix}* non précisé, Selenium applique la stratégie par défaut suivante :
 - *dom* pour les locators qui commencent par `<document.>`
 - *xpath* pour les locators qui commencent par `<//>`
 - *identifier* pour les autres cas

Selenium core

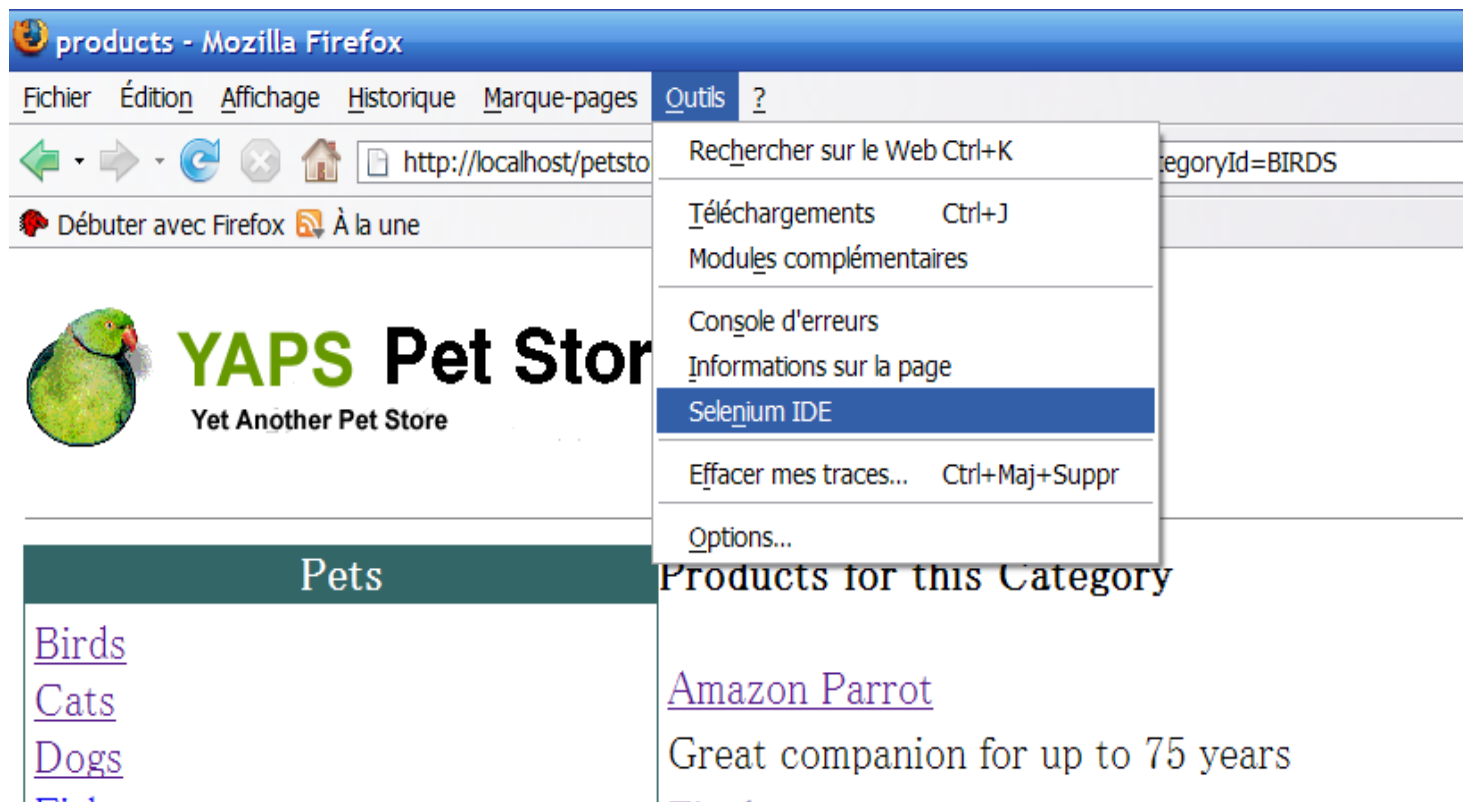
- **Selenium : approche critique**

- Doit être installé au sein du serveur !
- Pas d'automatisation (intégration continue)
- Langage ne permettant pas de bénéficier de la richesse des IDE (refactoring par exemple)

