

Selenium

A suite of tools in Web testing

- Qijie Xue

What is Selenium

- Selenium is a suite of tools for Web Automation testing
- It Supports:
 - Many browsers[IE, Chrome, Firefox, Safari...]
 - Many platforms[Windows, Linux, Mac]
 - Many Pro Languages[Java, C#, Python, ruby...]
 - Distributed test environment and test management[Selenium Grid or Selenium RC]
 - Record and play[Firefox plugin, called Selenium IDE]

When to use Selenium

- Regression testing that occurs frequently
- Verification testing after bug fix
- Automate the test cases that run time and time again

History of Selenium

2004

Jason Huggins in ThoughtWorks

- Developed a JS class library to drive the browser, it's called **Selenium Core**
- Selenium IDE and Selenium Remote Control

2006

Simon Stewart in Google

- Started a new project, called WebDriver, it uses Browser native API to driver the browser. (implementation vary for different browsers)

2008

Selenium 2.0

- Selenium married with WebDriver, Selenium-WebDriver becomes a key component of Selenium

Selenium Members

- Selenium Core
- Selenium IDE
- Selenium RC(Remote Control)
- Selenium WebDriver
- Selenium Grid

Selenium 2

WebDriver, the future of selenium



C# Web Driver installation

- Run the following command in the Package manger Console in Visual Studio
 - PM> Install-Package Selenium.WebDriver
- Name space
 - using OpenQA.Selenium
- Download driver server for IE, Chrome, and put it in the environment variables directory (system32)
 - IEDriverServer.exe
 - chromedriver.exe

Supported Web Drivers

- `OpenQA.Selenium.IE`
- `OpenQA.Selenium.Chrome`
- `OpenQA.Selenium.Firefox`
- `OpenQA.Selenium.Safari`
- `OpenQA.Selenium.PhantomJS`

Initialize Web Driver

```
1. QA.IWebDriver theDriver = null;
2. switch (this.browser)
3. {
4.     case Browsers.IE:
5.     {
6.         QA.IE.InternetExplorerOptions _ieOptions = new QA.IE.InternetExplorerOptions();
7.         _ieOptions.IntroduceInstabilityByIgnoringProtectedModeSettings = true;
8.         theDriver = new QA.IE.InternetExplorerDriver(_ieOptions);
9.     }; break;
10.    case Browsers.Chrome:
11.    {
12.        theDriver = new QA.Chrome.ChromeDriver();
13.    }; break;
14.        case Browsers.Firefox:
15.        {
16.            theDriver = new QA.Firefox.FirefoxDriver();
17.        }; break;
18.    case Browsers.Safari:
19.    {
20.        theDriver = new QA.Safari.SafariDriver();
21.    };break;
22.    case Browsers.PhantomJS:
23.    {
24.        theDriver = new QA.PhantomJS.PhantomJSDriver();
25.    };break;
26.    default:
27.    {
28.        QA.IE.InternetExplorerOptions _ieOptions = new QA.IE.InternetExplorerOptions();
29.        _ieOptions.IntroduceInstabilityByIgnoringProtectedModeSettings = true;
30.        theDriver = new QA.IE.InternetExplorerDriver(_ieOptions);
31.    }; break;
32.    }
33.    return theDriver;
```

Web Element Locator

- `OpenQA.Selenium.IWebDriver wd = null;`
 - `wd.FindElement(OpenQA.Selenium.By)`
 - `wd.FindElements(OpenQA.Selenium.By)`
- `OpenQA.Selenium.IWebElement we= null;`
 - `we.FindElement(OpenQA.Selenium.By)`
 - `we.FindElements(OpenQA.Selenium.By)`

Element Selector

```
.FindElements(By.XPath("@//a/h2"));
```

- ClassName
- CssSelector
- Equals
- Id
- LinkText
- Name
- PartialLinkText
- ReferenceEquals
- TagName

Manipulate Element

- `OpenQA.Selenium.IWebElement we = null;`
 - `we.click()`
 - `we.SendKeys(string keys)`
- Use `OpenQA.Selenium.Interactions`
 - `Actions()`
 - `TouchActions()`

Interactions.Actions

```
(new OpenQA.Selenium.Interactions.Actions(WebDriver)).SendKeys(WebElement, "SharePoint").Perform();
```

- Click
- ClickAndHold
- ContextClick
- DoubleClick
- DragAndDrop
- DragAndDropToOffset
- Equals
- GetHashCode
- GetType

