

# SW Verification

201211387 하헌규

201211337 김재현

201211354 성현구

201211369 이병찬

# 목차

---

- 정적분석도구
- 시스템테스트
- CTIP

- 정적분석 : 프로그램 실행 없이 코드를 분석하는 단계
- 대표적으로 PMD, Findbugs, Checkstyle 등이 존재
- 본 프로젝트에서는 Findbugs, JDepend, SonarCloud 사용

# 정적분석도구 - Findbugs

- JAVA프로그램 분석도구
- 오브젝트 코드(.class)를 분석
- 바이트 코드를 읽어 속도가 빠르지만 빌드 과정이 필수

# 정적분석도구 - Findbugs

```
findbugs {
    ignoreFailures true
}

tasks.withType(FindBugs) {
    reports {
        xml.enabled false
        html.enabled true
    }
}
```

◀ build.gradle 설정 화면

```
:findbugsMain
Scanning archives (1 / 1)
2 analysis passes to perform
Pass 1: Analyzing classes (22 / 22) - 100% complete
Pass 2: Analyzing classes (1 / 1) - 100% complete
Done with analysis
FindBugs rule violations were found. See the report at: file:///home/travis/build/Nineclown/sonarqube-test/build/reports/findbugs/main.html
```

▲ travis를 통한 빌드

# 정적분석도구 - Findbugs

## [FindBugs](#) Report

### Project Information

Project:

FindBugs version: 3.0.1

Code analyzed:

- C:\Users\nineClown\Documents\GitHub\sonarqube-test\build\classes\java\main

### Metrics

89 lines of code analyzed, in 1 classes, in 1 packages.

Metric	Total	Density*
High Priority Warnings	2	22.47
Medium Priority Warnings	1	11.24
<b>Total Warnings</b>	<b>3</b>	<b>33.71</b>

(\* Defects per Thousand lines of non-commenting source statements)

### Contents

- [Internationalization Warnings](#)

◀ Findbugs 실행 결과 화면

# 정적분석도구 - JDepend

- 패키지별로 의존성 측정 가능
- 패키지 의존성과 관련된 데이터 품질을 수치화
- 결과를 그래프로 표현 가능

# 정적분석도구 - JDepend

```
tasks.withType(JDepend) {  
    reports {  
        xml.enabled = true  
        text.enabled = false  
    }  
}
```

◀ build.gradle 설정 화면

```
526 :jdependMain  
527 Download https://repo1.maven.org/maven2/jdepend/jdepend/2.9.1/jdepend-2.9.1.pom  
528 Download https://repo1.maven.org/maven2/org/apache/ant/ant-jdepend/1.9.6/ant-jdepend-1.9.6.pom  
529 Download https://repo1.maven.org/maven2/org/apache/ant/ant-parent/1.9.6/ant-parent-1.9.6.pom  
530 Download https://repo1.maven.org/maven2/org/apache/ant/ant-jdepend/1.9.6/ant-jdepend-1.9.6.jar  
531 Download https://repo1.maven.org/maven2/jdepend/jdepend/2.9.1/jdepend-2.9.1.jar  
532 :jdependTest  
533 :testPicked up _JAVA_OPTIONS: -Xmx2048m -Xms512m  
534
```

▲ travis를 통한 빌드



# 정적분석도구 - JDepend

## JDepend Analysis

Designed for use with [JDepend](#) and [Ant](#).

### Summary

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

Package	Total Classes	Abstract Classes	Concrete Classes	Afferent Couplings	Efferent Couplings	Abstractness	Instability	Distance
Default	1	0	1	0	1	0	1	0
java.lang	No stats available: package referenced, but not analyzed.							

### Packages

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

#### Default

[Afferent Couplings](#): 0

[Efferent Couplings](#): 1

[Abstractness](#): 0

[Instability](#): 1

[Distance](#): 0

Abstract Classes	Concrete Classes	Used by Packages	Uses Packages
None	sum	None	<a href="#">java.lang</a>

▲ JDepend 실행 결과 화면

# 정적분석도구 - SonarQube

- 여러가지 분석 툴을 통합한 스타일의 정적분석 도구
- 웹페이지에서 분석 결과를 쉽게 확인 가능
- SonarLint(IDE 플러그인), SonarCloud(클라우드 버전) 등의 파생 프로그램 존재 (본 프로젝트에서는 SonarCloud를 사용)

