Selenium and web testing

A. Mansurov

March 10, 2015
Web testing challenges

- Many factors affecting performance and user experience
- Large and diverse audience - various usage paths, load handling
- Possible entry point to database and system services
- Technical considerations - network speed, screen resolution, ...
- Broad infrastructure - firewalls, customer experience based on geography or device
- Different browsers
Selenium

- automated software testing tool for testing web applications
- capabilities to operate across different browsers and operating systems
- set of tools
  - IDE - Firefox plugin to record actions
  - RC - Remote Control, complex tests in Java, C#, PHP, Python, Ruby, Perl
  - WebDriver - RC successor, sends commands to browser and retrieves results
  - Grid - parallel tests across different machines and browsers
Advantages

- open source
- extendable for various technologies that expose DOM
- execute scripts across different browsers
- supports mobile devices
- tests within browser, no focus needed
- parallel tests
Selenium IDE

- Firefox plugin to develop test cases
- Features
Demo
Additional IDE features

- Debugger (extra plugin)
- Assertions, verifications
- Synchronization points
- Patterns and regular expressions
- Extensions
- Execution against Firefox - or export as WebDriver or Remote Control Script
Selenium RC

- allows to write automated web application UI tests
- Java, C#, Perl, Python or PHP to create complex tests
- Architecture
Selenium RC

- Two parts
- Selenium Server - launch & kill browsers, interpret & execute commands, HTTP proxy between browser and application under test
- Client library - interface between a programming language and Server
Three types:

- **Action** - manipulate state of application - click, select, waitForPageToLoad ...

- **Accessors** - evaluate state and store results in variable for assertions - storeAllFields, storeSelectedValue, ...

- **Assertions** - verify state of application - waitForSelected, verifyAllLinks
Locators

- identify HTML element the command refers to
- identified using FirePath and FireBug
- id=id
- name=name
- dom=javascriptExpression
- xpath=xpathExpression
- link=textPattern
- css=cssSelectorSyntax
Webdriver

- aka Selenium 2.0
- tool for automated web-application testing
- different underlying framework
- interacts directly with browser
- multi-browser testing
- handling multiple frames, windows, popups and alerts
- complex page navigation
- advanced user navigation (drag-and-drop)
- AJAX based UI elements
Architecture

Web Application

Selenium Web Driver

Selenium Test (Java, C#, Ruby, Python, Perl, PHP, Java Script)
WebDriver demo

```java
package seldemo;

public class Rdemo {
    public static void main(String[] args) throws InterruptedException {
        // Instantiate the RC Server
        selenium.start(); // Start
        selenium.open("/html/div[3]/a"); // Open the URL
        selenium.windowMaximized();

        // Click on Link Math Calculator
        selenium.click("//*[@id='menu']/div[3]/a");
        Thread.sleep(2000); // Wait for page load
        selenium.click("//*[@id='menu']/div[4]/div[3]/a");
        Thread.sleep(4000); // Wait for page load

        // Focus on Text Box
        selenium.focus("name=cp1");
        // enter a value in Text box 1
        selenium.type("css=input[name='cp1']", "584736");
        // enter a value in Text box 2
        selenium.focus("name=cp2");
        selenium.type("css=input[name='cp2']", "987654321");

        // Click Calculate button
        selenium.click("http://www.calculator.net/" + "table/tr/td[2]/input");

        // verify if the result is 5
        String result = selenium.getText("http://www.calculator.net/" + "p/2"]);
        if (result == "5") {
            System.out.println("Pass");
        }
    }
}
```
Grid

- tool to run parallel tests across different machines and different browsers simultaneously
- distributes tests across multiple physical or virtual machines
- accelerates testing process
- execute multiple instances of WebDriver or RC tests in parallel
- code need not to be present on the system they execute
- Hub: central point where tests would be triggered
- Node: Selenium instance attached to the Hub which executes the tests
Grid architecture
The End.