Switch to Window

- First, get all browser windows of the driver
- For each window, check if it's the expected

```
1.  string theCurrent = wd.CurrentWindowHandle;
2.  IList<string> windows = wd.WindowHandles;
3.  foreach (var window in windows)
4.  {
5.      wd.SwitchTo().Window(window);
6.      if (wd.Title.ToLower() == title.ToLower())
7.      {
8.          return;
9.      }
10. }
11. wd.SwitchTo().Window(theCurrent);
```

Switch to Frame

- First, get all frames by tag name
- For each frame, check if it's the expected

```
1.  OpenQA.Selenium.IWebElement theFrame = null;
2.  var frames =
    wd.FindElements(OpenQA.Selenium.By.TagName("iframe"));
3.  foreach (var frame in frames)
4.  {
5.      if (frame.GetAttribute("name").ToLower() == name.ToLower())
6.      {
7.          theFrame = (OpenQA.Selenium.IWebElement)frame;
8.          break;
9.      }
10. }
11. if (theFrame != null)
12. {
13.     wd.SwitchTo().Frame(theFrame);
14. }
```

Advanced (Waits)

- Explicit Waits (has a timeout value, return immediately after expected condition is true)
 - PM> Install-Package Selenium.Support
 - OpenQA.Selenium.Support.UI.WebDriverWait
- Implicit Waits (this is global, it works throughout the life of web driver instance.)

```
WebDriverWait _webDriverWait = new UI.WebDriverWait(_IEDriver, TimeSpan.FromSeconds(10));  
_webDriverWait.Until((d) => { return d.Title.ToLower().StartsWith("微软虚拟学院"); });
```

```
WebDriver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(10));
```


Scroll use JS

- For entire html tag

1. `var js = "document.documentElement.scrollTop=10000";`
2. `_IEDriver.ExecuteScript(js, null);`

1. `var js = "document.documentElement.scrollLeft=10000";`
2. `_IEDriver.ExecuteScript(js, null);`

- For an element

1. `string id = "";`
2. `var js = "var q = document.getElementById('" + id + "`
 `').scrollTop=10000";`
3. `WebDriver .ExecuteScript(js, null);`

