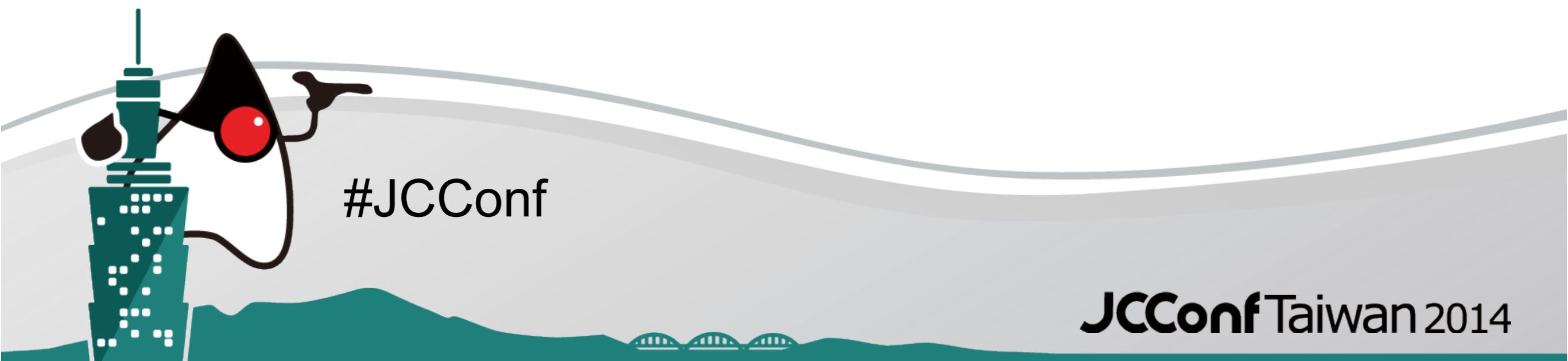


淺談 Geb 網站自動化測試

林彥宏 lyhcode@gmail.com



Who Am I



林彥宏

lyhcode.info

- 遊牧工程師 / Nomadic Developer
- 遠距工作者 / Remote Worker
- 喜歡分享程式設計的樂趣
 - Founder of 思創軟體，校園協同教學業師
 - Co-founder of PLWeb，開發程式語言學習網
- Groovy user since 2008

Geb (*pronounced “jeb”*)

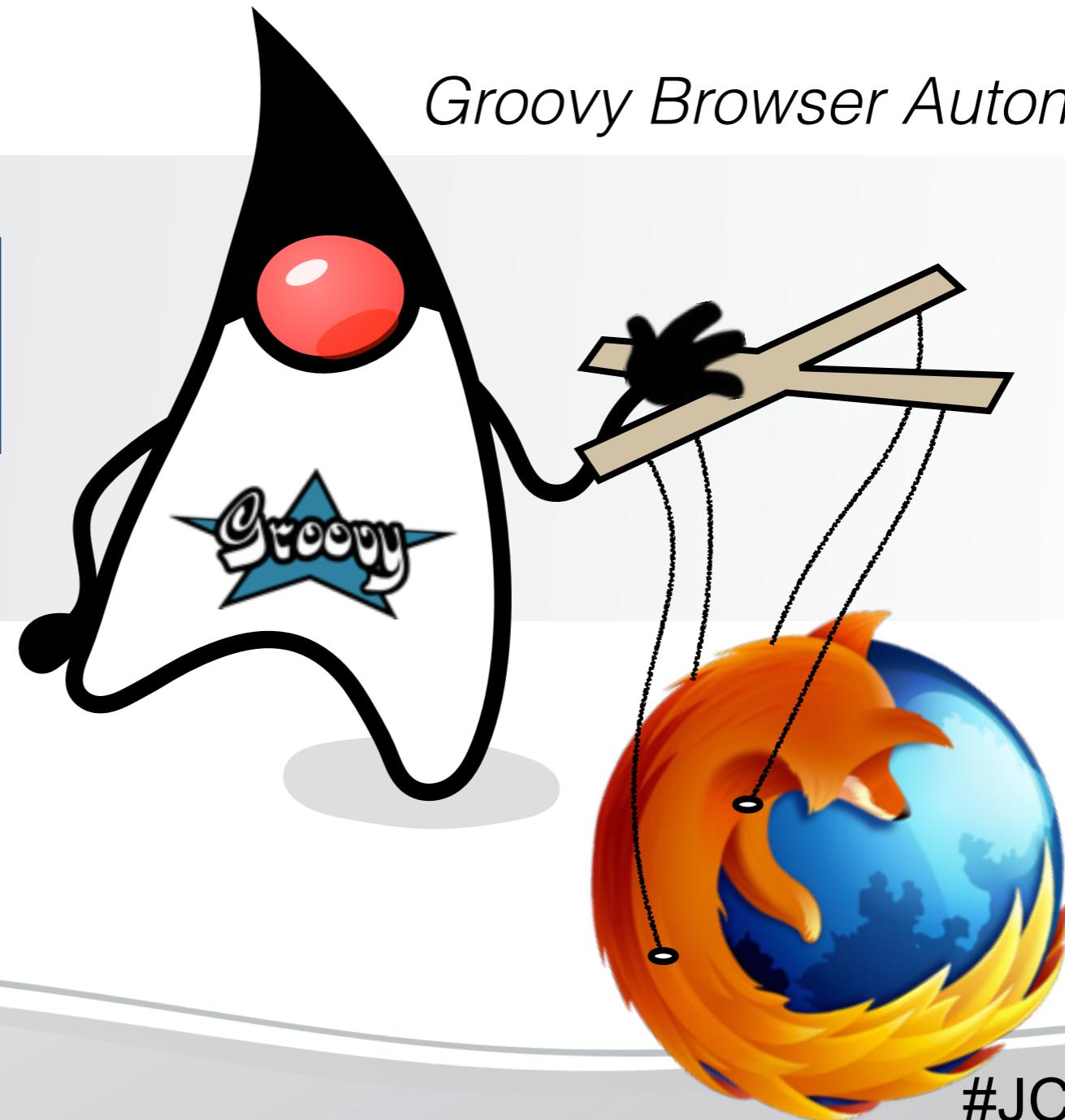
very groovy browser automation... web testing, screen scraping and more

What is Geb?