# 정적분석도구 - SonarCloud

```
.travis.yml × Test.java × sonarqube-test × gradlew × grad
1  language: java
2  jdk:
3    - oraclejdk8
4  install: true
5
6  addons:
7    sonarcloud:
8      organization: nineclown-github
9      token:
10     secure: 20cccee07ce13984260fc360c226cdc1ad1b0ed5
11
12  before_install:
13    - chmod +x gradlew
14
15  script:
16    ./gradlew clean check sonarqube build
17
18  cache:
19    directories:
20      $HOME/.m2/repository
21      $HOME/.sonar/cache
22      $HOME/.gradle
23      .gradle
24
```

◀ .travis.yml파일 설정 화면

# 정적분석도구 - SonarCloud

```
.travis.yml × Test.java × sonarqube-test × gradlew × gradlew.bat ×
1  plugins {
2      id "org.sonarqube" version "2.6"
3  }
4
5  apply plugin: 'java'
6
7  group 'org.test.sonarqube'
8  version '1.0-SNAPSHOT'
9  sourceCompatibility = 1.8
10
11 repositories {
12     mavenCentral()
13 }
14
15 dependencies {
16     testCompile group: 'junit', name: 'junit', version: '4.12'
17 }
18
```

◀ build.gradle 설정 화면

# 정적분석도구 - SonarCloud

```
4.0/bin/gradle
478 Download https://plugins.gradle.org/m2/org/sonarsource/scanner/gradle/sonarqube-gradle-plugin/2.6/sonarqube-gradle-plugin-2.6.pom
479 Download https://plugins.gradle.org/m2/org/sonarsource/scanner/api/sonar-scanner-api/2.9.0.887/sonar-scanner-api-2.9.0.887.pom
480 Download https://plugins.gradle.org/m2/org/sonarsource/scanner/api/sonar-scanner-api-parent/2.9.0.887/sonar-scanner-api-parent-2.9.0.887.pom
481 Download https://plugins.gradle.org/m2/org/sonarsource/parent/parent/40/parent-40.pom
482 Download https://plugins.gradle.org/m2/org/sonarsource/scanner/gradle/sonarqube-gradle-plugin/2.6/sonarqube-gradle-plugin-2.6.jar
483 Download https://plugins.gradle.org/m2/org/sonarsource/scanner/api/sonar-scanner-api/2.9.0.887/sonar-scanner-api-2.9.0.887.jar
484 :clean UP-TO-DATE
485 :compileJava
486 :processResources NO-SOURCE
487 :classes
488 :compileTestJava NO-SOURCE
489 :processTestResources NO-SOURCE
490 :testClasses UP-TO-DATE
491 :test NO-SOURCE
492 :check UP-TO-DATE
493 :sonarqube
494 Download https://repo1.maven.org/maven2/junit/junit/4.12/junit-4.12.pom
495 Download https://repo1.maven.org/maven2/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.pom
496 Download https://repo1.maven.org/maven2/org/hamcrest/hamcrest-parent/1.3/hamcrest-parent-1.3.pom
497 Download https://repo1.maven.org/maven2/junit/junit/4.12/junit-4.12.jar
498 Download https://repo1.maven.org/maven2/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar
499 :jar
500 :assemble
501 :build
502
503 BUILD SUCCESSFUL in 59s
504 4 actionable tasks: 3 executed, 1 up-to-date
505
506
507 The command ". /gradlew clean check sonarqube build" exited with 0.
508 store build cache
509
510
511 Done. Your build exited with 0.
```

Travis CI About Us Blog Status Help Hyeongoo Sung

Nineclown / sonarqube-test build unknown

Current Branches Build History Pull Requests More options

✓ master i wanna go home -> #3 passed Restart build

Commit 9a60ef0 Ran for 2 min 8 sec  
Compare ddb9b2c..9a60ef0 3 minutes ago  
Branch master  
nineclown authored and committed