Selenium IDE

- Environnement de développement logiciel pour les tests Selenium
- Se présente sous la forme d'un add-on Firefox
- Permet d'enregistrer, d'éditer et de réexécuter une navigation
- En saisie manuelle, fournit « l'autocomplete » pour toutes les commande Selenium
- Possibilité d'enregistrer les tests en
 - en HTML (langage de base pour les scripts Selenese)
 - en java

Selenium IDE



Lancement de selenium IDE (1)

Selenium IDE

Selenium IDE *
Fichier Édition Options ?
Base URL http://localhost/petstore
Lancer Pas-à-pas Faire un pas
Table Source

Commande	Cible	Valeur
open	/petstore/	

Commande
Cible
Valeur
Rechercher

Log Référence

open(url)
Arguments:
♦ url - the URL to open; may be relative or absolute
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load before proceeding, ie. the "AndWait" suffix is implicit. *Note:* The URL must be on the same domain as the runner HTML

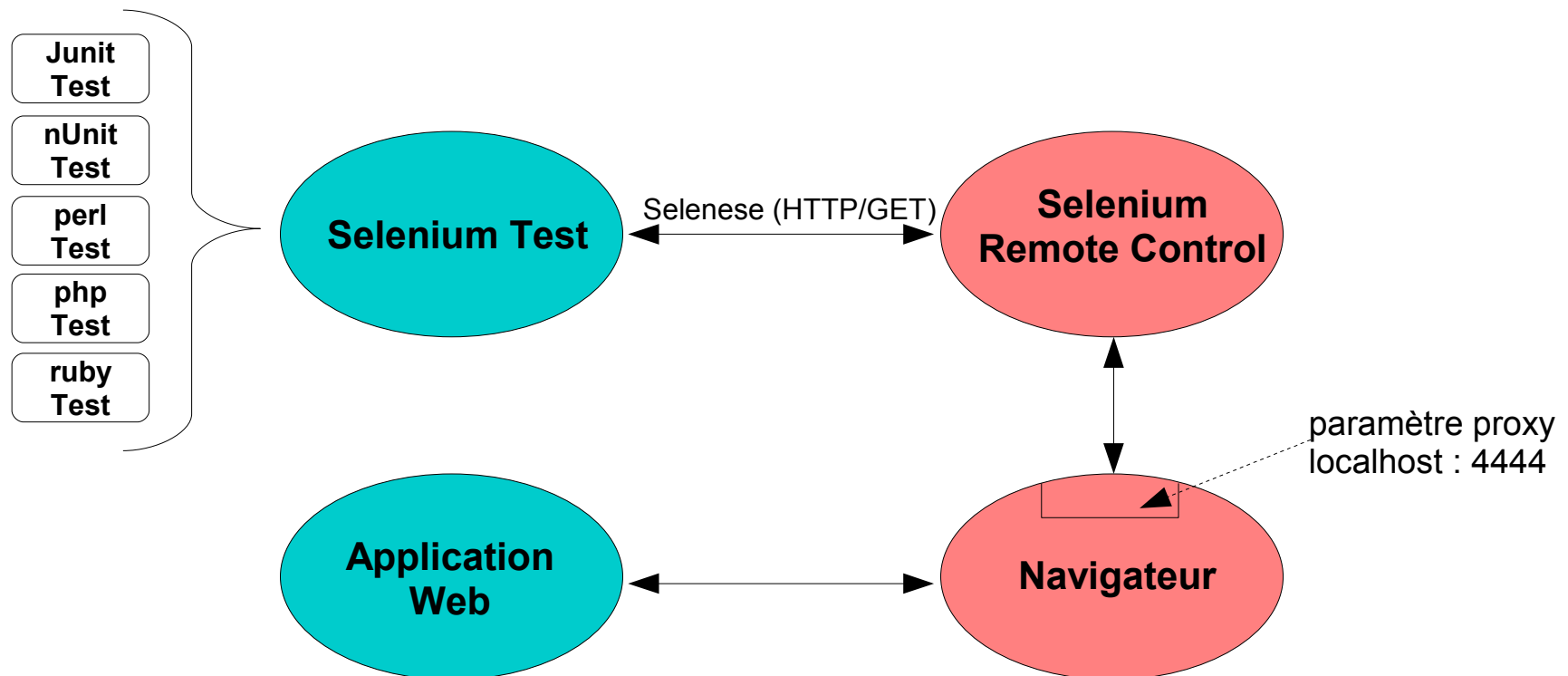
Lancement de selenium IDE (2)

Selenium Remote Control

Architecture

Constitué de deux parties

- serveur (sous forme de jar)
- API clientes (JAVA, autres ...)