Groovy Browser Automation

瀏覽器自動化

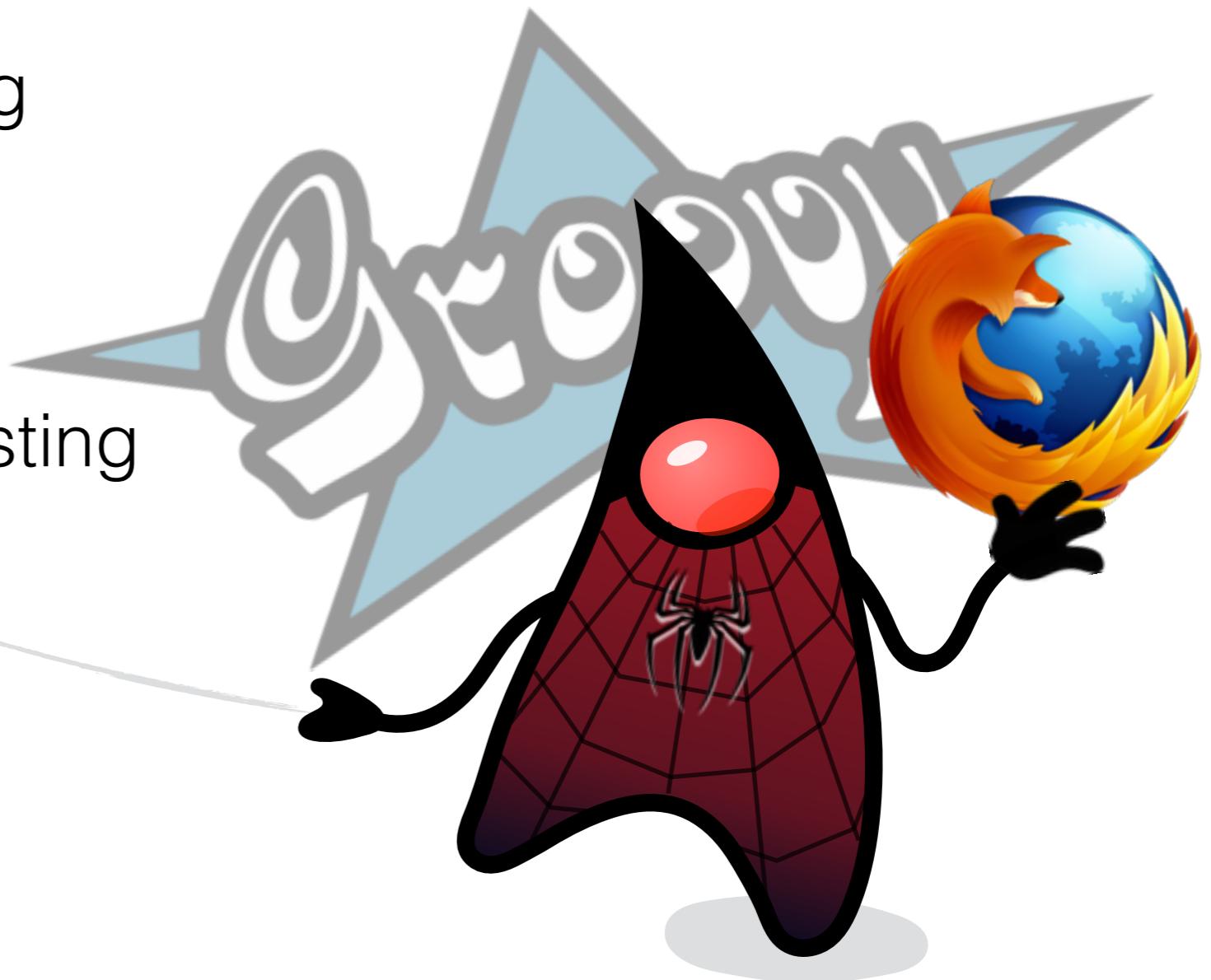
Browser Automation



Geb
(pronounced "jeb")

Geb 可以做什麼？

- Web Application Testing
 - Functional Testing
 - User Acceptance Testing
- Web Automation Script
- Web Crawler
- Webpage Screenshot



Geb

(pronounced “jeb”)

Geb 的優點

- Based on the **WebDriver**
- *jQuery-like selector and syntax*
- 易學易讀易寫的 Groovy DSL
- 實現 Page Object Pattern
- 高度整合測試框架，如 *JUnit, Spock, TestNG...*
- 開發流程整合，搭配 *Grails, Gradle, Maven, Jenkins CI...*



Geb *(pronounced “jeb”)*

very groovy browser automation... web testing, screen scraping and more

Geb

Groovy

WebDriver

Java SE

Geb
(pronounced "jeb")

SeleniumHQ

Browser Automation Suite



Selenium IDE



Selenium Remote Control



Selenium WebDriver



Geb 的核心驅動



Selenium Grid

Selenium Sponsors:



Geb

(pronounced “jeb”)

Selenium Core

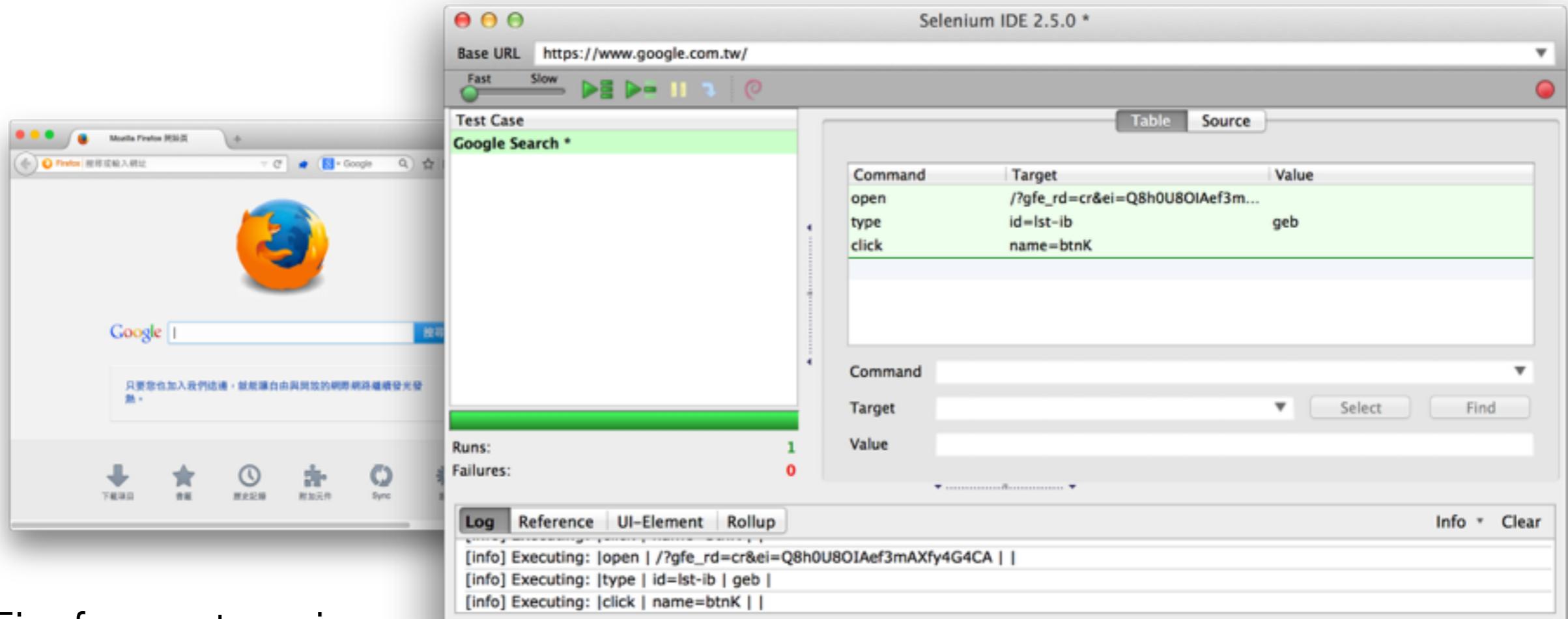


- Created by Jason Huggins in 2004 at ThoughtWorks.
- 源起自一個測試工具 “[JavaScriptTestRunner](#)”

Geb

(pronounced “jeb”)

Selenium IDE



- Firefox extension
- Easy record and playback

Geb

(pronounced “jeb”)

WebDriver

- Selenium 1.0 + WebDriver = Selenium 2.0
- WebDriver API 規範已提交 W3C 標準
 - <http://www.w3.org/TR/webdriver/>
- Selenium WebDriver
 - 支援多種程式語言：
 - Java, C#, Ruby, Python, Perl
 - 支援多種瀏覽器：
 - Google Chrome、Firefox 與 Internet Explorer 等
 - HtmlUnit (headless browser)