Job log View config

Remove log Raw log

```
1 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
2 Build system information system_info
399
400 removed '/etc/apt/sources.list.d/basho_riak.list'
401 Network availability confirmed.
402 127.0.0.1 localhost
403 ::1 ip6-localhost ip6-loopback
404 fe00::0 ip6-localnet
405 ff00::0 ip6-mcastprefix
406 ff02::1 ip6-allnodes
407 ff02::2 ip6-allrouters
408 172.17.0.12 travis-job-nineclown-sonarqube-test-365942892.travis-ci.net travis-job-nineclown-sonarqube-test-365942892
409 W: http://ppa.launchpad.net/couchdb/stable/ubuntu/dists/trusty/Release.gpg: Signature by key
```

◀ Travis 상에서 SonarQube 구동

# 정적분석도구 - SonarCloud

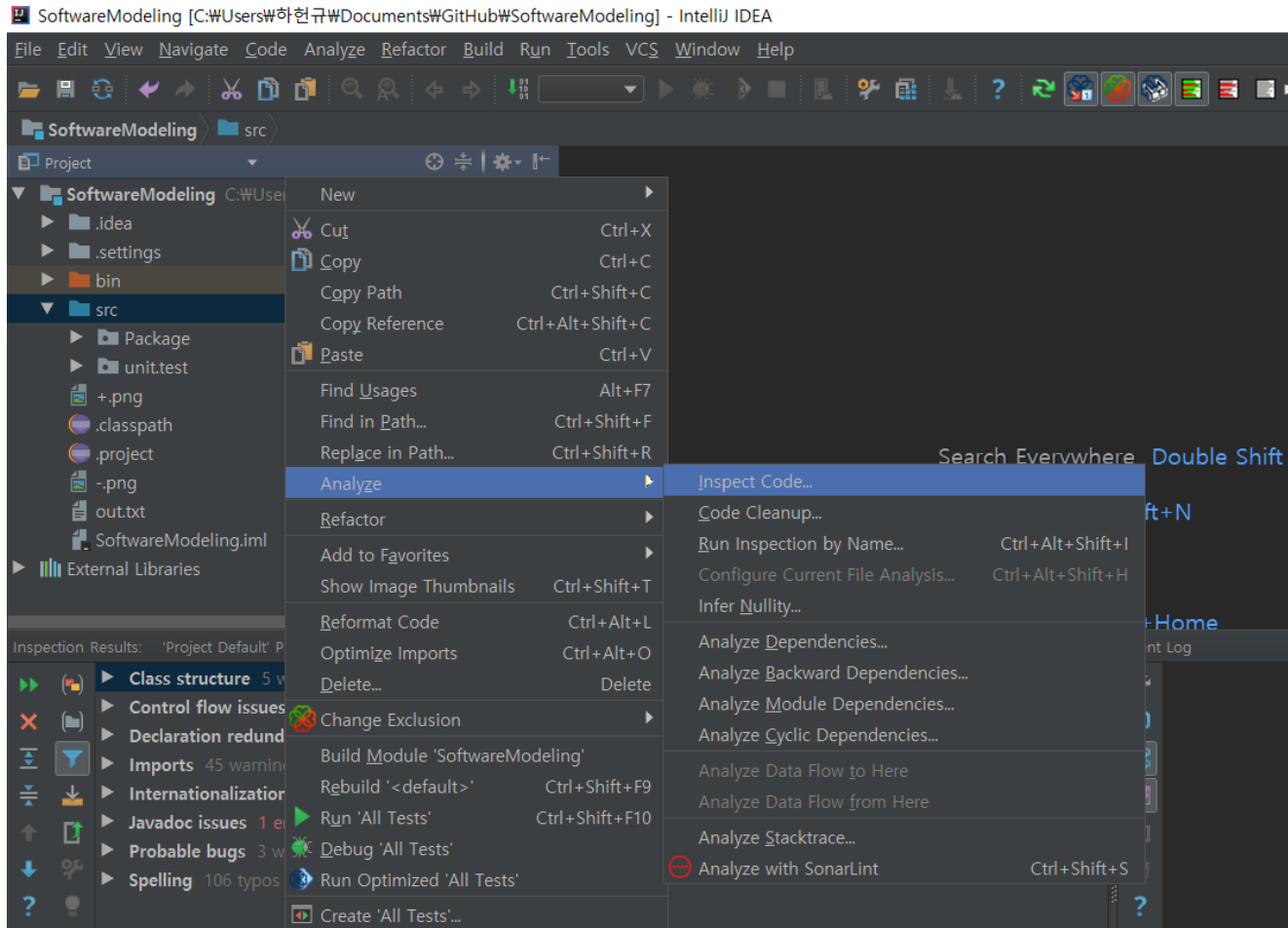
The screenshot displays the SonarCloud web interface for a project named 'sonarqube-test'. The main dashboard shows the following metrics:

- Quality Gate:** Passed
- Bugs:** 1 (C)
- Vulnerabilities:** 0 (A)
- Code Smells:** 16 (B)
- Coverage:** 0.0%
- Duplications:** 14.5% (2 Duplicated Blocks)
- Lines of Code:** 99

The interface also includes a left sidebar with filters for Quality Gate (Passed, Warning, Failed) and Reliability (A, B, C). The top navigation bar shows 'My Projects' and 'My Issues'. The bottom of the dashboard indicates '1 of 1 shown'.

▲ 빌드에 성공하면 자동으로 SonarCloud Project 생성

# 정적분석도구 - IntelliJ



◀ IntelliJ 정적 분석 도구

# 정적분석도구 - IntelliJ

## ◀ IntelliJ 정적 분석 결과

- ▶ Class structure 5 warnings
- ▶ Control flow issues 2 warnings
- ▶ Declaration redundancy 101 warnings
- ▶ Imports 45 warnings
- ▶ Internationalization issues 8 warnings
- ▶ Javadoc issues 1 error 3 warnings
- ▶ Probable bugs 3 warnings
- ▶ Spelling 106 typos

- ▶ Class structure 5 warnings
- ▶ Control flow issues 2 warnings
- ▼ Declaration redundancy 101 warnings
  - ▶ Actual method parameter is the same constant 1 warni
  - ▶ Declaration access can be weaker 51 warnings
  - ▶ Declaration can have final modifier 13 warnings
  - ▼ Empty method 4 warnings
    - ▼ Test 4 warnings
      - The method is empty
      - The method is empty
      - The method is empty
      - The method is empty

💡 Delete unnecessary method(s) >>

```
@Override  
public void keyPressed(KeyEvent arg0) {}
```