Source : http://selenium-grid.openqa.org/how_it_works.html

Selenium Remote Control

- **Modes de lancement**
 - **Usage**
 - `java -jar selenium-server.jar [-interactive] [options]`
 - **Interactif**
 - `java -jar selenium-server.jar -interactive`
 - **Multiwindow**
 - `java -jar selenium-server.jar -multiWindow`
 - **Choix du port d'écoute**
 - `java -jar selenium-server.jar -port <xxxx>`
 - Par défaut 4444

Selenium Remote Control

- **Support du protocole https**
 - **navigation mixte http/https**
 - **lanceurs de navigateur particulier**
 - ***chrome : pour Firefox**
 - ***iehta : pour Internet Explorer**
 - **mode proxyInjection**
 - ***pifirefox : pour Firefox**
 - ***piiexplore : pour Internet Explorer**

Selenium & JAVA

- **Maven**

- **jar serveur et api clientes dans un repository**
 - <http://maven.openqa.org>
- **plugin pour piloter le server**
 - **selenium-maven-plugin**

Selenium & JAVA

- **Configuration de maven**
 - **Repository d'OpenQA**

```
<repositories>
  <repository>
    <id>openqa</id>
    <url>http://maven.openqa.org</url>
    <layout>default</layout>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
    <releases>
      <enabled>>true</enabled>
    </releases>
  </repository>
</repositories>
```

Selenium & JAVA

- **Configuration de maven**
 - **Dépendances**

```
<dependencies>
  <dependency>
    <groupId>org.openqa.selenium.client-drivers</groupId>
    <artifactId>selenium-java-client-driver</artifactId>
    <version>${selenium.version}</version>
  </dependency>
  <dependency>
    <groupId>org.openqa.selenium.server</groupId>
    <artifactId>selenium-server</artifactId>
    <version>${selenium.version}</version>
  </dependency>
</dependencies>
```

Selenium & JAVA

- **Configuration de maven**
 - **Arrêt/Relance du serveur Selenium (1)**

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>selenium-maven-plugin</artifactId>
  <version>1.0-beta-2</version>
  <executions>
    <execution>
      <!-- START SERVER -->
    </execution>
    <execution>
      <!-- STOP SERVER -->
    </execution>
  </executions>
</plugin>
```

Selenium & JAVA

- **Configuration de maven**
 - **Arrêt/Relance du serveur Selenium (2)**

```
<!-- START SERVER -->
<execution>
  <id>start-selenium</id>
  <phase>pre-integration-test</phase>
  <goals>
    <goal>start-server</goal>
  </goals>
  <configuration>
    <background>true</background>
    <port>8888</port>
  </configuration>
</execution>
```

Selenium & JAVA

- **Configuration de maven**
 - **Arrêt/Relance du serveur Selenium (3)**

```
<!-- STOP SERVER -->
<execution>
  <id>stop-selenium</id>
  <phase>post-integration-test</phase>
  <configuration />
  <goals>
    <goal>stop-server</goal>
  </goals>
</execution>
```


Selenium & JAVA

- Classe de test java
 - lancer le navigateur

```
protected static final String url = "http://localhost/";  
protected static final String FIREFOX = "*custom C:/Program  
Files/Mozilla Firefox/firefox.exe";  
protected DefaultSelenium browser;  
  
//lancer navigateur  
public void setUp() throws Exception {  
    browser = new DefaultSelenium("localhost", 8888, FIREFOX, url);  
    browser.start();  
}
```

Selenium & JAVA

- **Classe de test java**
 - **arrêter le navigateur**

```
protected static final String url = "http://localhost/";  
protected static final String FIREFOX = "*custom C:/Program  
Files/Mozilla Firefox/firefox.exe";  
protected DefaultSelenium browser;  
  