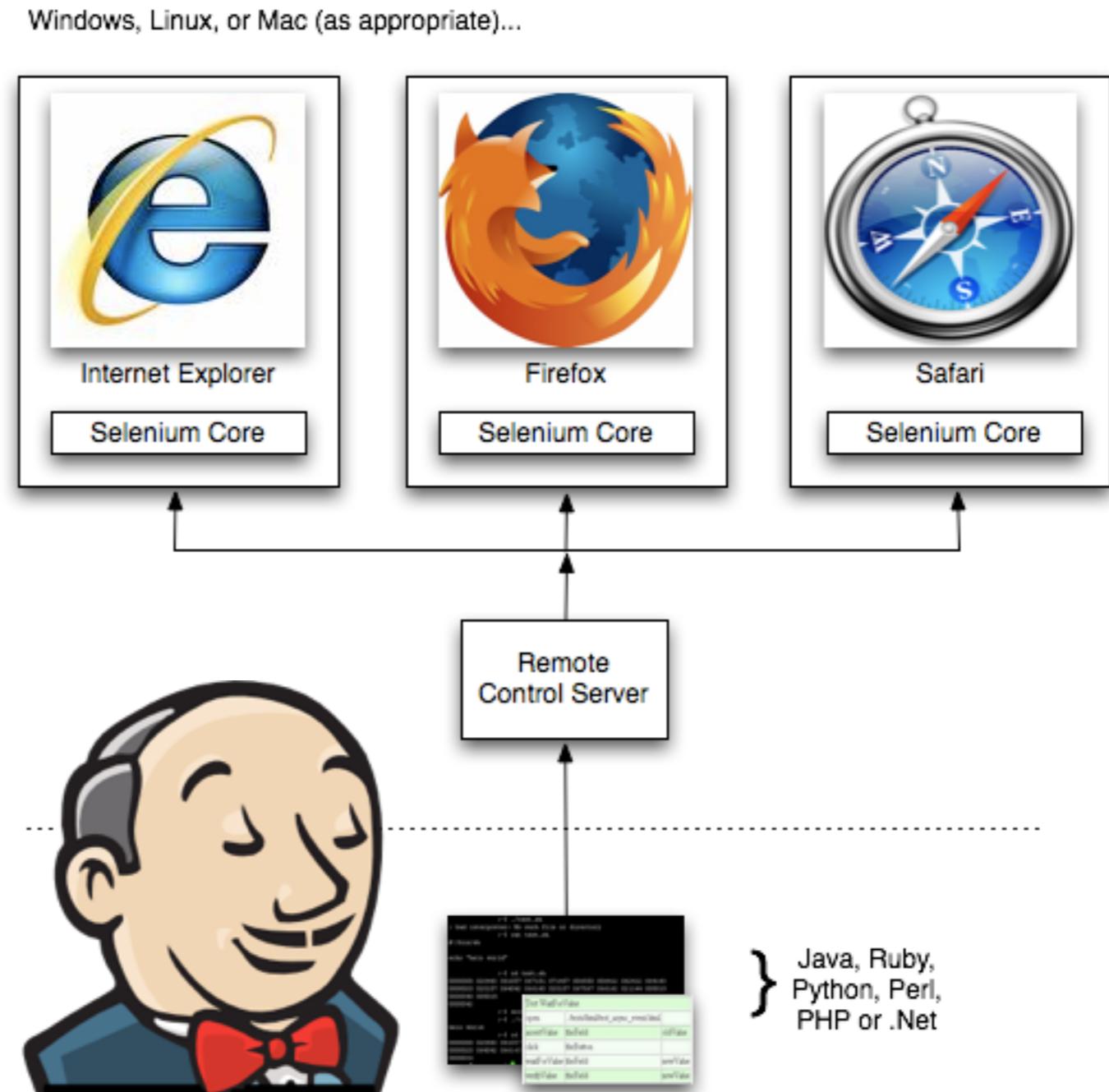


Geb

(pronounced "jeb")

Selenium Remote Control

- Selenium RC
- Client / Server 架構
- Great for testing under a Continuous Integration system (e.g. Jenkins CI)



自動化測試的難題

為什麼我們還需要 Geb ?

```
987. WebDriver driver = new FirefoxDriver();  
988. driver.get("http://www.google.com");  
989. WebElement element = driver.findElement(By.name("q"));  
990. element.sendKeys("geb");  
991. element.submit();  
992. System.out.println("Title: " + driver.getTitle());  
993. driver.quit();
```



Geb *(pronounced “jeb”)*

very groovy browser automation... web testing, screen scraping and more

Geb

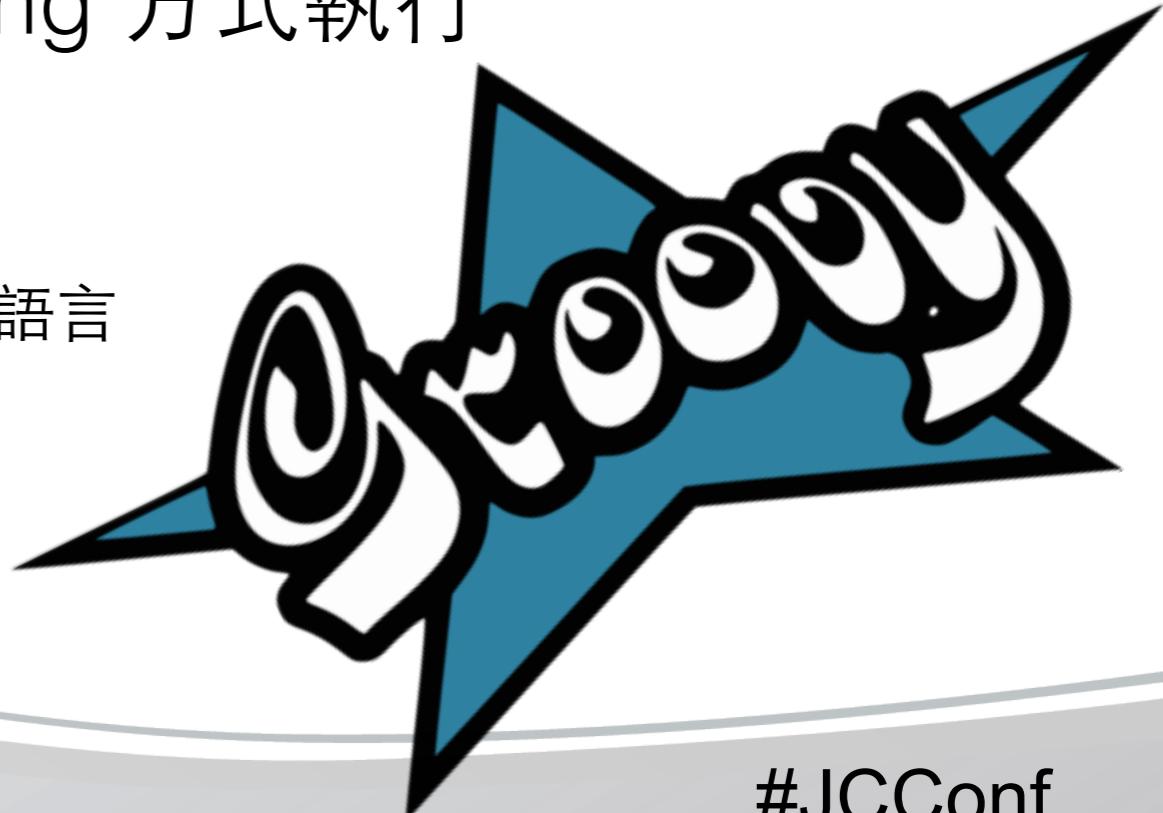
Groovy

WebDriver

Java SE

Why Groovy

- Groovy makes Java better, more powerful and easier to use.
- Groovy 易學易寫，從 Java 到 Groovy 門檻極低
- 程式碼更精簡，支援以 Scripting 方式執行
- Groovy 發揮 DSL 優勢
DSL, Domain-Specific Language 領域描述語言
- 共享 Java 豐富資源



Geb

(pronounced “jeb”)

Groovy DSL

```
1. task hello {  
2.     doLast {  
3.         println "Hello"  
4.     }  
5. }  
6.
```

