모바일 테스팅 플랫폼 분야
Stack 통합 Test 결과보고서
[Selendroid]

2014. 07.

nipa 정보통신산업진흥원 National IT Industry Promotion Agency
nipa 공개SW역량프라자 Open Source Software COMPENTENCE PLAZA

목 차

I. Stack 통합 테스트 개요

공개SW Stack 통합테스트는 여러 공개SW들의 조합으로 시스템 Stack을 구성한 후 Stack을 구성하는 공개SW의 상호 운용성에 중점을 두고 기능 및 성능테스트 시나리오를 개발하여 테스트를 진행한다.

본 통합테스트를 통해 안정된 Stack 정보를 제공하여 민간 및 공공 정보시스템 도입 시 활용될 수 있도록 한다.

1. 목적

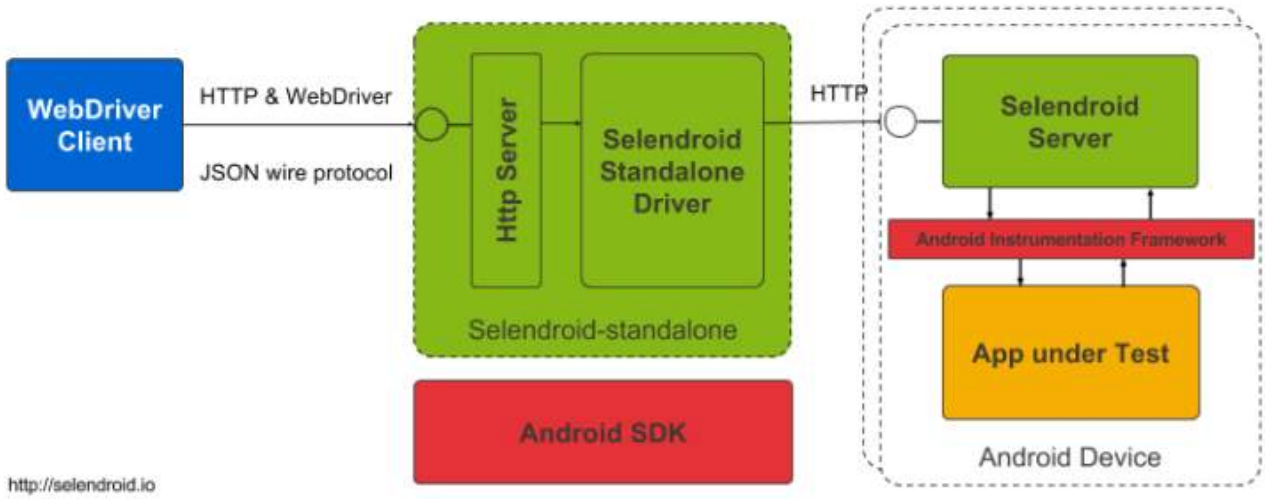
□ 공개SW Stack 통합 테스트 수행 목적

- o 공개SW로 구성된 Stack이 유기적으로 잘 동작함을 확인
- o 다양한 Stack 구성에 기반을 둔 테스트를 통해 안정된 Stack 조합 규명
- o 공개SW 시스템 도입을 위한 Stack 참조모델의 신뢰성 정보로 활용
- o 공개SW의 신뢰성과 범용성에 대한 사용자 인식 제고

Ⅱ. 테스트 대상 소개

1. Selendroid 소개

Selendroid는 안드로이드 및 iOS 모바일 환경을 대상으로 어플리케이션 (native and hybrid application)의 UI와 모바일 웹을 테스트 할 수 있도록 시스템 구성 요소를 제공하는 테스트 자동화 프레임워크이다. Selendroid는 테스트 환경 구성 시, 모바일 실 디바이스와 에뮬레이터 환경의 구분 설정이 가능하므로 다양한 테스트 환경을 제공한다.