//arreter navigateur  
public void tearDown() throws Exception {  
    browser.stop();  
}
```

Selenium & JAVA

- **Classe de test java**
 - **un test : démarrer une application**

```
public void testLoadWebApplication() throws Exception{  
    browser.open("http://localhost/petstore");  
    browser.waitForPageToLoad(WAIT_TIME);  
    assertEquals(myResources.  
        getString("index.title"), browser.getText("//title"));  
}
```

Selenium & JAVA

- **Classe de test java**
 - **un test : s'authentifier**

```
public void testSignOn() throws Exception{  
    browser.click("identifier=index_signon_link_id");  
    browser.waitForPageToLoad(WAIT_TIME);  
    assertEquals(myResources.  
        getString("signon.title"), browser.getText("//title"));  
    browser.type("//input[@title='custIdsignIn']", "test1");  
    browser.type("//input[@title='custIdPassword']", "test1");  
    browser.click("signon_existing_button_signin_id");  
    browser.waitForPageToLoad("30000");  
    assertEquals(myResources.  
        getString("viewcustomer.title"), browser.getText("//title"));  
}
```

Selenium & JAVA

▪ Actions

- Exemple : à droite langage Selenese ; à gauche appel en java à partir de l'instance de navigateur récupérée
 - **open --> browser.open(String)**
 - **type --> browser.type(String, String)**
 - **select --> browser.select(String, String)**
 - **check --> browser.check(String)**
 - **click --> browser.click(String)**
 - **submit --> browser.submit(String)**

Selenium & JAVA

▪ Accessors

- A droite langage Selenese ; à gauche appel en java à partir de l'instance de navigateur récupérée

- **store<Xxx> --> get<Xxx>**

- **Exemple**

- **storeAlert** --> **browser.getAlert()**
- **storeBodyText** --> **browser.getBodyText()**
- **storeEval** --> **browser.getEval(String)**
- **storeText** --> **browser.getText(String)**
- **storeTitle** --> **browser.getTitle();**

Selenium & JAVA

▪ Assertions

- Utilisation des assertions du langage java (JUnit)