*Example: Define a **Gradle** Task*



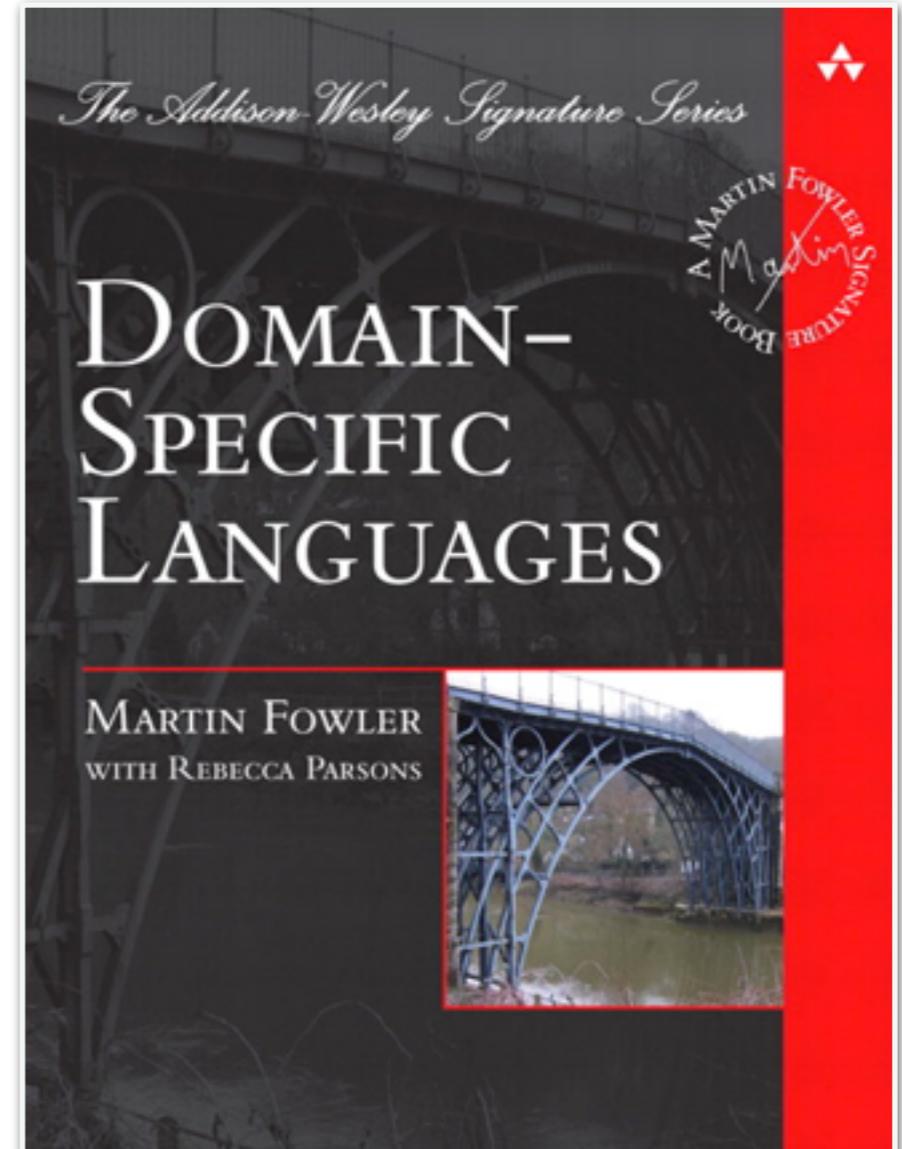
Geb Why Are DSLs Important?

(pronounced “jeb”)

- To improve programmer's productivity.
- Smaller and easier to understand, allow **nonprogrammer** to see the code that drives important parts of their business.

— Martin Fowler

ThoughtWorks



Geb *(pronounced “jeb”)*

very groovy browser automation... web testing, screen scraping and more

Geb

Groovy

WebDriver

Java SE

Geb Awesome

- jQuery-like
 - CSS based content lookup (Navigator API)

```
$('input[name=username]').value()
```

- Method Chaining

```
$('h3').children().attr('href')
```

- DSL featured

```
Browser.drive {  
    go 'http://google.com'  
}
```

This is Geb code, not jQuery :)

開始使用 Geb

1. 安裝 JDK
2. 安裝 Groovy
3. 使用 Groovy Shell 或 Groovy Console

安裝 Groovy

- 建議使用 GVM 工具 (for Mac OS X, Linux)
 - gvmtool.net
 - Install Groovy with GVM tool

```
gvm install groovy
```

安裝 Geb ?

- Geb 是一個 Java/Groovy Library ,
not standalone software
- 主要套件名稱是 geb-core
- 可以從 Maven Repository 取得所需 JAR 檔案

Geb

(pronounced “jeb”)

Geb with Groovy Shell



The screenshot shows a terminal window titled "learning-geb – firefox-bin – 100x25". The window contains Groovy Shell code demonstrating the use of Geb:

```
MacBook:learning-geb $ groovysh
Groovy Shell (2.3.6, JVM: 1.8.0_25)
Type ':help' or ':h' for help.

groovy:000> println 'Hello Geb'
Hello Geb
====> null
groovy:000> import groovy.grape.Grape
====> groovy.grape.Grape
groovy:000> Grape.grab(group: 'org.gebish', module: 'geb-core', version: '0.9.3')
====> null
groovy:000> Grape.grab(group: 'org.seleniumhq.selenium', module: 'selenium-firefox-driver', version:
  '2.43.1')
====> null
groovy:000> browser = new geb.Browser()
====> geb.Browser@75390459
groovy:000> browser.go 'http://google.com'
====> null
groovy:000>
```

A large dollar sign (\$) followed by "groovysh" is displayed at the bottom left of the terminal window.

REPL (Read–Eval–Print Loop) 適用於 Geb step-by-step 的執行方式

Dependencies of Geb

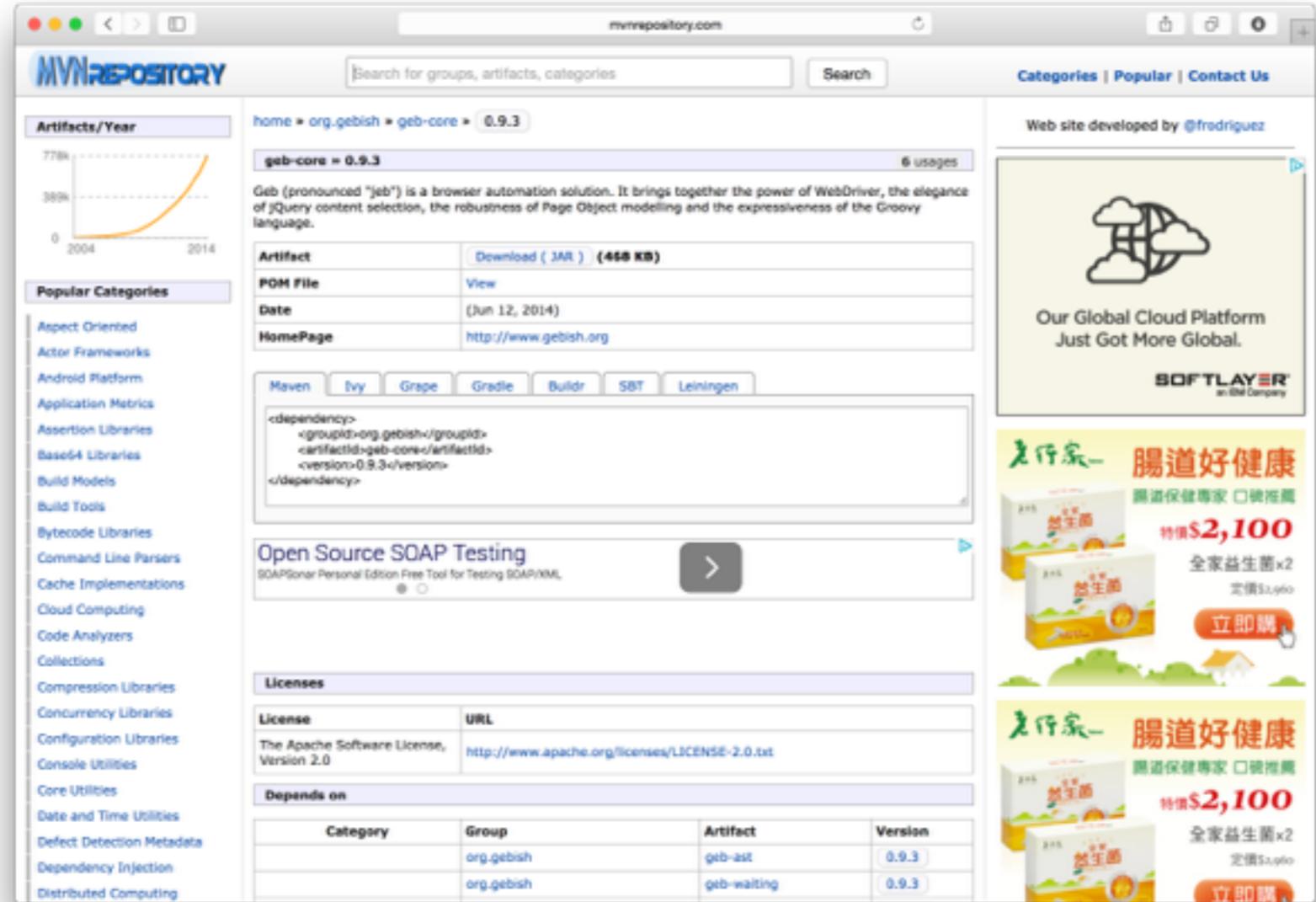
Geb 需要 2 個套件 (artifact)

```
<dependency>
    <groupId>org.gebish</groupId>
    <artifactId> geb-core </artifactId>
    <version>0.10.0</version>
</dependency>
```

```
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId> selenium-firefox-driver </artifactId>
    <version>2.44.0</version>
</dependency>
```

Geb
(pronounced “jeb”)

搜尋 Maven Repository



mvnrepository.com

Geb

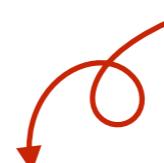
(pronounced “jeb”)

Using Grapes

1. groovy.grape.Grape.grab(group: 'org.gebish', module: 'geb-core', version: '0.10.0')

2. groovy.grape.Grape.grab(group: 'org.seleniumhq.selenium', module: 'selenium-firefox-driver', version: '2.44.0')

Attention: Not 'org.codehaus.geb'



Grape lets you quickly add maven repository dependencies to your classpath.