[그림 1-1. Selendroid 기능 구조]

□ Selendroid 기능 구조

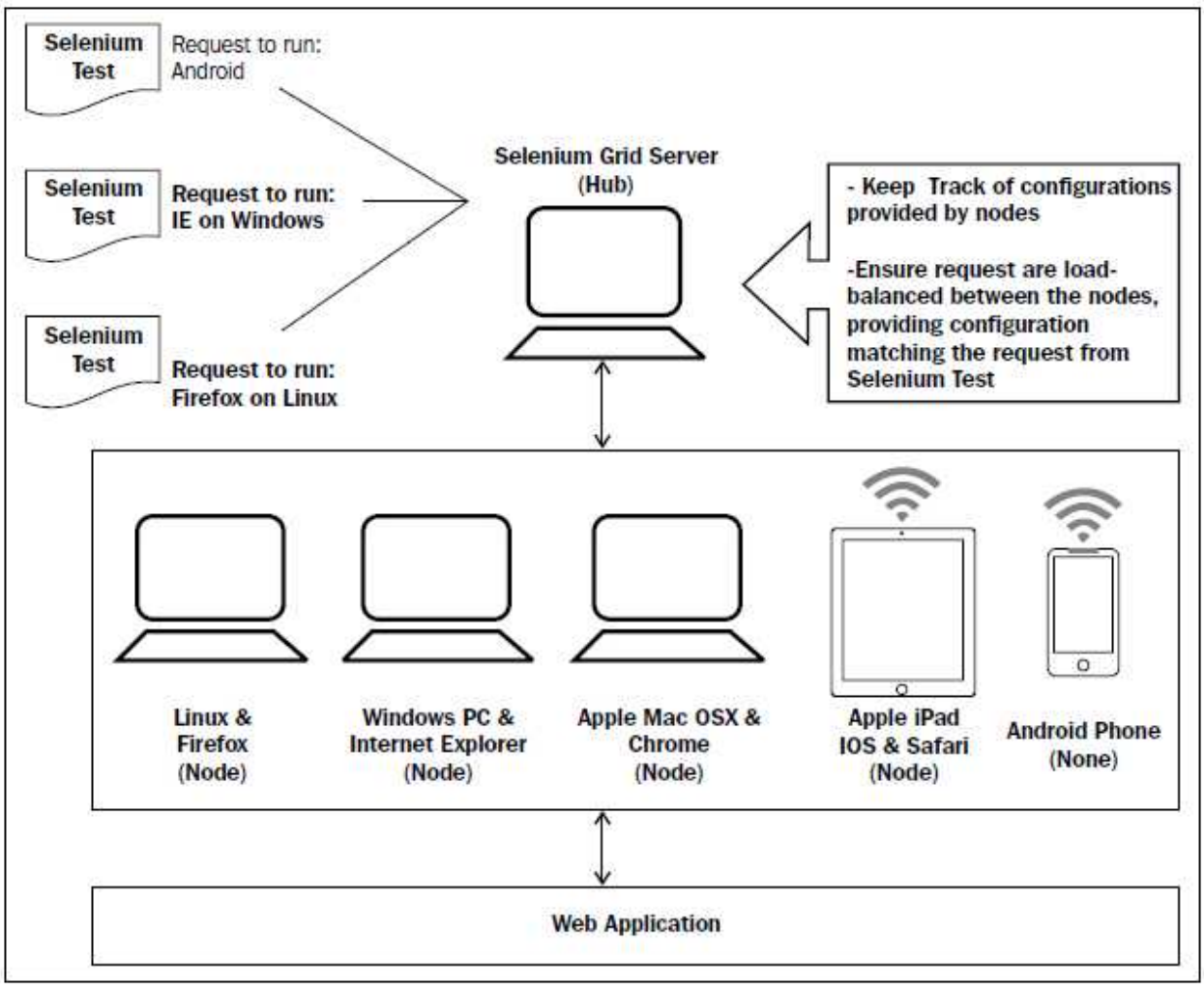
Selendroid는 다음과 같은 4개 주요 기능 요소로 구성된다.