 - ~~verify<Xxx> et verifyNot<Xxx>(locator, pattern)~~
 - ~~assert<Xxx> et assertNot<Xxx>(locator, pattern)~~
- ~~waitForNot<Xxx>~~
- ~~waitFor<Xxx> -->~~
 - browser.waitForCondition
 - browser.waitForFrameToLoad
 - browser.waitForPageToLoad
 - browser.waitForPopUp

Selenium & JAVA

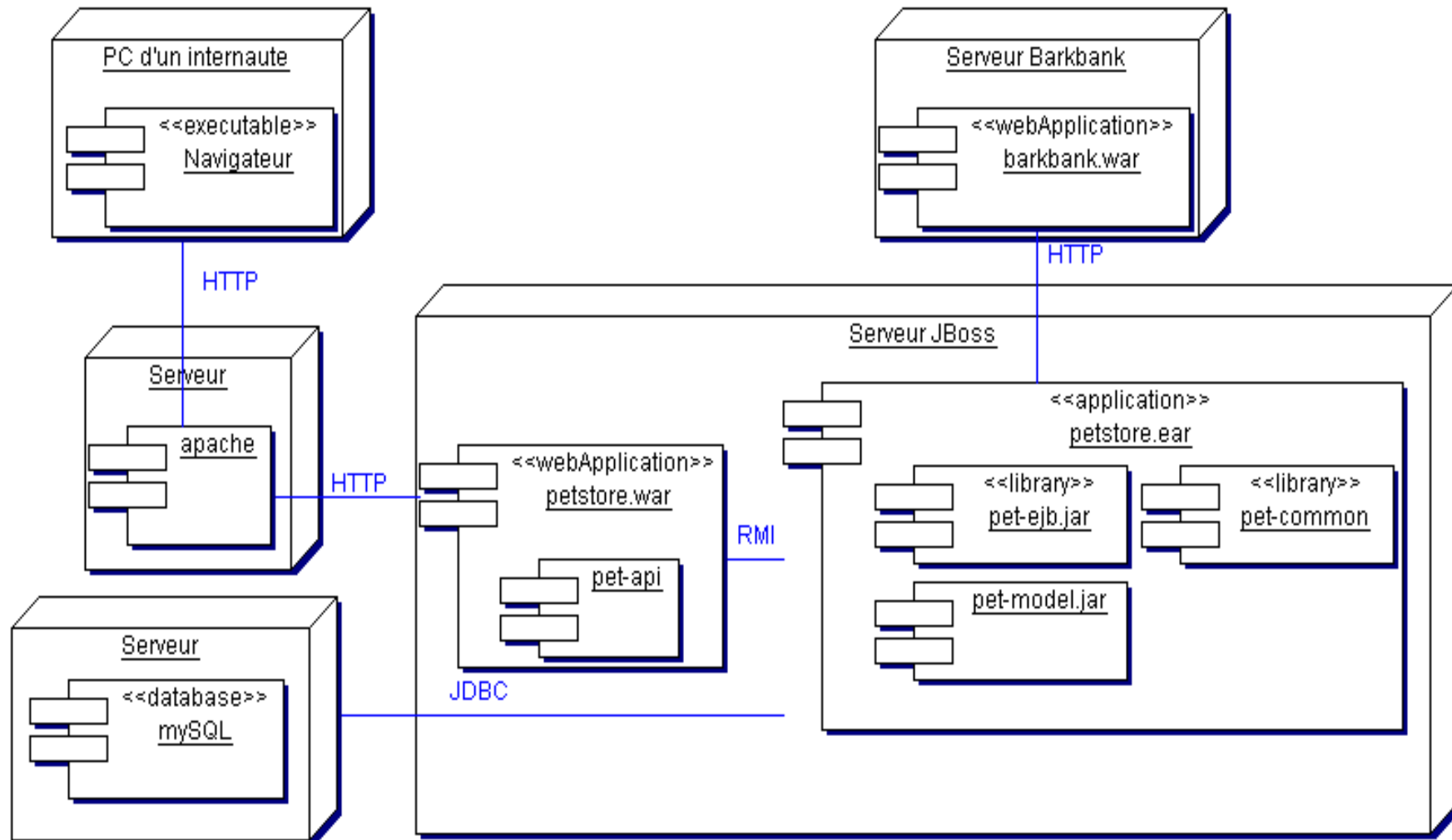


Diagramme de déploiement (démonstration)



Démonstration

www.parisjug.org



Résumé / Conclusion

Tableau récapitulatif

	Selenium core	Selenium IDE	Selenium RC
Navigateur	Tous	Firefox	Plusieurs
Installation serveur	Oui	Non	Non
Integration continue	Non	Non	Oui
Tests de performance	Non	Non	Non
Https	Oui	Oui	Oui
Multiples domaines	Non	Oui	Oui

Résumé / Conclusion

- **Selenium se décline en plusieurs composants**
 - **core**
 - **IDE**
 - **Remote control**
 - **Dernier né : Selenium grid (non traité)**
- **Langage de script simple : selenese**
- **Fournit une api pour réaliser des tests fonctionnels en java : intégration continue**

Bibliographie / liens

- **Selenium**

- <http://www.openqa.org/>
- <http://selenium-core.openqa.org/>
- <http://selenium-ide.openqa.org/>
- <http://selenium-rc.openqa.org/>
- <http://selenium-grid.openqa.org/>
- <http://wiki.openqa.org/>

- **Revue de « web testing tools »**

- <http://java-source.net/open-source/web-testing-tools>

- **Revue de « stress testing tools »**

- <http://www.softwareqatest.com/qatweb1.html>

- **Cohen, F. Java Testing and Design, From Unit Testing to Automated Web Tests. Prentice Hall. 2004**

Bibliographie / liens

- **Maven**

- <http://mojo.codehaus.org/selenium-maven-plugin/>
- <http://www.i-proving.ca/space/Technologies/Maven/Maven+Recipes>
- **Massol V., Van Zyl J., Porter B., Casey J. et Sanchez C., Better builds with maven, Mergere Library Press, 2006**



**Paris
JUG**

Questions / Réponses

www.parisjug.org



Sponsors



Merci de votre attention!



www.parisjug.org

Xebia

aneo
the other solution.

OceaneConsulting
Prestation Java