Geb
(pronounced "jeb")

Browse Google Web Site

- 建立 *Browser* 物件
3. `b = new geb.Browser()`
- 命令瀏覽器開啟 URL 位址顯示網頁
4. `b.go 'http://google.com'`
5. `b.title`
- 取得網頁標題“Google”
- 

Geb

(pronounced “jeb”)

檢測元素



利用「檢測器」觀察即時的 *DOM* 內容；而非 *HTML* 原始碼

```

<div id="sb_ifc0" class="sbib_b" dir="ltr">
  <div id="gs_lc0" style="position: relative;">
    <input id="lst-ib" class="gsfi lst-d-f" type="text"
      value="" title="搜尋" autocomplete="off" name="q"
      maxlength="2048" aria-haspopup="false"
      role="combobox" aria-autocomplete="both"
      style="border: medium none; padding: 0px; margin:
      0px; height: auto; width: 100%; position: absolute; z-index: 6; left:
      0px; outline: medium none; direction: ltr;
      spellcheck="false"></input>
    <div id="gs_sc0" class="gsfi" style="background:
      none repeat scroll 0% 0% transparent; color: tra...
  
```

DOM 選擇器(Selector)

```
<form class="tsf" action="/search"  
      id="tsf" method="GET"  
      name="f" role="search">
```

- A. \$('form.tsf')
- B. \$('form#tsf')
- C. \$('form[name=f]')

選哪一個好呢？

Geb Navigator API

```
<form class="tsf" action="/search"  
id"tsf" method"GET"  
name"f" role"search">
```

```
$( 'form.tsf' ).attr('action')
```

```
$( 'form[name=f]' ).attr('action')
```

```
$( 'form' , name: 'f' ).attr('action')
```

```
$( 'form' , id: 'tsf' ).attr('action')
```

表單存取

6. `form = b.$('form#tsf')`

7. `form.q = 'jcconf 2014'`

修改表單欄位資料；立刻
在瀏覽器畫面產生改變

利用選擇器找到目標表單元素



Geb
(pronounced "jeb")

表單送出

8. `form.btnk().click()`

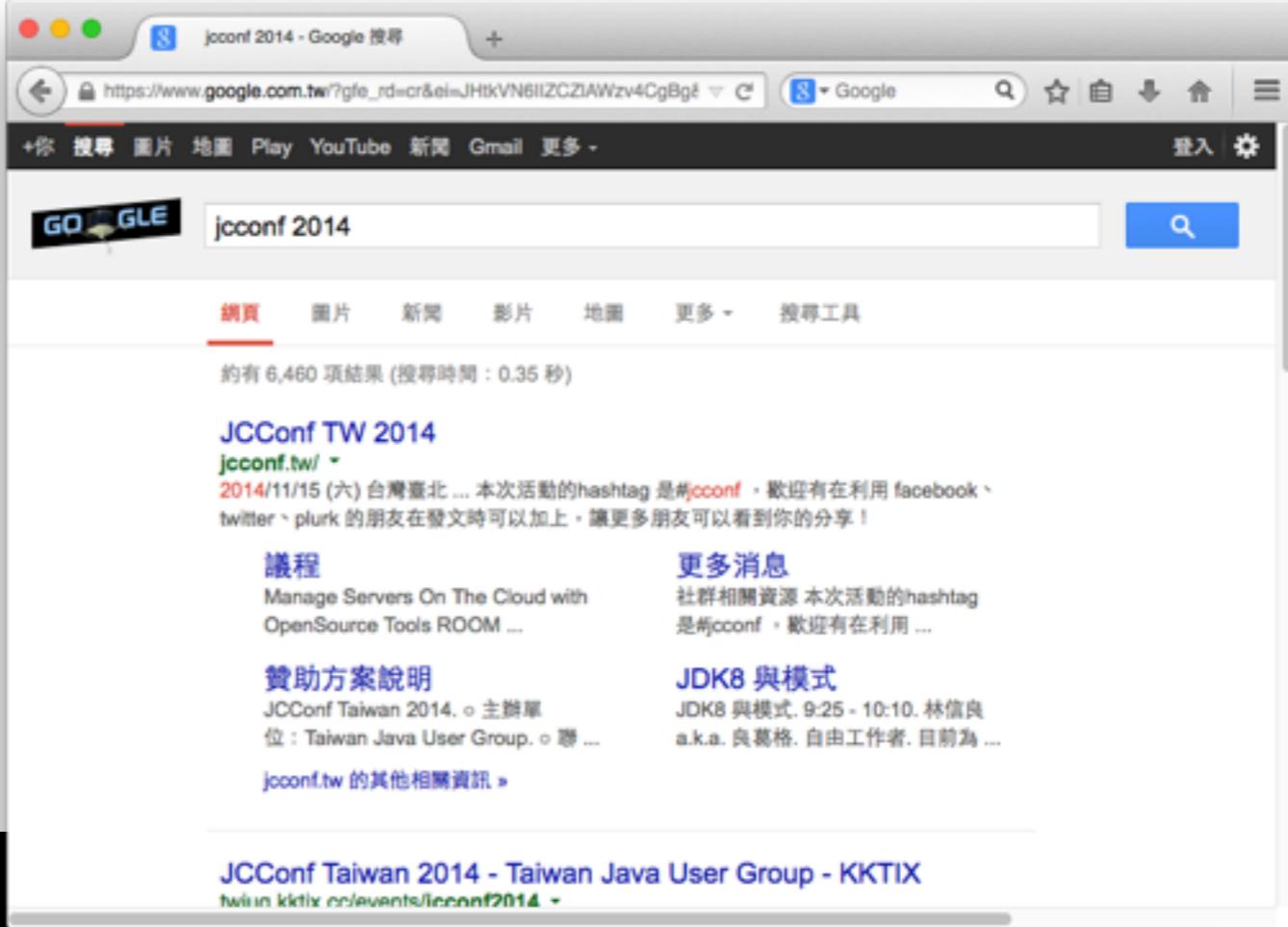


Geb
(pronounced “jeb”)

瀏覽器即時互動

瀏覽器畫面更新後，*Geb* 程式
可以獲取新的資料

```
groovy:000> b.title
===> jcconf 2014 – Google 搜尋
groovy:000> b.currentUrl
===> https://www.google.com.tw/?gfe_rd=cr&ei=JHtkVN6IIIZCZlAWzv4CgBg&gws_rd=ssl#q=jcconf+2014
```



JCConf Taiwan 2014 - Taiwan Java User Group - KKTIX
tw@in.kktix.cc/events/jcconf2014 -

存取網頁內容

搜尋結果的所有標題文字

```
9. b.$('h3').collect { it.text() }
```

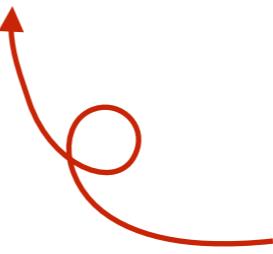
```
10. b.$('h3 a').collect { it.attr('href') }
```

搜尋結果中所有的網頁連結

Geb
(pronounced "jeb")

點擊連結 Click

```
11. b.$('h3 a').first().click()
```



點進去搜尋排名在第一筆的網站連結

Scripting with Groovy

GoogleSearch.groovy

```
1. @Grab('org.gebish:geb-core:0.10.0')
2. @Grab('org.seleniumhq.selenium:selenium-firefox-driver:2.44.0')
3. 
4. b = new geb.Browser()
5. b.go 'http://google.com'
6. 
7. form = b.$('form#tsf')
8. form.q = 'jcconf 2014'
9. form.btnK().click()
10. 
11. b.quit()
```