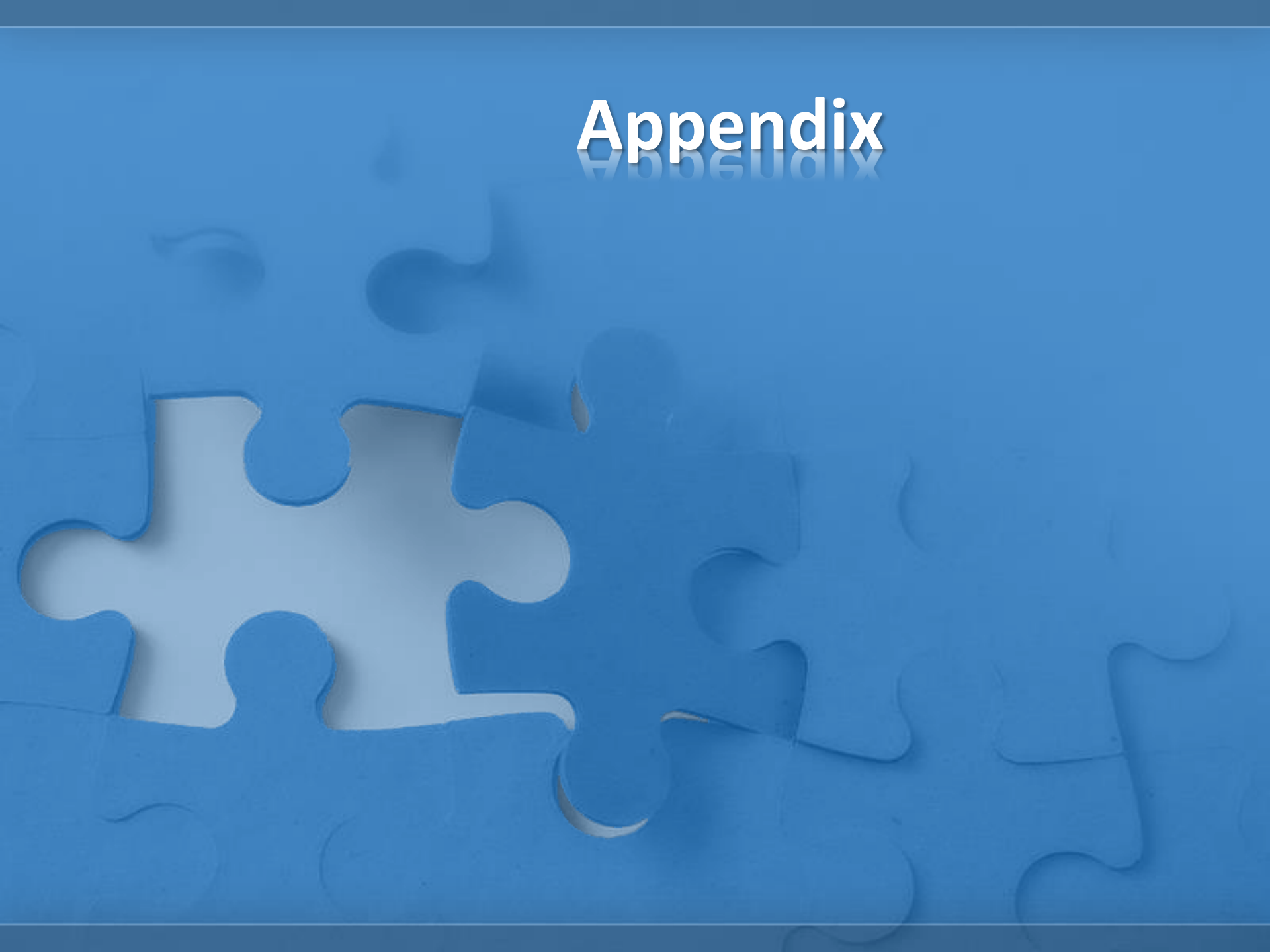
Screen shot

- OpenQA.Selenium.Screenshot
- Chrome.ChromeDriver _chrDriver =
Chrome.ChromeDriver();
 - 1.Screenshot _screenshot = _chrDriver .GetScreenshot();
 - 2._screenshot.SaveAsFile("filepath",System.Drawing.Imaging.ImageFormat.Jpeg);

Clean Up

- `WebDriver.Close()` - Close the current browser window
- `WebDriver.Quit()` - Close all browser windows that associated to the driver

Appendix



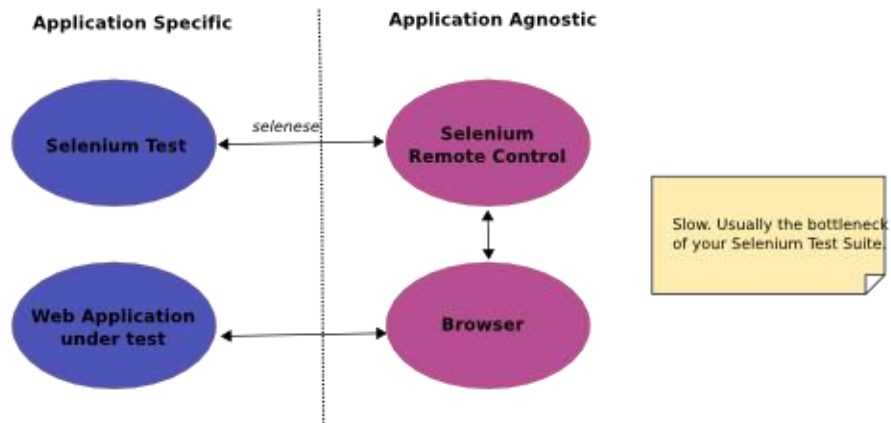
XPath Syntax

路径表达式	结果
bookstore	选取 bookstore 元素的所有子节点
/bookstore	选取根元素 bookstore 注释：假如路径起始于正斜杠(/)，则此路径始终代表到某元素的绝对路径！
bookstore/book	选取所有属于 bookstore 的子元素的 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择所有属于 bookstore 元素的后代的 book 元素，而不管它们位于 bookstore 之下的什么位置。
//@lang	选取所有名为 lang 的属性

CSS Selector

- Tag selector : E **Sample**: td, a
- ID selector : #id **Sample**: #note
- Class selector :E.className **Sample**: div.note
- Descendant selector : E F **Sample**: #link a
- Groups of selector :E1,E2,E3 **Sample**: td,p,div{}

Traditional Selenium Setup



Resources

- SeleniumHQ
- <http://www.seleniumhq.org/download/>
- WebDriver
- <http://blog.csdn.net/smilings/article/details/7395509>
- C# API
- <http://selenium.googlecode.com/git/docs/api/dotnet/index.html>