# 시스템테스트 - TestRail

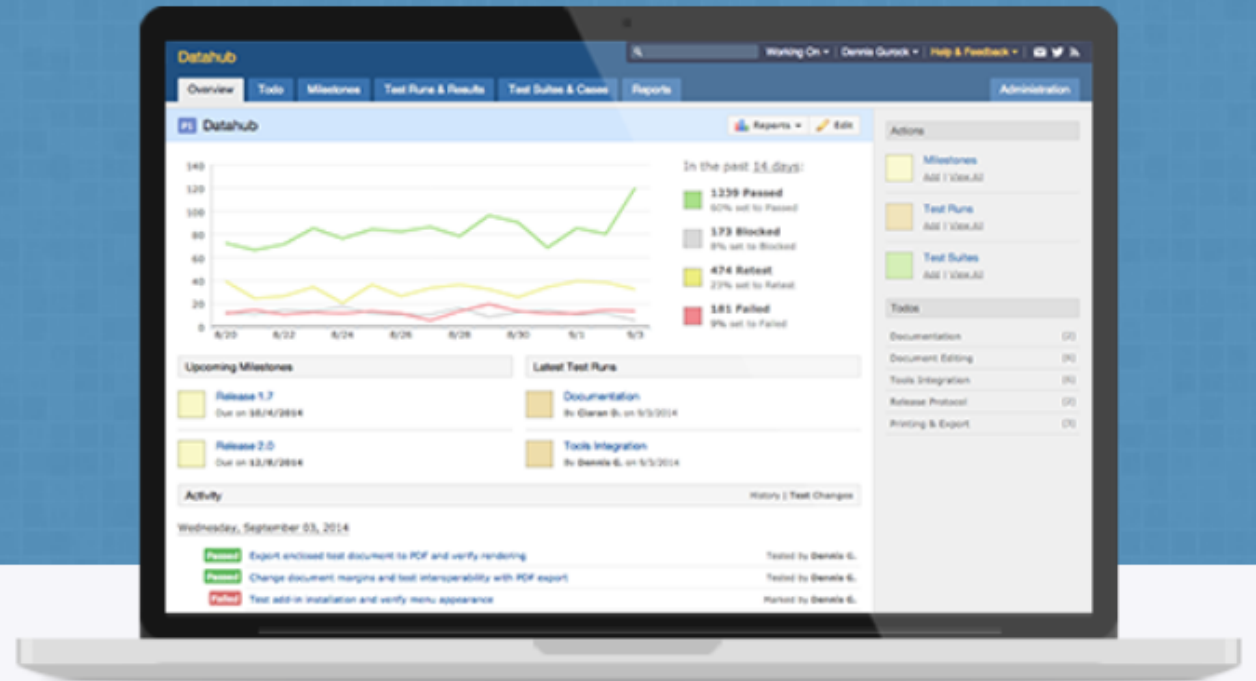
## GitHub Test Management with TestRail

Integrate our test case management software with GitHub

- Manage test cases, plans and runs
- Start test runs and capture test results
- Push bug reports to GitHub and link issue IDs
- Track test results, activity and progress

Try TestRail For Free

See Features



# 시스템테스트 - TestRail

The screenshot shows the TestRail QA dashboard. At the top, there is a dark blue header with the text 'TestRail QA' on the left and navigation links 'Working On', 'ha hun kyu', and 'Help & Feedback' on the right. Below the header, there are two tabs: 'Dashboard' (selected) and 'Administration'. The main content area is titled 'All Projects' and features a large yellow informational box. This box contains an information icon, the heading 'Add your first project to TestRail', a welcome message, and an 'Add Project' button. To the right of this box is a 'New to TestRail?' section with a tutorial link. On the far right, there is a sidebar with an 'Add Project' button, a status indicator '0 active and 0 completed projects.', and a 'Todos' section showing 'None.'

▲ 초기 화면

# 시스템테스트 - TestRail



## Add Project

**Project** Access Defects References

**Name \***

Ex: New Widget, Intranet or Payroll Software

**Announcement**

---

You can post an announcement to the project overview page. This could include links to the project's issue tracker or knowledge base, for example.

**Show the announcement on the overview page**

▲ 프로젝트 생성

# 시스템테스트 - TestRail

P2 software\_Verification\_test

Reports Edit



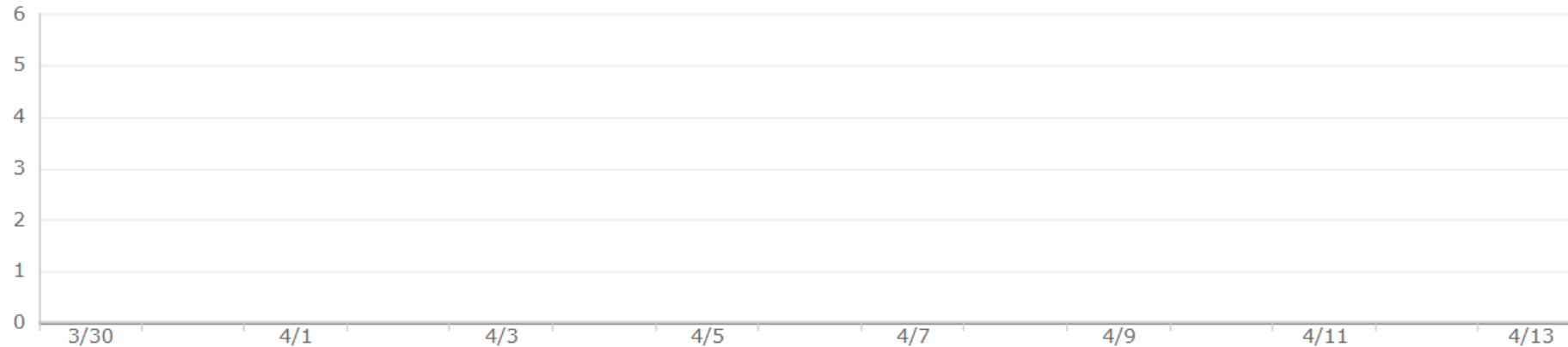
### Congratulations! You have created your first project

Now that you have created your first project in TestRail, you can add your project milestones and create your first test cases. If you are new to TestRail, it is highly recommended to take a look at TestRail's [user guide and manual](#).