使用 *@Grab* 引入需要的套件

```
$ groovy GoogleSearch.groovy
```

Geb
(pronounced “jeb”)

Re-write in DSL Style

```
1. @Grab('org.gebish:geb-core:0.10.0')
2. @Grab('org.seleniumhq.selenium:selenium-firefox-driver:2.44.0')
3.
4. Browser.drive {
5.     go 'http://google.com'
6.
7.     $('form#tsf').with {
8.         q = 'jcconf 2014'
9.         btnK().click()
10.    }
11. }.quit()
```

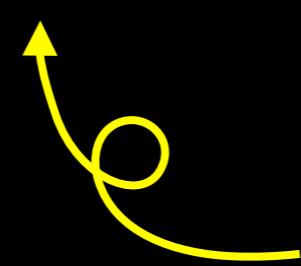
DSL 語法使用 Groovy Closure 實現

Geb

(pronounced “jeb”)

waitFor

```
1. Browser.drive {  
2.     go 'http://google.com'  
3.  
4.     $('form#tsf').with {  
5.         q = 'jcconf 2014'  
6.         btnK().click()  
7.     }  
8.  
9.     waitFor {  
10.         title.contains('jcconf 2014')  
11.     }  
12.  
13.     $('h3').each {  
14.         println it.text()  
15.     }  
16. }.quit()
```



一直等…一直等…一直等…
直到標題變成 “jcconf 2014”
或是 *Timeout* (逾時)

預設的 Wait Timeout 預設值為 5 秒

```
1. #!/usr/bin/env groovy
2. @Grab('org.gebish:geb-core:0.10.0')
3. @Grab('org.seleniumhq.selenium:selenium-firefox-driver:2.44.0')
4. import geb.Browser
5.
6. def keywords = args.join(' ')
7.
8. Browser.drive {
9.     go 'http://google.com'
10.
11.    $('form#tsf').with {
12.        q = keywords
13.        btnK().click()
14.    }
15.    waitFor {
16.        $('h3').size() > 0
17.    }
18.    $('h3').each {
19.        println "* ${it.text()}"
20.    }
21. }.quit()
```

GoogleSearch

command-line tool

```
$ ./GoogleSearch jcconf 2014 | sort | less
```

Geb
(pronounced “jeb”)

Reporting

- 產生測試報告
 - 原始碼 (.html)
 - 網頁快照 (*.png)

Geb

(pronounced "jeb")

Reporting

```
1. Browser.drive {  
2.     config.reportsDir = new File('..')      設定報表路徑  
3.  
4.     go 'index.html'  
5.     report 'home page'                   report 1  
6.  
7.     go 'login.html'  
8.     report 'login page'                  report 2  
9.  
10.    $('#form').with {  
11.        username = '...'  
12.        password = '...'  
13.        btnLogin().click()  
14.    }  
15.  
16.    waitFor { title.contains('Dashboard') }  
17.    report 'dashboard page'              report 3  
18. }
```

ScreenShot

command-line tool

```
$ ./ScreenShot http://jcconf.tw
```



```
1.#!/usr/bin/env groovy  
  
2. @Grab('org.gebish:geb-core:0.10.0')  
  
3. @Grab('org.seleniumhq.selenium:selenium-firefox-driver:2.44.0')  
  
4. import geb.Browser  
  
5. Browser.drive {  
  
6.     config.reportsDir = new File('.')  
  
7.     go args[0]  
  
8.     report "screenshot"  
  
9. }.quit()
```

\$./ScreenShot http://jcconf.tw

Geb
(pronounced “jeb”)

Cross-browser Testing

- selenium-chrome-driver
- selenium-firefox-driver
- selenium-ie-driver
- selenium-safari-driver
- selenium-htmlunit-driver

HtmlUnit Driver

- HtmlUnit - Java GUI-Less browser, supporting JavaScript, to run against web pages
- 適合沒有桌面環境的 Server 執行
- 支援 JavaScript ?
- 優點：速度快；缺點：不適合複雜的網頁

設定 Geb Driver

```
1. @Grab('org.gebish:geb-core:0.10.0')  
2. @Grab('org.seleniumhq.selenium:selenium-htmlunit-driver:2.44.0')  
3. import geb.Browser  
4. driver = 'htmlunit'  
5.  
6. Browser.drive {  
7.     go ...  
8. }.quit()
```

Geb

(pronounced “jeb”)

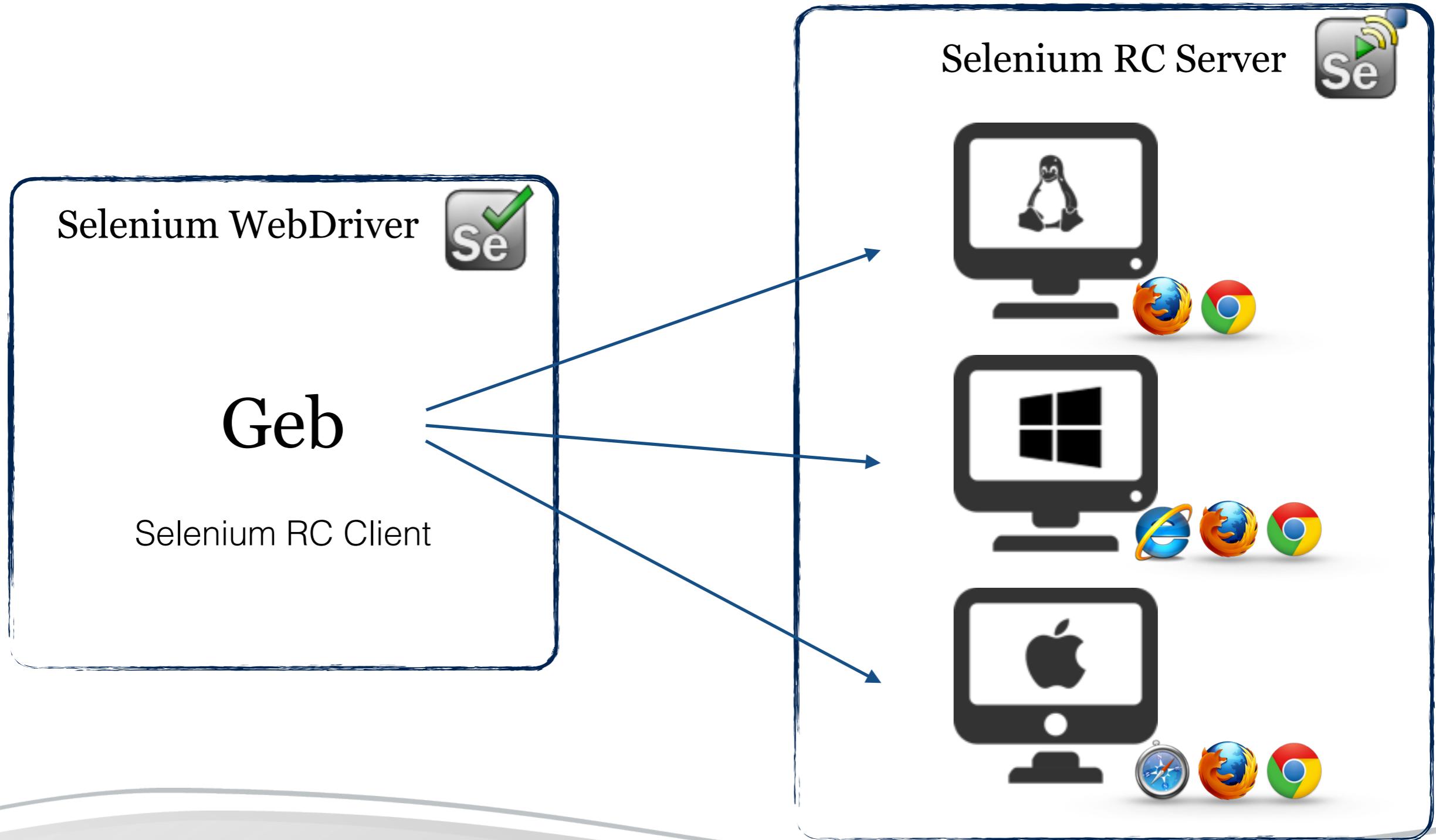
ChromeDriver

```
1. @Grapes([
2.     @Grab('org.gebish:geb-core:0.10.0'),
3.     @Grab('org.seleniumhq.selenium:selenium-chrome-driver:2.44.0')
4. ])
5. import geb.Browser
6.
7. System.setProperty('webdriver.chrome.driver', '/tmp/chromedriver')
8.
9. driver = {
10.     new org.openqa.selenium.chrome.ChromeDriver()
11. }
12.
13. Browser.drive {
14.     go ...
15. }
```