·Selendroid-Client : Selenium Java Client에 기초한 Java Client Library

·Selendroid-Server : 안드로이드 디바이스 상 설치된 어플리케이션과 연동되어 테스트 환경을 구성하는 app automation 기능단위

·AndroidDriver-App : 모바일 web을 테스트하기 위한 안드로이드 드라이버 webview application

·Selendroid-Standalone : Selendroid-Clinet 와 Selendroid-Server 간 Proxy 역할을 담당하고 안드로이드 에뮬레이터 동작 기능 및 Selendroid-server의 생성과 AUT(App Under Test) 설치기능을 포함하여 테스트 대상인 안드로이드 디바이스를 관리하는 기능단위



[그림 1-2. Selenium Grid Architecture]

Selendroid는 Selenium Grid Architecture의 적용이 가능하므로 Selendroid-standalone 서버 및 Selendroid-Grid-Plugin 설정을 통해 안드로이드 OS와 iOS 별 적용되는 모바일 Web Browser를 대상으로 유연한 테스트 환경을 구성할 수 있다.

□ Selendroid 시스템 요구사항 (2014년 7월 30일 기준)

항목	정보
OS (Android SDK 지원 기준)	- Windows XP(32-bit), Vista(32-or 64-bit), or Windows 7(32-or 64bit) - Mac OS X 10.5.8 or later (x86 only) - Linux (tested on Ubuntu Linux, Lucid Lynx) • GNU C Library (glibc) 2.7 ot later is required • On Ubuntu Linux, version 8.04 or later is required • 64-bit distributions must be capable of running 32-bit applications
JAVA SDK	- minimum 1.6
Android SDK	- latest version (http://developer.android.com/sdk/index.html 참조)

[표 1. Selendroid 시스템 요구사항]

※ 추가적인 자세한 정보는 아래의 링크 내역 참조

- http://selendroid.io/setup.html
- http://developer.android.com/sdk/index.html
- http://www.packtpub.com/sites/default/files/downloads/Distributed_Testing_with_Selenium_Grid.pdf

Ⅲ. Stack 통합 테스트

1. 테스트 환경

□ Selendroid SW 환경

SW	Version
Mobile Application APK	0.10.0
Android SDK	Latest Android-SDK
Java SDK	1.8.0_05
Eclipse	Luna Release(4.4.0)

[표 2. 테스트 SW 환경] - 2014년 7월 30일 기준

□ Stack 환경

Stack	OS	네트워크 정보(IP)
A (Selendroid Server)	Ubuntu 12.04 LTS (64-bit)	121.162.249.95
A (Android Mobile)	Kitkat 4.4.2	N/A

[표 3. Stack 환경]

※ 참고 - Stack A의 PC 상 Selendroid-standalone Server 설정 및 JDK와 Eclipse 적용에 따라 모바일 OS 별 Real Device 및 Emulator 환경 구성을 통해 제반 인터페이스 관련 기능 테스트 수행.

□ HW 환경

※ PC 환경

제조사	모델명	CPU	MEM	Disk	NIC
HP	dc7900 CMT	Quad-Core 2.66Ghz~4P	2.8GiB	265GB	Gigabit 1Port

※ Mobile Device 환경

제조사	모델명	CPU	MEM	RAM	해상도
Samsung	SHV-E330S	옥타코어(1.6GHz 쿼드 + 1.2GHz 쿼드)	32GiB	2GB	1920×1080

[표 4. HW 환경]

2. 주요 테스트 방법

□ 탐색적 테스팅(Exploratory Testing)

탐색적 테스팅은 테스트 엔지니어의 지적 능력을 최대한 공유, 활용하는 것을 목적으로 하는 테스트 접근법으로 테스트를 수행할 대상을 실행시켜 사용함과 동시에 사용 측면에서 문제가 되는 부분에 집중하여 테스트를 설계 및 계획한다. 이러한 과정은 효율적 진행을 위한 Time Boxing을 통해 수행되므로 테스트 케이스 작성을 최소화할 수 있고, 상대적으로 적은 시간에 집중적인 테스트를 가능하게 한다.