Next recommended steps:

- Read [Getting Started](#)
- Add [project milestones](#)
- Enter [test cases](#)
- Add [additional users](#)

test testrail and github



In the past 14 days:

- 0 Passed**  
0% set to Passed
- 0 Blocked**  
0% set to Blocked
- 0 Retest**  
0% set to Retest
- 0 Failed**  
0% set to Failed

# 시스템테스트 - TestRail

Overview | Todo | Milestones | Test Runs & Results | **Test Cases** | Reports

**Title \***  
test

<b>Section *</b> Test Cases ▼	<b>Template *</b> Test Case (Text)
<b>Type *</b> Other ▼	<b>Priority *</b> Medium
<b>Estimate</b> <input type="text"/>	<b>References</b> <input type="text"/>





**Preconditions**  
C#으로 다른 TEST CASE와 REFERENCING 가능  
The preconditions of this test case. Reference other test cases with [C#] (e.g. [C17]).

**Steps**  
TEST CASE를 수행하기 위한 STEP  
The required steps to execute the test case.




**Expected Result**  
TEST CASE 실행 후 예상되는 결과  
The expected result after executing the test case.

Test Case 추가 ▶

# 시스템테스트 - TestRail




Sort: Section | Filter: None |  Add Case |  Edit ▼ |  Delete |  Columns

**Test Cases (2)**

<input type="checkbox"/>	<input type="checkbox"/>	ID	Title	
	<input type="checkbox"/>	C4	1-1	
	<input type="checkbox"/>	C6	1-2	

[Add Case](#) | [Add Subsection](#)

**TEST (2)**

<input type="checkbox"/>	<input type="checkbox"/>	ID	Title	
	<input type="checkbox"/>	C7	2-1	
	<input type="checkbox"/>	C8	2-2	

[Add Case](#) | [Add Subsection](#)

# 시스템테스트 - TestRail

Navigation: Overview | Todo | Milestones | **Test Runs & Results** | Test Cases | Reports

Test Run: R4 Test Run 4/13/2018

Summary:

- 0 Passed (0% set to Passed)
- 0 Blocked (0% set to Blocked)
- 0 Retest (0% set to Retest)
- 0 Failed (0% set to Failed)

**0% Passed**  
1 / 1 untested (100%).

testcase

Sort: Section | Filter: None

Test Cases (1)

ID	Title	Status
T4	test	Untested

T4 test

Type	Priority	Estimate	References
Other	Medium	None	None

**Preconditions**

C#으로 다른 TEST CASE와 REFERENCING 가능

**Steps**

TEST CASE를 수행하기 위한 STEP

**Expected Result**

TEST CASE 실행 후 예상되는 결과

**Results & Comments** | History & Context | Defects

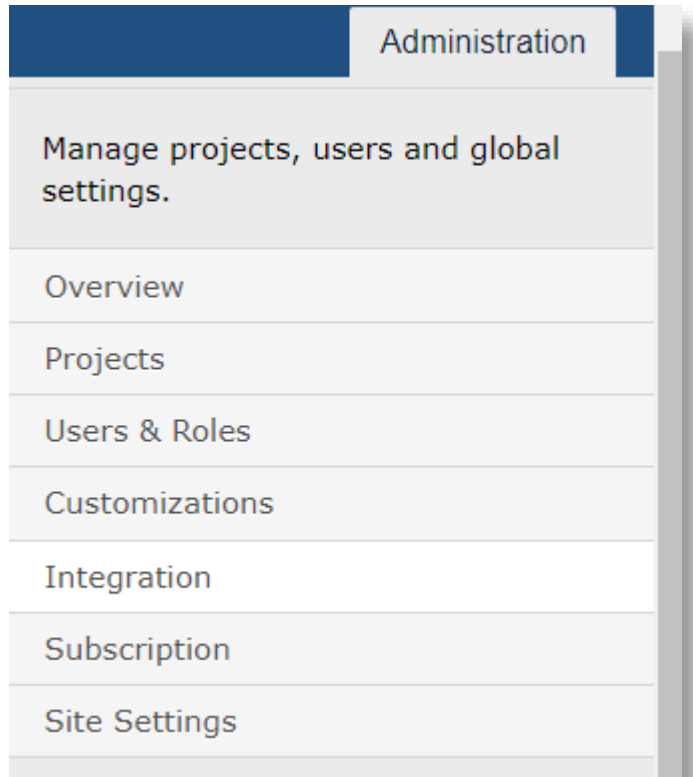
Add a comment ..

No test results or comments so far.