Geb
(pronounced "jeb")

RemoteWebDriver



RemoteWebDriver

- Launch a selenium server

```
java -jar selenium-server-standalone-2.44.0.jar -port 4000
```

→ <http://192.168.0.144:4120/wd/hub>

```
driver = new RemoteWebDriver(  
    new URL('http://192.168.0.144:4120/wd/hub') ,  
    DesiredCapabilities.firefox()  
)
```

Geb
(pronounced “jeb”)

Cloud Browser Testing

- VM-based browsers on-demand
- 輕鬆測試各種[瀏覽器/作業系統/版本]的組合
- <https://saucelabs.com/>
- <http://www.browserstack.com/>



Geb

(pronounced “jeb”)

BrowserStack

The screenshot shows the BrowserStack dashboard. At the top, there's a banner about new Android versions. The navigation bar includes links for LIVE, AUTOMATE, SCREENSHOTS, RESPONSIVE, Upgrade, Docs, Help, and Account.

The main area features a "Quick start" section with tabs for "Test a public URL" and "Test an internal URL". A URL input field contains "https://www.ironyun.com/". Below it, a "Favorites" section lists IE 7, XP, IE 10 Desktop, and B. The "Start testing" button is visible.

On the left, there are sections for "Local Testing" (with a "Setup Local Testing" button) and "Issue tracker" (with a "Capture a bug" button and a link to "View complete issue tracker list").

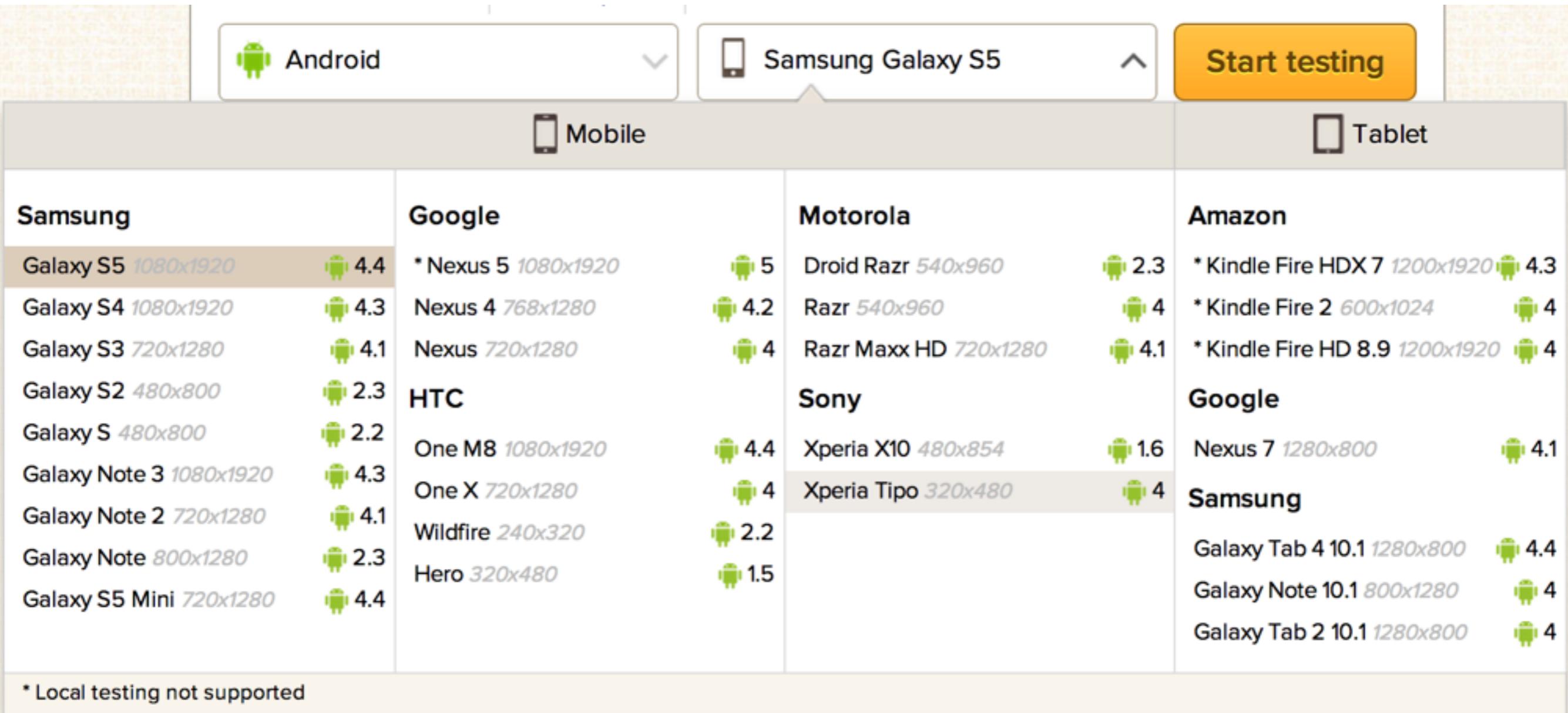
The central part of the dashboard displays a grid of browser and OS combinations. The columns are labeled: Desktop, IE, Firefox, Safari, Chrome, and Opera. The rows list various operating systems: Windows 8.1, Windows 8, Windows 7, and Windows XP. Each cell in the grid contains numerical values representing test results or metrics.

Desktop	IE	Firefox	Safari	Chrome	Opera			
Windows 8.1	10 Desktop	35 aurora	25	5.1	40 dev	30	26 dev	16
Windows 8	10	34 beta	24		39 beta	29	25	15
Windows 7		33	23		38	28	24	12.16
Windows XP		32	22		37	27	23	12.15
		31	21		36	26	22	12.14
		30	20		35	25	21	12.10
		29	19		34	24	20	12
		28	18		33	23	19	
		27	17		32	22	18	
		26	16		31		17	

Geb

(pronounced “jeb”)

BrowserStack



The screenshot shows the BrowserStack mobile testing interface. At the top, there are dropdown menus for "Android" (selected) and "Samsung Galaxy S5" (selected). To the right is a large orange "Start testing" button. Below these are two tabs: "Mobile" (selected) and "Tablet". The main area is a grid of device options:

Samsung	Google	Motorola	Amazon
Galaxy S5 1080x1920	* Nexus 5 1080x1920	Droid Razr 540x960	* Kindle Fire HDX 7 1200x1920
Galaxy S4 1080x1920	Nexus 4 768x1280	Razr 540x960	* Kindle Fire 2 600x1024
Galaxy S3 720x1280	Nexus 7 720x1280	Razr Maxx HD 720x1280	* Kindle Fire HD 8.9 1200x1920
Galaxy S2 480x800	HTC	Sony	Google
Galaxy S 480x800	One M8 1080x1920	Xperia X10 480x854	Nexus 7 1280x800
Galaxy Note 3 1080x1920	One X 720x1280	Xperia Tipo 320x480	4.1
Galaxy Note 2 720x1280	Wildfire 240x320	2.2	Samsung
Galaxy Note 800x1280	Hero 320x480	1.5	Galaxy Tab 4 10.1 1280x800
Galaxy S5 Mini 720x1280			Galaxy Note 10.1 800x1280
			Galaxy Tab 2 10.1 1280x800

* Local testing not supported

Geb
(pronounced “jeb”)

BrowserStack



Geb + BrowserStack

```
1. def browserStackBrowser = System.getProperty("geb.browserstack.browser")  
2. if (browserStackBrowser) {  
3.     driver = {  
4.         def username = System.getenv("GEB_BROWSERSTACK_USERNAME")  
5.         def accessKey = System.getenv("GEB_BROWSERSTACK_AUTHKEY")  
6.         new BrowserStackDriverFactory().create(  
7.             browserStackBrowser, username, accessKey)  
8.     }  
9. }
```