□ 리스크 기반 테스트(Risk based Testing)

리스크 기반 테스팅 기법은 테스트 대상에 비해서 테스트 자원이 부족한 경우 효과적이고, 효율적인 테스트 수행을 위해 적용 될 수 있다. 해당 기법은 크게 리스크 식별과 리스크 분석, 그리고 리스크 계획의 세 단계로 구분 진행된다.
리스크 식별 단계에서는 제품 품질관점에서 테스트 대상이 될 항목을 식별하고, 프로젝트나 제품에 대한 리스크 요소를 식별한다.
리스크 분석 단계에서는 장애 발생가능성과 장애로 인한 영향을 식별하고 리스크 우선순위를 결정한다.
마지막으로 리스크 계획 단계에서는 리스크의 우선순위에 따른 대처 방안 및 완화 정책을 수립하며, 이후 테스트 수행 시 커버리지를 고려하여 선택과 집중을 통해 테스트를 수행하게 된다.

□ 시나리오 테스트

시나리오 테스트 기법은 단일 기능에 대한 결함 여부를 확인하는 것이 아니라, 서로 다른 컴포넌트 사이의 상호작용과 간섭으로 발생할 수 있는 결함을 발견하기 위한 기법이다.

본 테스트에서는 리스크 분석을 통해 Selendroid의 기능 및 비기능 항목들에 대한 기능적/ 기술적 아이템을 정의하였다. 또한, 각 정의된 아이템 별 사용자 시나리오를 바탕으로 테스트 아이디어를 도출하였다.

3. 기능 테스트 수행 결과

기능 테스트 수행 관련 세부 케이스는 별첨 「Selendroid 테스트 케이스」 문서를 참고한다.

□ 탐색적 테스트 현황

기능 아이템	기본 차터	테스트 아이디어
Selendroid 환경구성	1	19
Selendroid 테스팅	2	12
Selendroid Inspector	1	8
Selendroid Scaling	1	4
합 계	5	43

[표 5. 테스트 아이디어 현황]

□ 테스트 결과

탐색적 테스트를 통한 테스트 수행 결과 내용은 아래와 같다.

분류		PASS	FAIL	N/T	N/A
기능	테스트 아이디어				
Selendroid 환경구성	19	19	0	0	0
Selendroid 테스팅	12	12	0	0	0
Selendroid Inspector	8	8	0	0	0
Selendroid Scaling	4	4	0	0	0

[표 5. 테스트 결과]

Ⅵ. 종합

□ Selendroid 테스트 수행 결과 공개 SW로 구성된 Stack 상에서 치명적 이슈 발생 없이 모바일 테스팅 프레임워크로서 제공하는 제반 기능들이 각 기능 resource들과 유기적으로 동작함을 확인하였다.

□ Selendroid는 Mobile Testing Framework로서 Selenium API 및 Grid Architecture의 적용이 가능하다. 또한, 안드로이드 OS 및 iOS 시스템에 대한 Mobile Device 테스트 환경을 제공하고, emulator 설정이 가능하여 실 디바이스(Real Device) 없이도 모바일 web 및 application을 대상으로 테스팅이 가능하다.

테스트 suite의 스크립트 작성의 경우, Java를 비롯한 Ruby, Python 적용이 가능하다.

Selendroid는 Plugin 설정을 통해 개발 환경인 Eclipse와 연동되어 Dashboard 기능 및 log관리와 Debugging 기능을 제공하므로 다양한 모바일 테스팅 환경에 유용한 솔루션으로 적용될 수 있다.

※ 참고 자료

[1] www.selendroid.io

[2] https://github.com/selendroid

[3] http://search.maven.org/#search|ga|1|selendroid

[4] http://docs.seleniumhq.org/docs/03_webdriver.jsp

[5] http://www.ontestautomation.com/category/test-automation-tools/