Buttons: Add Result | Pass & Next | Assign To

◀ Test 결과 세팅 전

# 시스템테스트 - TestRail



Github 연동 설정 ▶

A screenshot of the TestRail Defects configuration page. The page has two tabs: "Defects" and "References". The "Defects" tab is active. The page contains the following sections:

- Defect View Url**: A text input field containing "https://github.com/gkgjsrb/software\_verification\_test/issues/%id%". Below it is a description: "The web address of a case of your defect tracker. Use %id% as the placeholder for the actual case ID. [Learn more](#)".
- Defect Add Url**: A text input field containing "https://github.com/gkgjsrb/software\_verification\_test/issues/new". Below it is a description: "The web address for adding a new case to your defect tracker. [Learn more](#)".
- Defect Plugin**: A dropdown menu with "GitHub" selected. Below it is a description: "The plugin for integrating TestRail with your defect tracker. The plugin can be configured below. [Learn more](#)".
- A code block containing configuration instructions for GitHub:

```
; Please configure your GitHub connection below.  
; Note: requires GitHub API v3 or later. You  
; can alternatively specify a GitHub Enterprise  
; server address.  
[connection]  
address=https://api.github.com  
user=gkgjsrb  
password=*****  
  
[repository]  
owner=gkgjsrb  
name=software_verification_test
```
- At the bottom, a note: "Make sure to use HTTPS for a secure connection to your defect tracker. User variables are recommended to store the user & password securely (can also be used to customize the login per user). [Learn more](#)".



# 시스템테스트 - TestRail

### Add Result

**Status \***  
Failed  
Set the test status (*passed, failed* etc.).

**Comment**  
Describe your test result.

Drop files here to attach, or click to browse.