Testing Frameworks

Geb 可以搭配多種測試框架

- JUnit 3
- JUnit 4
- **Spock**
- TestNG
- Cucumber
- EasyB

Geb 與 Spock 搭配很對味



Framework	JAR	Class Name
Spock	geb-spock	geb.spock.GebSpec
JUnit 3	geb-junit3	geb.junit3.GebTest
JUnit 4	geb-junit4	geb.junit4.GebTest
TestNG	geb-testng	geb.testng.GebTest

Geb

(pronounced "jeb")

Page Object

```
1. class LoginPage extends Page {  
2.     static url = "http://myapp.com/login"  
3.     static at = { heading.text() == "Please Login" }  
4.     static content = {  
5.         heading { $("h1") }  
6.         loginForm { $("form.login") }  
7.         loginButton(to: DashboardPage) { loginForm.login() }  
8.     }  
9. }
```

Geb

(pronounced “jeb”)

GebSpec

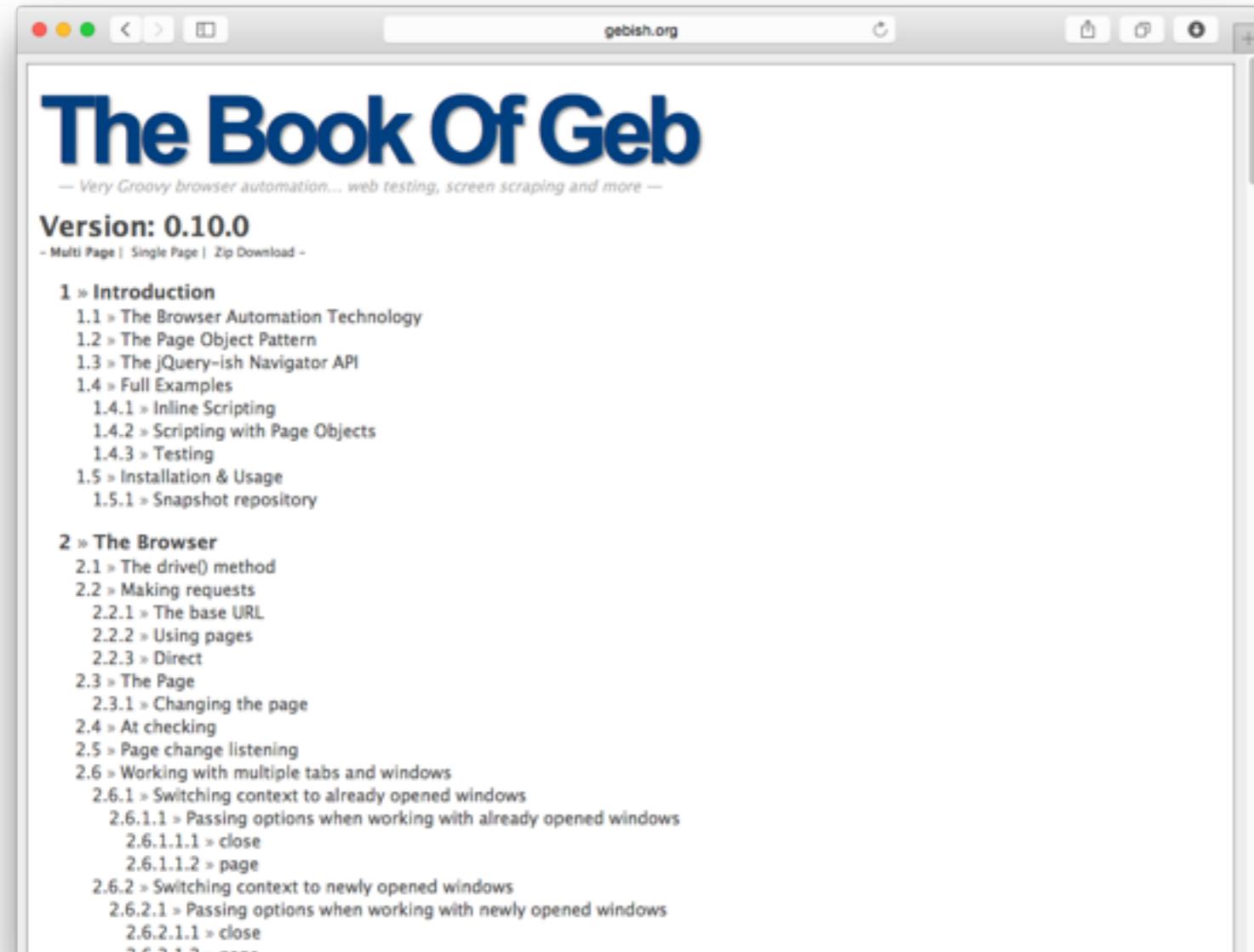
```
1. class LoginSpec extends GebSpec {  
2.     def "login to dashboard section"() {  
3.         given:  
4.             to LoginPage  
5.  
6.         when:  
7.             loginForm.with {  
8.                 username = "admin"  
9.                 password = "password"  
10.            }  
11.  
12.        and:  
13.            loginButton.click()  
14.  
15.        then:  
16.            at DashboardPage  
17.        }  
18.    }
```

開發流程整合

- Gradle + Geb
 - <https://github.com/geb/geb-example-gradle>
- Maven + Geb
 - <https://github.com/geb/geb-example-maven>
- Grails + Geb
 - <https://github.com/lyhcode/grails-realestate>

Geb
(pronounced “jeb”)

官方學習資源



www.gebish.org/manual/current/

Geb
(pronounced “jeb”)

我的教學文章



1. Groovy Tutorial (7) 使用 Geb 開發 Web Test 網站自動化測試（上）
2. Groovy Tutorial (8) 使用 Geb 開發 Web Test 網站自動化測試（中）
3. Groovy Tutorial (9) 使用 Geb 開發 Web Test 網站自動化測試（下）

www.codedata.com.tw/tag/groovy/

限時免費

Geb
(pronounced "jeb")

我的電子書

正體中文版



learngeb.readbook.tw

Java / Groovy 程式設計系列

啟動 Geb

網站自動化測試之美

Learning Web Test with Geb

啟動 Geb

readbook , alincode and lyhcode

網站自動化測試之美

READ DONATE

You can also download this book as: ePUB (for iPad, iPhone, Mac), MOBI (for Kindle) or PDF (for Windows and others).

Subscribe to this book updates:

enter your email Subscribe

Published with GitBook Contact

線上直接閱讀全文

PDF, ePub, Kindle
離線閱讀電子書檔案格式下載

幫推、按讚、分享

learngeb.readbook.tw

Return to book

Review this book

About the author

前言

1. 簡介

- 1.1. 測試！測試！
- 1.2. 認識 Selenium
- 1.3. 認識 Geb

2. 安裝

- 2.1. 安裝 Groovy
- 2.2. 安裝 Gradle
- 2.3. 安裝 Maven

3. 入門

- 3.1. Run Geb with Groovy Shell
- 3.2. GoogleSearch command line ...
- 3.3. 執行 Geb Script
- 3.4. Navigator API
- 3.5. 撰寫 Geb 程式碼
- 3.6. 第一個自動化測試程式

網站自動化測試之美

Learning Web Test with Geb

作者

- 劉艾霖（資深網站開發者）
- 林彥宏（Java / Groovy 開發技術推廣者）

這是一本開放源碼電子書，使用 GitBook 協作出版，本書全文與範例程式碼放在 GitHub 網站。

<https://github.com/lyhcode/learning-geb>

摘要

Geb 是瀏覽器自動化 (browser automation) 的解決方案。

以強大的 Selenium WebDriver 作為基礎，直接控制瀏覽器進行網站操作，而非模擬 HTTP 存取的方式，即使面對 AJAX 或是複雜的 RIA (Rich Internet Application) 也能輕鬆駕馭。

發揮 Groovy 程式語言作為 DSL (Domain-Specific Language) 的優勢，建構出一套瀏覽器自動化專屬的語法，並提供類似 jQuery 的操作方法，存取網頁 DOM 節點更方便，撰寫測試程式更簡潔快速。

可以用簡單的 Groovy Script 方式執行，也可以配合 Maven 或 Gradle 建置專案，能夠無縫整合 JUnit、Spock 或 TestNG 等測試框架，並且搭配 Jenkins CI 進行專案開發流程整合。

如何取得電子書？

learngeb.readbook.tw

Q & A



lyhcode@gmail.com
lyhcode.info

Thank You