**Assign To**  
Assign to another team member.

**Version**  
The version you tested against.

**Elapsed** Start  
How long the test took.

**Defects** Push Add  
A list of IDs in your bug tracker.

Add Result  Cancel

### Push Defect

**Summary \***  
Failed test: test

**Milestone** **Assignee**  
gkgjsrb

**Labels**  
bug

**Description**  
GITHUB와 연동  
T4: test  
<https://gkgjsrb.testrail.io/index.php?/tests/view/4>

Submit  Cancel

# 시스템테스트 - TestRail

## Failed test: test #4

Edit

New issue

Open

gkgjsrb opened this issue a minute ago · 0 comments



gkgjsrb commented a minute ago

Owner



GITHUB와 연동

T4: test

<https://gkgjsrb.testrail.io/index.php?/tests/view/4>

Assignees



gkgjsrb

Labels



bug

Projects



None yet

Milestone



No milestone



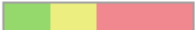


gkgjsrb self-assigned this a minute ago

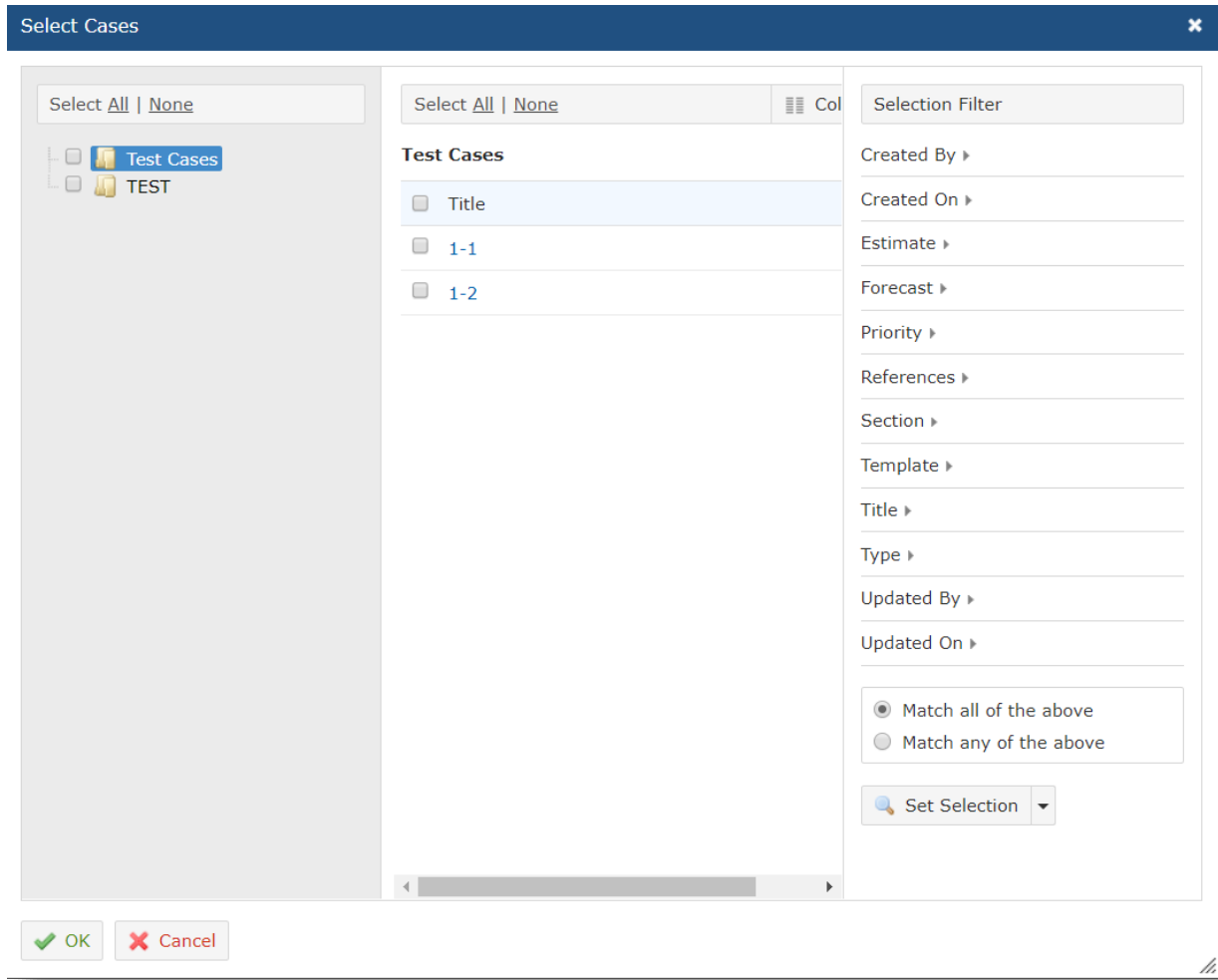


gkgjsrb added the **bug** label a minute ago

# 시스템테스트 - TestRail

Open			
Test Run-3 4/13/2018	ha hun k.		100%
Test Run-2 4/13/2018	ha hun k.		75%
Test Run 4/13/2018	ha hun k.		25%

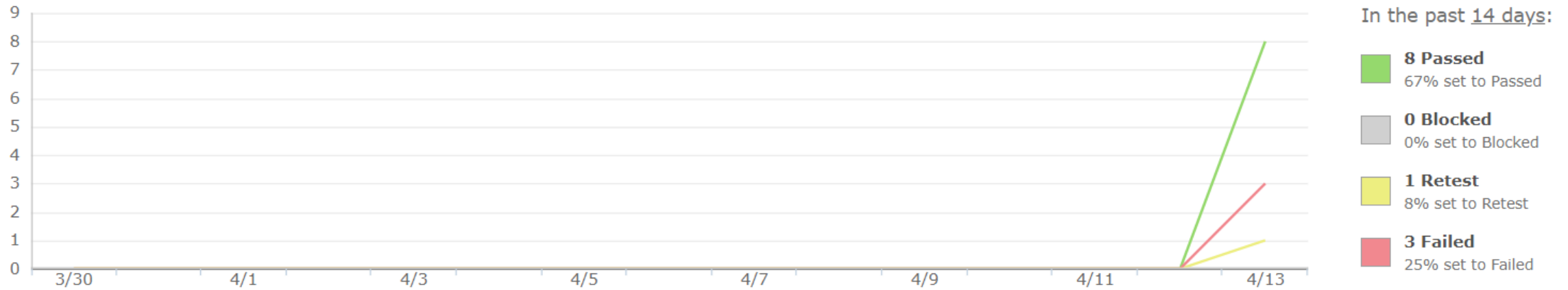
# 시스템테스트 - TestRail



◀ Test Case 관리

# 시스템테스트 - TestRail

test testrail and github



# CTIP



감사합니다 :)